The Dissertation Committee for John Anderson Mills III
certifies that this is the approved version of the following dissertation:

Human-Based Percussion and Self-Similarity Detection in Electroacoustic Music

Committee:

Elmer Hixson, Supervisor

Michael Becker, Supervisor

Brian Evans

Mark Hamilton

Dennis McFadden

Russell Pinkston

Human-Based Percussion and Self-Similarity Detection in Electroacoustic Music

by

John Anderson Mills III, B.S.; M.S.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2008

To all my friends who help me stay the right kind of crazy

Acknowledgments

I would like to express my gratitude to everyone who helped me complete this dissertation. I first must thank my advisor, Dr. Elmer Hixson, for his immensely helpful guidance, patience, and support. I greatly appreciate the efforts of the rest of the committee, Dr. Michael Becker, Dr. Brian Evans, Dr. Mark Hamilton, Dr. Dennis McFadden, and Dr. Russell Pinkston for overseeing this work. I would especially like to thank Dr. Pinkston for transferring a love of composing, analyzing, and listening to electroacoustic music, and to Dr. McFadden for guidance and direction concerning the collection of human percussion judgments. Dr. Heidi Spratt deserves thanks for her guidance on applying statistical verification techniques. Thanks go to Mr. Ted Argo, Mr. Dhruv Bansal, and Mr. Andrew Stalick for technical help, and to Ms. Delia Davila, Ms. Beth Cotton, Ms. Emily Weerts, and Ms. Adrienne Foreman for proofreading the text. Dr. Preston Wilson is thanked for providing laboratory space during portions of this research. Thanks also to the participants of the discussions of audio properties and of the human judgment collection.

I would also like to thank all of the amazing people who helped me to enjoy life while still allowing me to remain focused on my doctoral work. Finally, I would like to express my gratitude toward my parents, who have certainly suffered with me the longest.

John Anderson Mills III

The University of Texas at Austin
August 2008

Human-Based Percussion and Self-Similarity Detection in Electroacoustic Music

Publication No. ____________________

John Anderson Mills III, Ph.D.
The University of Texas at Austin, 2008

Supervisor: Elmer Hixson
Supervisor: Michael Becker

Electroacoustic music is music that uses electronic technology for the compositional manipulation of sound, and is a unique genre of music for many reasons. Analyzing electroacoustic music requires special measures, some of which are integrated into the design of a preliminary percussion analysis tool set for electroacoustic music. This tool set is designed to incorporate the human processing of music and sound. Models of the human auditory periphery are used as a front end to the analysis algorithms. The audio properties of percussivity and self-similarity are chosen as the focus because these properties are computable and informative.

A collection of human judgments about percussion was undertaken to acquire clearly specified, sound-event dimensions that humans use as a percussive cue. A total of 29 participants was asked to make judgments about the percussivity of 360 pairs of synthesized snare-drum sounds. The grouped results indicate that of the dimensions tested rise time is the strongest cue for percussivity. String resonance also has a strong effect, but because of the complex nature of string resonance, it is not a fundamental dimension of a sound event. Gross spectral filtering also has an effect on the judgment of percussivity but the effect is weaker than for rise time and string resonance. Gross spectral filtering also has less effect when the stronger cue of rise time is modified simultaneously.

A percussivity-profile algorithm (PPA) is designed to identify those instants in pieces of music that humans also would identify as percussive. The PPA is implemented using a time-domain, channel-based approach and psychoacoustic models. The input parameters are tuned to maximize performance at matching participants' choices in the percussion-judgment collection. After the PPA is tuned, the PPA then is used to analyze pieces of electroacoustic music. Real electroacoustic music introduces new challenges for the PPA, though those same challenges might affect human judgment as well.

A similarity matrix is combined with the PPA in order to find self-similarity in the percussive sounds of electroacoustic music. This percussive similarity matrix is then used to identify structural characteristics in two pieces of electroacoustic music.

Contents

List of Tables

List of Figures

List of Acronyms

AMT	automatic music transcription
CASA	computational auditory scene analysis
DFT	discrete Fourier transform
FFT	fast Fourier transform
GPP	group percussivity profile
IIR	infinite impulse response
IRB	internal review board
MDS	multi-dimensional scaling
MFCC	mel-frequency cepstral coefficients
MIDI	musical instrument digital interface
MIR	music information retrieval
NPP	non-pitched percussive
PCM	pulse-code modulation
PNP	pitched non-percussive
PPA	percussivity-profile algorithm
PSM	percussive similarity matrix
RMS	root mean square
SDPE	single, damped, percussive event
SNR	signal-to-noise ratio
SPL	sound pressure level
SPP	single-value percussivity profile

Chapter 1

Introduction

Electroacoustic music uses electronic technology for the compositional manipulation of sound and is a unique genre of music for many reasons. Many of these reasons will be explored in this dissertation, but for the present, it is sufficient to say that composers of electroacoustic music have fewer limitations than other composers. The traditional music score is rarely an appropriate tool to describe the musical ideas contained in pieces of electroacoustic music. In fact, a recording of a piece of electroacoustic music often is the only objective representation of that piece of music.

When speaking about the analysis of the several pieces of music in *Electroacoustic Music: Analytical Perspectives* [1], Risset [2] states that, "... one can by no means reduce to a blind and automatic dissection according to a priori principles; each work requires its own approach ..." This attitude is appropriate toward the vast majority of electroacoustic music compositions due to the diversity of techniques used in composition. One of the difficulties is "the lack of a written document creates great difficulties for the musicologist who insists on carrying out rigorous, 'objective' work [2]."

Figure 1.1 shows a graphical score [3] of the sixth minute of David Berezan's "Unheard Voices, Ancient Spaces [4]." Time is represented along the horizontal axis, the vertical axis represents the "range of frequency [3]," and contrast or darkness of color represents amplitude. Berezan "made very subjective decisions regarding layout, but strictly adhered to the time line [3]." This graphical score was drawn by the composer after the composition had been created.

Though the percussive content of the piece is not obvious from this graphical score, it does show how far the ideas of electroacoustic music can stray from a traditional music score. The creation of such an analytical representation is time intensive. Commenting

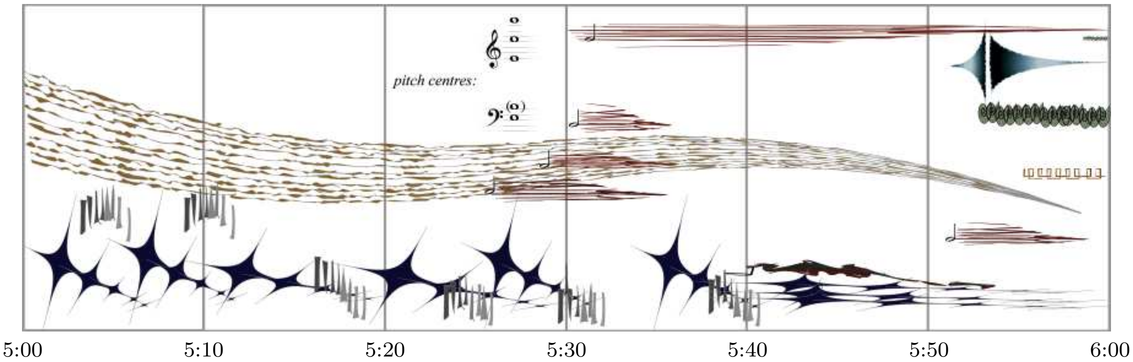


Figure 1.1: A graphical score by David Berezan of the sixth minute of his piece, "Unheard Voices, Ancient Spaces [3]." Time is represented along the horizontal axis, the vertical axis represents the "range of frequency [3]," and contrast or darkness of color represents amplitude. Berezan "made very subjective decisions regarding layout, but strictly adhered to the time line [3]."

on producing the score, Berezan said that he "likely would not undertake such a task just for myself on a regular basis simply owing to the time requirements [3]!"

Clearly, some rudimentary, automated analysis tools would be helpful in at least beginning the analysis of electroacoustic music compositions. Tools that present a visual representation of at least some aspects of the music would be most useful to music analysts. A visual representation could be published, like a traditional music score, in a journal article in order to indicate analytical concepts about the composition. A time-based visual representation also could be scrolled across a screen while a piece of music was being played. This presentation would allow listeners, even first-time listeners, to understand more from their listening experience and might even be useful to those with hearing difficulties.

As is described in later chapters, analyzing electroacoustic music requires special measures. One important measure is that electroacoustic music will at some point need to be analyzed purely as sound without musical foreknowledge. Ironically, one must drop traditional musical expectations in order to analyze the music. It is notable that when analyzing audio as pure sound, the analysis techniques can apply to any sound and not just musical sounds. Analysis of other types of music as well as non-musical sound may benefit from these pure-sound techniques.

The research of this dissertation is intended to be a beginning for an automatic visualization tool set with consideration for the human processing of electroacoustic music and sound. The choice is made to use models of the human auditory periphery as a front end to any analysis algorithms. This choice is made to learn more about how humans process music and sound as well as to imbue the resulting representations of the music with perception-driven qualities. The results of these tools are a set of time-specified properties about a piece of music.

During the initial design phase of the algorithms in this research, it was uncertain exactly which musical and sound qualities would be of interest when analyzing electroacoustic music. Informal discussions with musicians and music listeners led to an array of musical and audio properties that describe an instant of music. These audio properties are arranged by concept level in Table 1.1 from low level (measurable dimension of sound and music) to high level (psychological or musical concepts of sound). These audio properties were considered to be informative about the content when tracking through a piece of electroacoustic music. Some apply only to music, and some apply more directly to pure sound. This research has been previously presented by the author [5].

The properties of percussivity and self-similarity were chosen to be the audio properties explored in this research. These properties are a good compromise between feasibility (computability) and utility. Percussivity is useful because, according to Tanghe et al. [6], "drum events provide important clues about the rhythmical organization of a musical piece" and self-similarity can provide even more of the same clues. After choosing these properties, it became apparent that there was a lack of research concerning which sound-event dimensions humans use as a percussive cue. This deficit motivated the collection of human percussion judgments found in Chapter 3.

The tool set of this research is embodied by the percussivity-profile algorithm (PPA) and the percussive similarity matrix (PSM). The PPA is designed to detect percussivity in pieces of music based on models of human hearing and is tuned to perform similarly to humans at percussion judgment tasks. It, therefore, is able to indicate instants in electroacoustic music that humans would identify as percussive. The results of the PPA are used by another algorithm to generate a PSM that indicates areas of self-similarity in the percussive elements of electroacoustic music.

1.1 Organization of the Dissertation

After sufficient background is provided in Chapter 2 to give context for the current research, the contributions of this dissertation are organized into three chapters. Chapter 3 describes the collection of human percussion judgments. This collection involved first creating stimulus sounds that crossed between percussive and non-percussive, and then

Table 1.1: Possible audio properties of an arbitrary piece of music mentioned by musicians and music listeners in informal discussions. These audio properties are arranged by concept level from low level (measurable dimension of sound and music) to high level (psychological or musical concepts of sound).

Low Level

amplitude	loudness	spectral centroid
brightness	panning	signal-to-noise ratio
harmonics	energy	perceived energy
compression		

Medium Level

percussivity	self-similarity	backwardness
commodulation	noisiness	fundamental frequencies
acoustic sounds	event durations	number of sound sources
harshness	event distinction	vigor
tempo		

High Level

reverberation	event types	predictability
important moments	mood	time signature
number of layers	sound quality	effected sound
conventionality	musical key	

asking participants to make judgments about those sounds. The choices then were collected and processed to discover physical dimensions of the stimulus sounds that indicate percussion. The grouped results of this collection are shown, and these results motivate the design of the algorithm presented in Chapter 4.

Chapter 4 describes the PPA. First, the general algorithm design and specific modeling techniques are described; this then is followed by the full implementation details of the PPA. The tuning of the PPA parameters according to the results of the collection of human percussion judgments follows next. Some practical considerations also are examined, and finally one constructed and two electroacoustic music examples are used to show the capabilities and limitations of the PPA.

Chapter 5 explores the use of the PPA and another tool, the similarity matrix, to find percussive self-similarity in pieces of music. A metric of percussive similarity is presented and used in order to turn the similarity matrix into a PSM. Some practical considerations are explored before the utility of the percussive self-similarity is presented using one constructed example and two electroacoustic music examples.

Chapter 2

Background

The research in this dissertation draws from many different disciplines including human hearing, human percussion judgment, electroacoustic music, music information retrieval (MIR), and musical self-similarity. The following sections are intended to give enough background in these areas to understand the context of the current research. First, a few definitions and clarifications of terms and expressions will be made.

2.1 Definitions

In order to be completely clear about terms and ideas, it will be helpful to establish several definitions. The word *sound* is used in two contexts in this dissertation. The first context corresponds to Webster's [7] definition, "the sensation perceived by the sense of hearing," and is the general idea of sound. The second context corresponds to another of Webster's definitions, "a particular auditory impression," and is what one hears when a solitary event occurs in the physical world. This second meaning of the word *sound* pertains to the physical stimulus and can be used interchangeably with the expression *sound event*, though the latter is used without the ambiguity of the other meanings of *sound.* A *sound dimension* is any directly measurable physical quantity associated with a sound event, such as frequency content, sound pressure level (SPL), and temporal fluctuation.

In order to discuss some of the dimensions of sound events, a simplified amplitude envelope of an isolated sound event [8] is shown in Figure 2.1. The initial rise from no amplitude to the maximum amplitude is the *attack* and the duration of the *attack* is known as the *rise time.* Immediately following the attack, a decrease in amplitude called the *decay* takes place until the amplitude remains constant for a duration. This constant

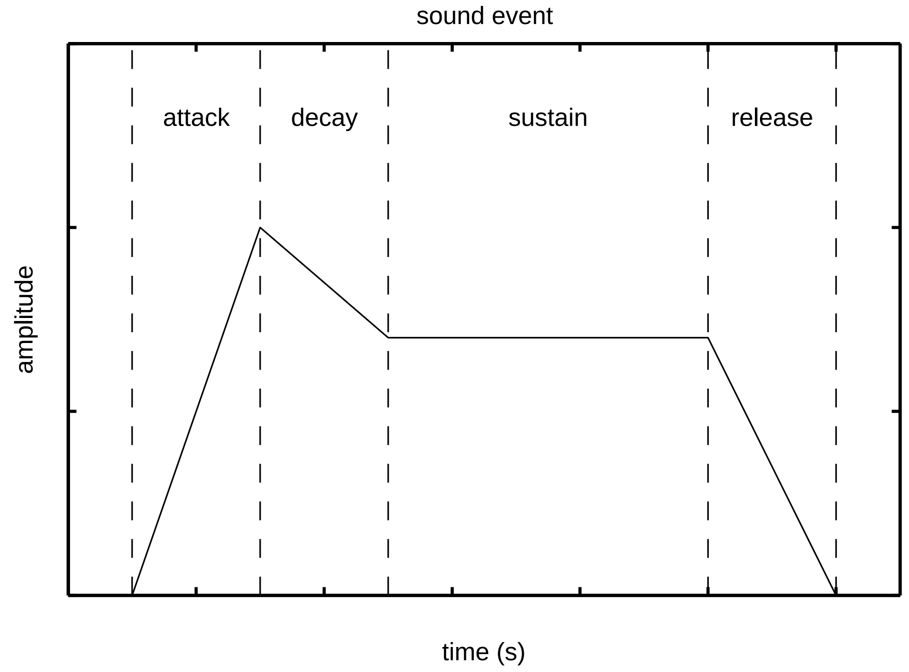


Figure 2.1: The parts of a simplified amplitude envelope of a sound event [8].

amplitude section of the envelope is known as the *sustain* or steady-state portion of the envelope. Finally, the amplitude decrease from the sustain level to no amplitude is known as the *release* and the duration of the *release* is known as the *fall time.*

In the context of this dissertation, the term *percussion* may have one of three meanings: a family of musical instruments, the beating or striking of a musical instrument, or the introduction a sudden pressure change into the air (perhaps explosively). The expression *percussive sound* is defined as any sound that a human would judge to arise from the second or third meaning of percussion. The term *percussivity* is used throughout this dissertation to indicate the quality of being percussive and refers to how percussive a sound or instant would be judged to be. It is worth noting that the definition of *percussion* is a based in the physical domain, and the definitions of *percussive sound* and *percussivity* are based in the psychological domain.

One dimension of a percussive sound which is important in the context of this research is whether it is *pitched.* A pitched percussive sound carries with it a strong sense of pitch or tone, and a non-pitched percussive sound does not. According to Rossing [9], *pitch* is “an attribute of auditory sensation by which sounds may be ordered from low to high."

A new expression is introduced here to encapsulate the idea of a percussive sound while trying to remove preconceptions which might be associated with that notion. In order to be completely clear, the expression *single, damped, percussive event (SDPE)* is defined to be

- a single sound event created by the impact of one object with another without either object breaking (for example, a strike of a xylophone, a hand clap, or a ball bounced against a wall),
- a single sound event created by the direct introduction of a extremely sudden pressure change in the air (for example, a balloon pop, a pistol shot, or a vocal plosive),
- or any synthetic or electronically manipulated sound event which is reminiscent of these (for example, an electronic drum).

The source of a single SPDE may be physically complex but should appear to originate from the same impact or pressure change (example: snare drum). "Damped" in this context signifies that the sound event decreases in amplitude after the initial attack. This definition of an SDPE is in the psychological domain.

2.2 Electroacoustic Music

Emmerson and Smalley [10] offer the following definition of electroacoustic music:

> Music in which electronic technology, now primarily computer-based, is used to access, generate, explore and configure sound materials, and in which loudspeakers are the prime medium of transmission. There are two main genres. Acousmatic music is intended for loudspeaker listening and exists only in recorded tape form (tape, compact disk, computer storage). In live electronic music the technology is used to generate, transform or trigger sounds (or a combination of these) in the act of performance; this may include generating sound with voices and traditional instruments, electroacoustic instruments, or other devices and controls linked to computer-based systems. Both genres depend on loudspeaker transmission, and an electroacoustic work can combine acousmatic and live elements.

The following short history of electroacoustic music is paraphrased from Risset [2] and Emmerson and Smalley [10]. Electroacoustic music has a 60-year history. It primarily originates from compositional techniques, aesthetic approaches, and technological advances developed in Europe, Japan, and the Americas in the 1950s. Early compositions fell into two stylistic categories: *musique concrète* and *elektronische Musik*. In *musique concrète*, the composer modifies and assembles sound recordings to make the musical work as a concrete recording rather than an abstract score. Some of the early composers of *musique concrète* were Pierre Schaeffer, Pierre Henry, Luc Ferrari, Francois Bayle, and Beatriz Ferreyra. In *elektronische Musik*, the composer uses only electronically produced sounds with precisely controlled parameters. Early practitioners of this style included Herbert Eimert, Karlheinz Stockhausen, Gottfried Michael Koenig, and Milton Babbitt. With his piece *Gesang de Jünglinge*, Stockhausen became one of the first composers to combine the two techniques.

In 1957, Max Mathews implemented the first digital computer synthesis of sound at Bell Laboratories, and initiated the move from analog to digital processing for electroacoustic music. This move opened the door to a world of processing that could be implemented with reproducibility and precision. Today nearly all recorded sound and music is created or at least assembled using computers. The techniques pioneered by electroacoustic musicians now appear in other musical genres, soundtracks, sound design for theater, and sound environments for museum exhibitions.

Although electroacoustic music has had an influence on modern music and sound design, and it has a rich history, there has been little critical attention towards it. The lack of an objective representation of the music contributes significantly to the paucity of theoretical writing about electroacoustic music [2]. It certainly does not receive the same analysis and critique as classical music [11]. There have been examples [1], [4], [12], [13] of analyses and visual representations of electroacoustic music, but they are few and far between compared to the analysis of other genres.

One trait of electroacoustic music which increases the difficulty of analyzing it is that composers are not limited to natural sounds. Electroacoustic composers have a larger palette of sounds than composers of music with more traditional instruments. Electroacoustic composers, for example, can create sounds by computer synthesis, tape manipulation, and analog circuitry. These techniques open up not only new sounds, but also new musical gestures as well. Smalley [14] has attempted to create a new musical language to explain and understand this new realm of sounds and musical gesture.

Electroacoustic-music composers also often use the blurry of perception as part of their compositions. An example of this is the repeating of a set of notes faster and faster until the notes become indiscernible and create an even tone. Moss's "Oscillococcinum" [15] uses this technique.

Composers can tamper with the basic ideas of music as part of a composition. A large portion of electroacoustic music certainly eschews traditional ideas of rhythm and tonal structure, but the tampering goes beyond that. For example, an electroacoustic music composer also can deviate from the traditional ideas of instruments, music voice, and performance space. This deviation from traditional notions of music is large enough that some use the term "sonic art" instead of electroacoustic music.

Bossis [16] indicates that human analysis of electroacoustic music is uniquely difficult:

> Analysis of electroacoustic music is particularly arduous due to the complexity of its composition, the constraints imposed by its very nature, the continuity of its timbral dimension, the specificity of the electronic instrumentation, and the difficulties inherent to its performance.

Algorithmic analysis for electroacoustic music certainly will require consideration of the sound outside of traditional expectations of music or even physical generation.

2.3 Human Hearing

In order to explain the models used in the algorithms of this research, the following paragraphs discuss some of the anatomy and physiology of a portion of the human auditory periphery. This explanation is simplified to the level of detail necessary to express the mechanisms of the models used in this dissertation. For a more complete description of the physiology of these mechanisms, see Pickles [17] and Moore [18].

The structures of the auditory periphery can be seen in Figure 2.2. Initially when a sound hits the ear, the pressure variation is guided by the pinna into the ear canal. The pinna acts as an impedance matching mechanism so that the pressure variation can travel the length of the ear canal and strike the eardrum (tympanic membrane). At the eardrum, the pressure variation is transduced into mechanical motion. The ossicles, three small bones named the hammer (malleus), anvil (incus), and stirrup (stapes), transfer and amplify the mechanical force at the eardrum to the oval (vestibular) window. The oval window is a membrane-covered opening which acts as the entrance of the cochlea for sound vibration.

A schematic of the structures of an uncoiled cochlea can be seen in Figure 2.3. The cochlea is a conical cavity roughly 35 mm long, coiled like the inside of an empty snail shell, and divided roughly in half along its length by the cochlear partition. The motion of the oval window sends a fluid wave into the top half of the cochlea, the scala vestibuli. The fluid wave travels the length of the cochlea from base to apex, and then travels through a hole known as the helicotrema. The fluid wave then passes through the bottom half of the cochlea from apex to base in the scala tympani eventually reaching the round window. The round window is another membrane-covered opening to the middle ear, but it is not connected to a bone like the oval window. The fluid wave sets the basilar membrane along the cochlear partition into motion.

A detailed cross-section of the cochlea can be seen in Figure 2.4. The cochlear partition is defined by two membranes, Reissner's membrane and the basilar membrane, and the fluid-filled space in between the membranes. The fluid-filled space is called the scala media and is separated from the scala vestibuli by Reissner's membrane and from the scala tympani by the basilar membrane.

A schematic of the detailed structures on the basilar membrane can be seen in Figure 2.5. A group of cells called the organ of Corti lay along the basilar membrane and contain the receptor cells called hair cells. There are two anatomically and functionally distinct types of hair cells: inner and outer. Inner hair cells are the primary sensory receptors of the auditory system, and are afferently innervated. Outer hair cells appear to act as an amplifier for sounds, generating motion in the ear. Both types of hair cells are located on the top of the organ of Corti, and have stereocilia that extend from the top of the organ of Corti to the tectorial membrane (inside the scala media). The tectorial membrane is a flap of tissue which extends over the organ of Corti into the scala media, but does not fully divide the scala media.

When the fluid wave travels in the cochlea it sets the basilar membrane into transverse motion. This motion causes the tectorial membrane to move with respect to the organ of Corti, which in turn causes the stereocilia on the apical surface of the hair cells to be sheared. Shearing of the stereocilia on an inner hair cell triggers a release of neurotransmitters from the base of the hair cell into the synapses with the primary auditory neurons.

Fluid waves traveling in the cochlea set the basilar membrane into unusual and specific motion. Four instants of this motion over time can be seen in Figure 2.6. Waves caused by high-frequency sound create motion of the basilar membrane near the base of

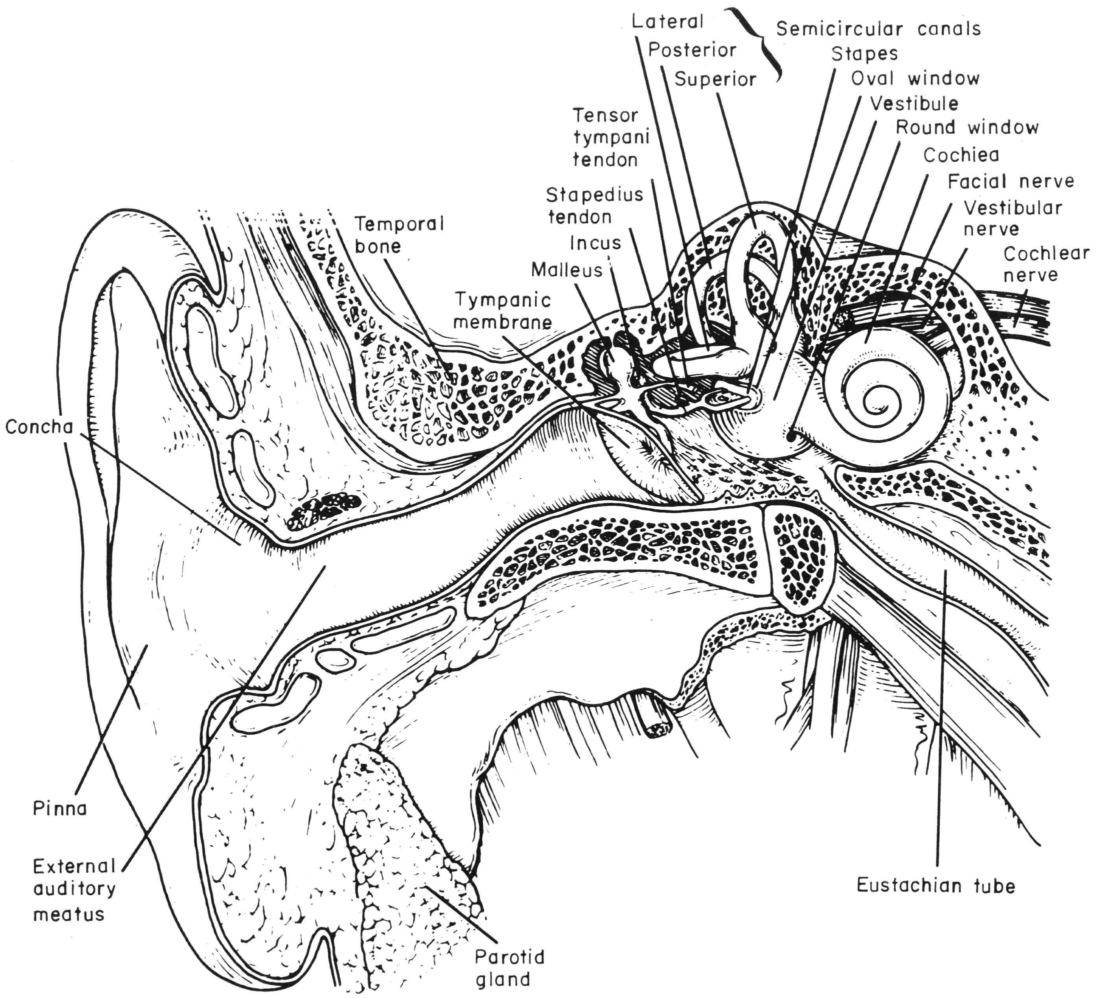


Figure 2.2: The anatomy of the external, middle, and inner ears in humans from Pickles [17] and originally from Kessel and Kardon [19].

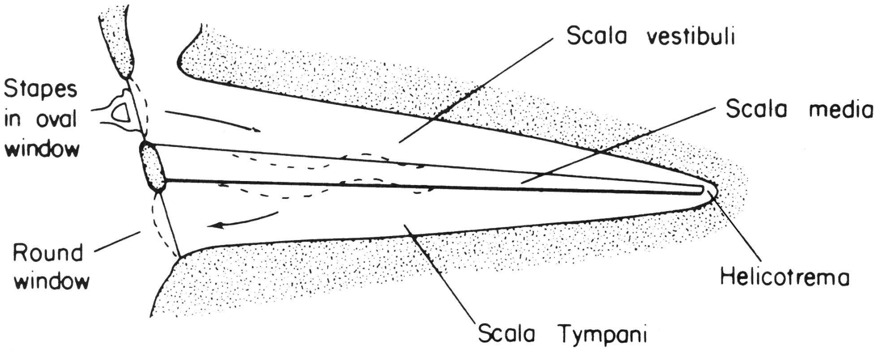


Figure 2.3: A schematic of the uncoiled cochlear duct from Pickles [17] and originally from Ryan and Dallos [20].

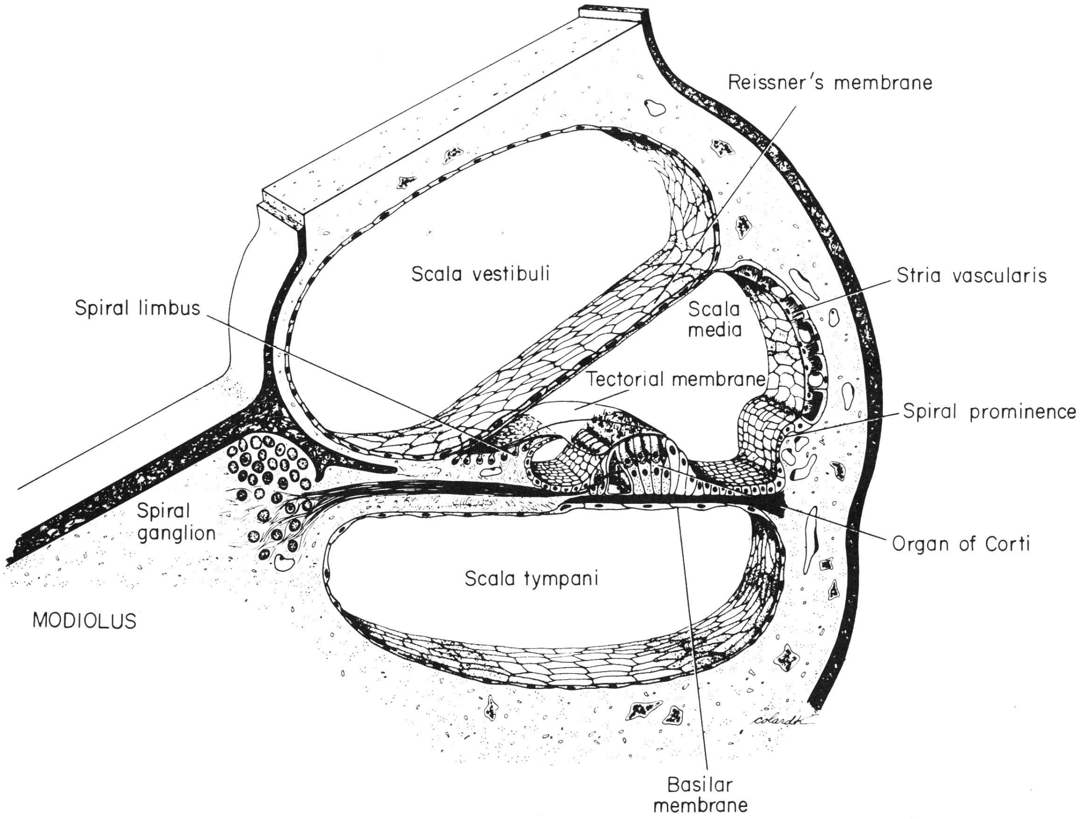


Figure 2.4: A cross-section of the cochlear duct from Pickles [17] and originally from Fawcett [21].

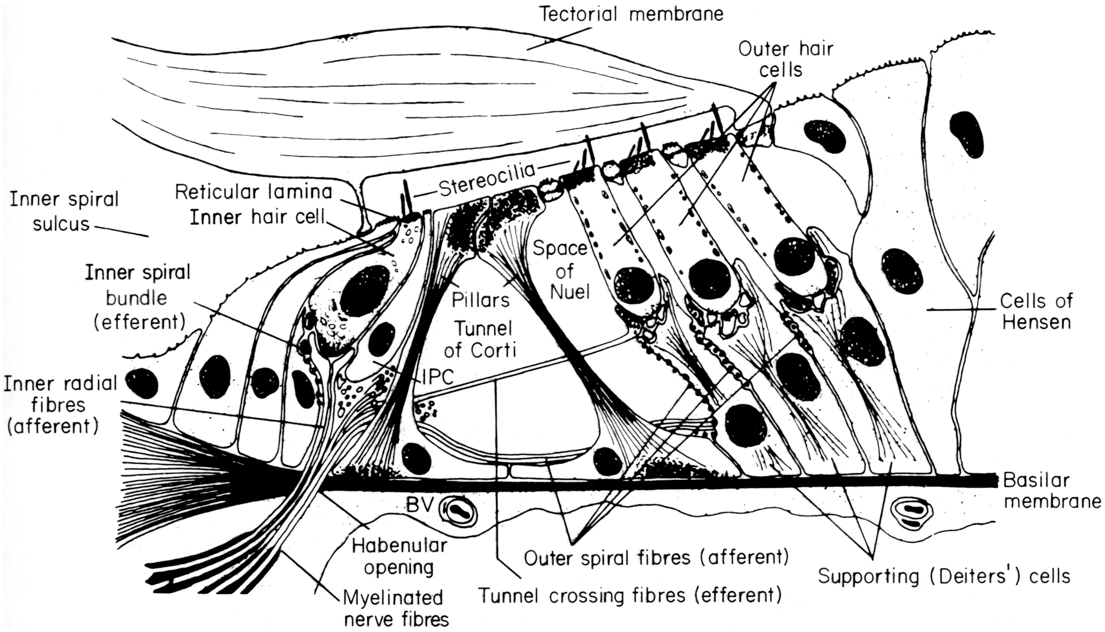


Figure 2.5: A schematic of the detailed structures on the basilar membrane from Pickles [17] and originally from Ryan and Dallos [20].

the cochlea, and low-frequency sound causes motion of the cochlear partition nearer to the apex. The transverse wave traveling along the cochlear partition has a sharp peak of maximum deflection corresponding with the frequency selection displayed by both the basilar membrane and the auditory nerve fibers. This is a first location for possible spectral analysis by the auditory periphery.

Auditory nerve fibers connect to the inner and outer hair cells all along the length of the basilar membrane. The frequency of sound that causes maximum deflection at the location of innervation by an auditory nerve fiber is in concert with the frequency selectivity of the auditory nerve fiber itself. A synapse, a specialized junction through which the cells of the nervous system signal to each other, exists at the interface between the hair cell and the auditory nerve fiber. When an inner hair cell is triggered by the appropriate motion, it releases neurotransmitters into the gap of the synapse (synaptic cleft). If the right combination of neurotransmitters occurs in the synaptic cleft then the auditory nerve fiber will initiate a neural spike, and these neural spikes are transmitted to the brain.

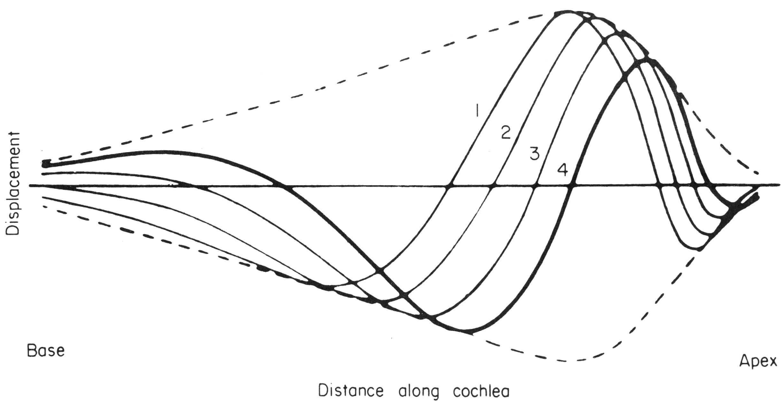


Figure 2.6: Traveling waves in the cochlea showing basilar membrane motion from Pickles [17] and originally from von Békésy [22]. The solid lines show four successive instants in time and the dotted line shows the envelope of motion which remains static throughout constant sound excitation.

2.4 Human Percussion Judgments

In designing an algorithm that can model human percussion judgment, the first step seems obvious: find out what sounds humans judge to be percussive and what dimensions of those sounds affect the judgment of the percussivity of a sound the most. Research into these questions is scarce. Researchers in several different areas simply have made assumptions about what is percussive without experimental evidence. Most music researchers interested in percussive sounds are not interested in sounds which challenge the definition of percussive. They use sounds from the core of the musical category of percussion. This will be discussed further in Section 2.5.3.

Rossing [23] describes the subcategories of the percussive instruments and remarks that “there may be differences of opinion as to whether aerophones and chordophones properly belong in the percussion family.” Aerophones include whistles, sirens, etc., and chordophones include the piano and harpsichord. The unquestioned members of the percussion instruments are the idiophones (xylophone, marimba, chimes, cymbals, gongs, etc.) and membranophones (drums). Rossing writes little about the short amount of time associated with the strike and the resulting attack of the generated percussive sound. From the research of this dissertation, this rise time may be the most important part of a sound in determining its percussivity. He does, however, write extensively about the resulting resonant motion of the different percussive instruments.

The research of Ohta et al. [24] stands out in relation to the current research. They played impulsive sounds ("characterized by very short duration, steep attack, and high sound pressure level") to 20 participants who rated the sounds according to a seven-point scale for 18 different semantic descriptors. Examples of the semantic descriptors are the ranges of "hard to gentle," "powerful to weak," and "reverberant to dead." The sounds came from the categories of "musical instruments, sports, construction, and explosion." Although the physical measures of the sounds did not include rise time, they did include D_{30}, the duration between the maximum sound pressure level, L_{max}, and $L_{max} - 30$ dB. Among other correlations, a large correlation exists between D_{30} and the descriptor "reverberant."

The research of Lakatos [25] is related closely to the measurements needed for the current research. He used a multidimensional scaling (MDS) algorithm to assess the similarity of three groups of sounds: harmonic, percussive, and the combination of both. He found that two of the dimensions which indicate similarity of percussive sounds correlate highly with spectral centroid and logarithm of rise time. He also found that the two dimensions which indicate similarity of the combined-group sounds correlate with logarithm of rise time and spectral centroid as well. This latter analysis, however, did not fully separate percussive sounds from harmonic sounds as significant group overlap occurred.

Other investigations have focused on measuring physical and perceptual properties that can be obtained from MDS of judgments of the striking of objects. McAdams et al. [26] used synthesized sounds of bar instrument impacts to quantify the psychophysical relations between dissimilarity judgments and the varied parameters of the synthesis. Giordano [27] works with participants judging mallet and material hardness directly from the sound of the impact between the two.

In an effort to help percussion detection and percussion classification research, Tanghe et al. [6] collected drum annotation from experienced drummers and percussionists for 49 real-world music excerpts. Their annotations are freely available on the Internet, but their database does not include any representative examples of electroacoustic music.

2.5 Music Information Retrieval

According to Fingerhut [28], in the 1990s the field of music information retrieval (MIR) appeared. It connected many different disciplines for the purpose of retrieving information from music at a higher level than the direct audio signal. That higher level information may be note and sound event characteristics (onset, pitch, duration), instrumentation, performance space characteristics, musical segmentation, dynamics, musical key, tempo, musical similarity to other pieces, or any number of other features. MIR also includes using higher level information to retrieve pieces of music from a large collection. The following MIR disciplines are pertinent to this research: automatic music transcription, onset detection, percussion detection, and percussive sound classification. The intersecting discipline of auditory scene analysis is also of interest.

It is important to note at this point, that most MIR research concerns music which follows a traditional musical score. This adherence to a score carries with it a strong expectation of pulse, beat, and traditional rhythm in the music being analyzed (for example, the work of Klapuri et al. [29] involves determining the musical meter, and the work of Laroche [30] involves tracking the beat and tempo of music). A great portion of the MIR percussion research also concerns percussion sounds that are known ahead of the analysis (for example, the work of Zils et al. [31] involves extracting the energy of drums sounds known a priori, and the work of Sandvold et al. [32] involves extracting kick, snare, and cymbal sounds from polyphonic music). This adherence to a traditional notion of rhythm and foreknowledge of percussion sounds does not integrate well with analysis of electroacoustic music for reasons stated in Section 2.2.

2.5.1 Automatic Music Transcription

Automatic music transcription (AMT) is the process of extracting enough information from a piece of music to create a visual representation of it. The goal of AMT systems has typically been to create a traditional musical score from a recording of a piece of music (for example, the work of Bello et al. [33] involves the evaluation of two AMT systems for transcribing monophonic and simple polyphonic music, and the work of Reis and Vega [34] involves the use of genetic algorithms for AMT to create traditional musical scores from audio). Obviously, any AMT system which works to produce a traditional music score will fail with a large portion of electroacoustic music compositions.

2.5.2 Onset Detection

A discipline related to AMT is onset detection. Onset detection involves determining when new notes or sound events start in a piece of music. This task is often made difficult by confounding sound energy in the spectrum of polyphonic music. Onset detection is generally focused in two realms: percussive sounds and non-percussive sounds. Percussion detection will be discussed in the Section 2.5.3. Non-percussive sounds are generally expected to have longer rise times and contain a significant pitch component.

Following the review work of Bello et al. [35], Collins [36] gives an even more complete review of the different onset detection methods for non-pitched percussive (NPP) and pitched non-percussive (PNP) sound events. He suggests that "the NPP case is effectively solved by fast intensity change discrimination processes, but that stable pitch cues may provide a better tactic for the latter." Although non-pitched non-percussive sound events are important to electroacoustic music, the review work of Bello et al. does not discuss them.

Dixon [37] extends the evaluation work of Collins by including a new set of algorithms and adding a significantly larger data set of piano notes. Some of Dixon's results contradict previously published results and suggest that "a similarly high level of performance can be obtained with a magnitude-based (spectral flux), a phase-based (weighted phase deviation), or a complex domain (complex difference) onset detection function." He also concludes that some of the algorithms are sensitive to implementation details or parameter settings.

You and Dannenberg [38] work to improve onset detection by using machine learning. A major issue with machine learning is that a library of training data must be available. You and Dannenberg work to solve this issue by training a machine-learning onset-detection algorithm by using musical instrument digital interface (MIDI) scores of orchestral music alongside the digital audio recordings. Problems arise with the accuracy of the score-to-audio alignment that they handle with semi-supervised and bootstrapping techniques. These techniques are used to iteratively refine the onset detection functions and the data used to train the functions. Their machine learning adaptations do improve the performance of a general purpose onset detection algorithm for use with orchestral music.

In order to deal with the problem of aligning the score and audio file, a second audio file is generated from the MIDI score. All three files are used to generate the onset times for the training data. The results of their work indicate that a machine-learning algorithm can be trained to perform well at onset detection given a large enough training set for a particular subclass of music.

2.5.3 Percussion Detection

Percussion detection involves the marking of the onset times of percussive sounds in music. The vast majority of this research is focused on sounds which are known ahead of time or are very limited in scope compared to sounds one might find in electroacoustic music. For example, FitzGerald's [39] work involves "polyphonic percussion transcription and sound source separation of a limited set of drum instruments, namely the drums found in the standard rock/pop drum kit."

A remarkable exception to the foreknowledge and limited-scope research is the work by Uhle et al. [40]. They use independent component analysis techniques to focus solely on some dimensions that they claim cause a sound to be percussive. The dimensions they chose are measures of percussiveness, dissonance, spectral flatness, and noise-likeness. These dimensions, although they may be appropriate, are not specifically motivated from experimental understanding of which dimensions make a sound percussive. Their results indicate as much as 95% accuracy at finding percussive sounds using single dimensions, but a false positive rate of as much as 71% as well. Nesbitt, Hollenberg, and Senyard's [41] work makes use of Uhle et al.'s work to transcribe Australian aboriginal music with some success.

Beat/rhythm detection generally uses the information from onset detection and percussion detection to try to establish the time signature and/or tempo of a piece of music. Gouyon et al. [42] compiles many different current algorithms for tempo induction and gives a performance comparison. As mentioned before, the work of Klapuri et al. [29] involves the determination of musical meter. Scheirer's work [43] also includes an autocorrelation mechanism for finding the tempo of a piece of music from the psychoacoustically-processed, unsegmented digital audio.

Percussion sound classification follows from percussion detection as well and concerns classifying a particular sound as one of a set of previously known percussion sounds or sound groups. The work of Van Steelant et al. [44] involves the use of support vector machines for discerning bass and snare onsets in music where the two may overlap. The work of Gouyon et al. [45] is another example of percussion classification, but this work uses zero-crossing rate to classify bass and snare sounds in popular music.

2.5.4 Auditory Scene Analysis

Auditory Scene Analysis by Bregman [46] describes an approach to understanding how humans organize sound. Through a huge number of examples and experiments, Bregman proposes the ideas of integrating and segregating the complex spectrum arriving at the eardrums into auditory streams associated with sound sources. A sound source is often the physical origin of a particular sound, but also could be a conglomerate of physical sources, for example, a violin section of an orchestra all playing the same note. An auditory stream is an individual sound source or group of sources linked together over time. An example of this is the “cocktail party effect,” (a description is given by Arons [47]) where a person at a cocktail party can listen to another individual in a conversation while other conversations are happening nearby. (A different phenomenon known also as the “cocktail party effect” is described by Pierce [48].)

Computational auditory scene analysis (CASA) is the field of trying to make computers perform auditory scene analysis and is a discipline which intersects with MIR. Much of CASA involves speech processing as in the initial work of Brown and Cooke [49], which explores using CASA in order to separate speech from background noise including other speech. Ellis [50] explores the idea of a prediction-driven CASA system that guesses at which energy in the total spectrum is due to particular sound sources. Ellis's prediction-driven system is an attempt to avoid the pitfalls of a data-driven approach that would always evaluate a particular sound in the same way, regardless of context. Ellis's main example is street noise, a sound mixture that is found in examples of electroacoustic music. Goto [51] extensively describes the application of CASA to traditional musical audio signals (though his description includes musical expectations that may be inappropriate for electroacoustic music).

It is important to mention CASA because, although it is not the specific path followed by this dissertation's research, CASA does offer a significant advantage over other methods for analyzing electroacoustic music. CASA is psychoacoustically motivated, and works toward stream separation without requiring a traditional notion of musical rhythm or melody.

2.5.5 Processing Choices

According to Scheirer [43], most engineering approaches to music analysis depend on some time-frequency representation of the sound. Some examples of this representation are discrete Fourier transform (DFT) (example: Moreau and Flexer [52]), “constant-Q” filters (example: Brown and Puckett [53]), wavelets (example: Kronland-Martinet and Grossman [54]), or psychoacoustic models (example: Meddis and O'Mard [55]). The choice for the time-frequency representation has an effect on what information can be retrieved from the music, and therefore the usefulness of a particular representation for a particular application. For example, the psychoacoustic models are generally inappropriate for an analysis system which depends on resynthesizing sounds from the original signal due to information loss inherent in the psychoacoustic filtering.

Another decision to make when designing algorithms for MIR is whether the system will attempt to separate the sound field into sound sources as defined in auditory scene analysis. Unseparated processing takes the audio signal as a whole and works to extract information without separation into sources. Scheirer [43] calls this approach “top-down” processing and uses it when estimating the tempo of real-world music examples. Separated processing first separates the audio signal into sources or notes, and then attempts to reconstruct information based on the information. CASA is an example of a separated-processing approach. The approach of the current research is unseparated processing, and is focused on the initial processing of the entire sound field by the auditory periphery without source separation.

In designing a separated-processing approach, another decision must be made about the algorithm. Is it analysis-only or analysis-resynthesis? Analysis-resynthesis allows for the reconstruction of the different sources in the total sound field. This can be used to reconstruct part of the sound field for better intelligibility (for example, the speech work of Brown and Cooke [49]) or further analysis (for example, the work of Zils et al. [31] involves resynthesis of individual drum sounds from the original audio for percussion classification).

When designing a music-analysis system, the decision also must be made whether to model human psychoacoustic behavior. It has been argued [56] that building psychoacoustic prototypes can lead to a better understanding of how humans process sound and pieces of music. If an algorithm tries to model human psychoacoustic behavior, the successes and failures of that algorithm can lead to new psychophysical models.

The above stands in contrast to a more mathematical or opaque approach to the algorithms. For example, Kostek [8] outlines many examples of applying neural nets to music information retrieval. These are examples of algorithms that can produce excellent results, but are less intuitive when trying to understand the results in terms of human psychoacoustic processes. It was the desire for this understanding that motivated the current research to use a more psychoacoustic approach.

2.6 Musical Self-Similarity

Self-similarity in the context of the current research refers to a property of a piece of music. A piece of music is said to have high self-similarity if there are many sections of the piece which are similar to other sections. Foote [57] created a useful tool when he used a similarity matrix based on mel-frequency cepstral coefficients (MFCCs) to show the similarity of short time windows of a piece of music with every other short time window. A similarity matrix already had been used in other fields (for example, the work of Church and Helfman [58] involves using similarity matrices to visualize similarity in text), but Foote brought the similarity matrix to music.

Foote's similarity matrix is a starting point for many other projects, such as audio thumbnailing [59] and audio segmentation [60]. Audio thumbnailing is a process of creating a small representation of a piece of music that would be part of an easily searchable database. Foote and Cooper [61] extend the usefulness of the similarity matrix by using it to find a beat spectrum of a piece of music, and later to segment audio [62].

A competing indicator of self-similarity is given by Dannenberg and Hu [63], in which they show several different techniques of music segmentation as well as one mechanism for displaying the relationship between segments. That mechanism uses differently shaded bars under the waveform graph sometimes with a piano roll notation as well.

Chai [64] uses a mixture of both self-similarity indication methods described above (similarity matrix and shaded bars) in his research on automatic music segmentation, summarization, and classification. He uses a similarity matrix along with approximate pattern matching for analyzing recurrent structural analysis. He then uses colored bars similar to Dannenberg and Hu's shaded bars to display similar segments of the analyzed music.

2.7 Concluding Remarks

Sufficient background now has been given in order to put into context the research presented in the next chapters. Electroacoustic music has some unique traits which require that it be given special treatment when analyzing it. In order to understand the models of the human auditory periphery used in this research, the physiological process of translating pressure variations at the ear into neural spikes was presented. Little research exists in the area of human percussion judgments, although a few articles are relevant. Music information retrieval is an active and broad research field that can be applied with some caveats to electroacoustic music. One tool for displaying self-similarity in music, the similarity matrix, stands out above the rest. Chapter 3 discusses the collection of more data in the field of human percussion judgment.

Chapter 3

Collection of Human Percussion Judgments

In order to design the algorithm for percussion detection, the dimensions of a sound event that define it as a percussive sound need to be known. As shown in Section 2.4, few articles exist in the literature concerning the specific dimensions determining the percussivity of a sound event. Data needed to be collected concerning which sound event dimensions determine percussivity according to human judgment.

This chapter will describe how human judgments about the percussivity of sound events were collected. The methodology section will describe choices made concerning the specific stimulus sounds and how collection of percussion judgments was achieved. The results will then be presented in several different forms along with their statistical validity. Finally some concluding remarks will be made about how the results showed which dimensions of the stimulus sounds could be used as cues to human judgment of a sound's percussivity. A summary of the research in this chapter has been previously presented by the author [65].

3.1 Methodology

In order to collect the human judgments about percussion, a collection methodology needed to be created. The following sections will describe the stimuli used as well as the technique collection techniques follow the suggestions of Martins [66].

3.1.1 Stimuli

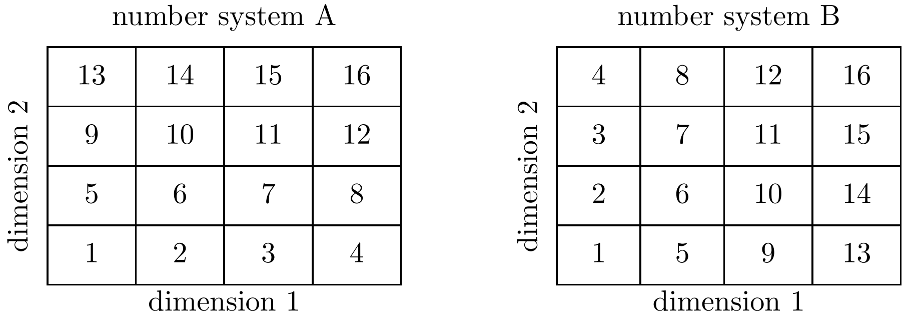
The base for the stimulus sounds for judgment collection was the sound of a snare drum synthesized using the program Csound [67]. A snare-drum sound was chosen because an unmanipulated snare-drum sound is generally accepted as a percussive sound and using a synthesized sound permitted the manipulation of many dimensions of the sound. The snare-drum synthesizer consists of a complex harmonic tone generator and a white noise generator that are gated and filtered both separately and together. The stimulus sounds were synthesized with many different dimension changes to the base sound that were chosen to explore each dimension's effect on the sound's percussivity.

All of the synthesized sounds retained the following properties: 100 milliseconds of silence preceded the sound, the sound lasted between 360–420 milliseconds, and the sound was followed by at least 200 milliseconds of silence. The output of the synthesis was a monophonic, 44.1 kHz, 16 bit WAV audio file, which then was used in the data-collection procedure described in Section 3.1.2.

Choices about which dimensions of sound to explore needed to be made. Based on discussions with peers, references [25] [40], and pilot work, the dimensions of rise time (see Section 2.1), tonal content, and gross spectral filtering were chosen initially for study. Tonal content was manipulated by two methods. The simpler method was to vary the percentages of the total synthesized sound from the tone generator and the noise generator. The sum of the two percentages was always 100%. This method was called "noise percentage." The other method, called "string resonance," was to use a Csound string resonator module (*streson*) to impose a tonal character on the noise by repeatedly echoing with feedback at a fixed time delay. This latter method creates harmonic character in the spectrum of the filtered sound with harmonics peaks at $\frac{n}{T}$ Hz where n is a positive integer and T is the time delay (see the description of comb filters in Dodge and Jerse [68]).

Three different stimulus sets were generated and each set consisted of 16 different sounds. Within each stimulus set, the sounds were arranged in a four-by-four grid and varied along two dimensions. Dimension 1 was varied along one axis and dimension 2 was varied along the other axis. Two different numbering systems were used for the stimulus sounds, as shown in Figure 3.1. Number system A varies dimension 1 locally and dimension 2 globally. Number system B varies dimension 1 globally and dimension 2 locally.

The dimensions for the three stimulus sets are presented in Table 3.1. Figure 3.2



number system A

dimension 2	13	14	15	16
	9	10	11	12
	5	6	7	8
	1	2	3	4
	dimension 1			

number system B

dimension 2	4	8	12	16
	3	7	11	15
	2	6	10	14
	1	5	9	13
	dimension 1			

Figure 3.1: The two numbering systems that identify the stimulus sounds. Number system A varies dimension 1 locally and dimension 2 globally. Number system B varies dimension 1 globally and dimension 2 locally.

Table 3.1: Dimension choices and ranges for the three sets of stimulus sounds.

stimulus set	**dimension 1**	**dimension 2**
A	rise time (10–70 ms)	string resonance (20–80%)
B	gross spectral filtering (low–high)	noise percentage (20–80%)
C	rise time (10–70 ms)	gross spectral filtering (low–high)

shows the time and frequency domain plots of three example stimulus sounds and more detailed descriptions of those stimuli follow.

For the stimuli in stimulus set A, the dimensions of rise time and string resonance were varied. Rise time was varied using the values of 10, 30, 50, and 70 milliseconds. String resonance was varied using the values of 20, 40, 60, and 80%, with the original signal automatically filling the remainder. The noise generator of the snare-drum synthesizer was fixed at a 100% level, which automatically sets the tone generator level to 0%. The example sound "set A, stimulus 1" in Figure 3.2 shows the stimulus sound with rise time set to 10 milliseconds and string resonance set to 20%.

For the stimuli in stimulus set B, the dimensions of noise percentage and gross spectral filtering were varied. The noise percentage was varied using the values of 20, 40, 60, and 80%. The gross spectral filtering was achieved by using different sets of low-pass and high-pass filters to change the spectral content of the stimuli. These filters remained constant throughout the length of the stimuli. The four implemented filter

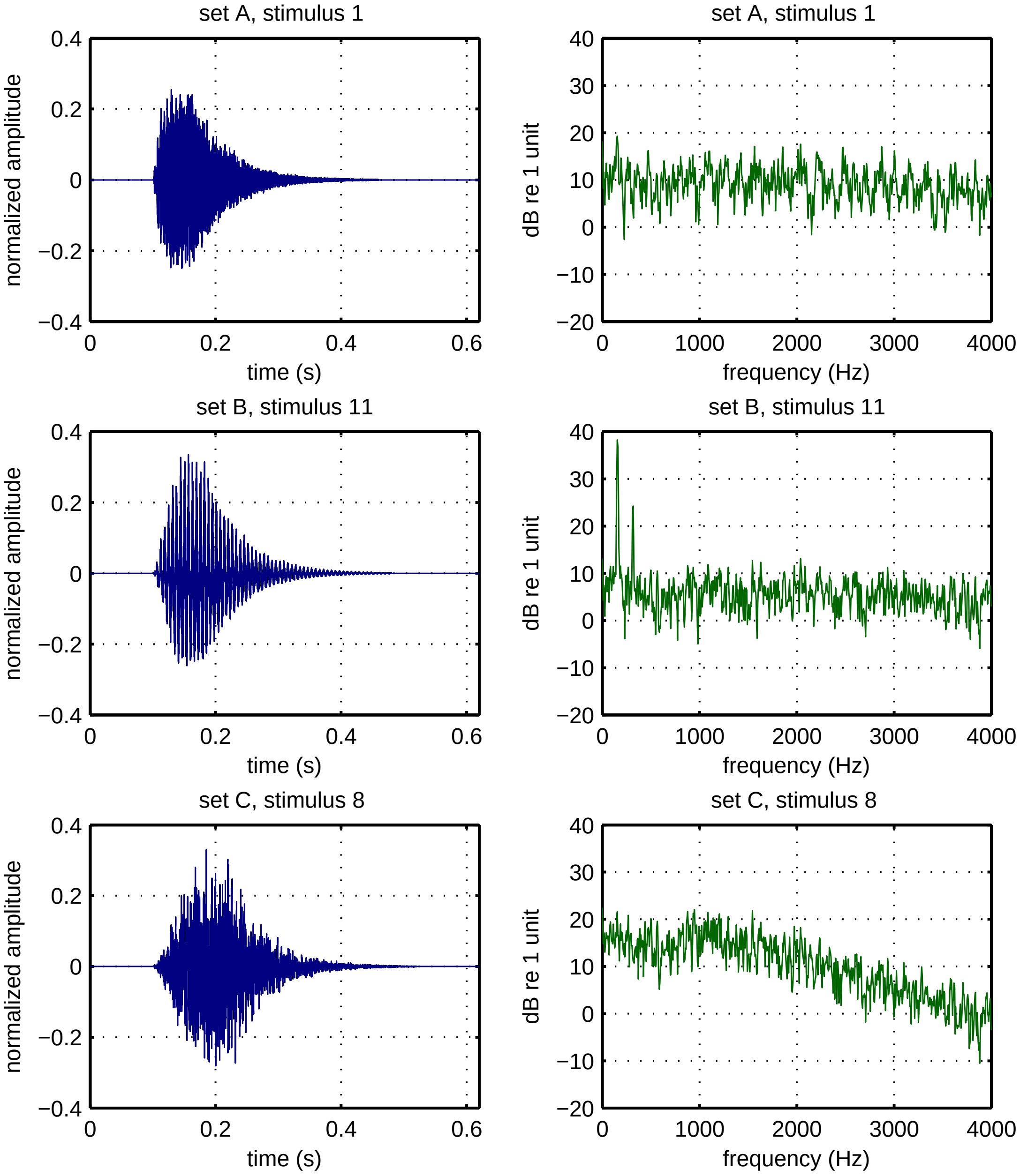


Figure 3.2: Amplitude (left) and spectral (right) plots of three examples of the stimulus sounds: set A stimulus 1 (top row), set B stimulus 11 (middle row), and set C stimulus 8 (bottom row). Stimulus 1 of set A is a mostly unmodified snare sound with a 10 ms rise time. Stimulus 11 of set B displays a low noise percentage, so the first two harmonics are visible in the spectrum. Stimulus 8 of set C is filtered with a single low-pass filter, which is visible in the spectrum.

Table 3.2: The four filter sets used for gross spectral filtering.

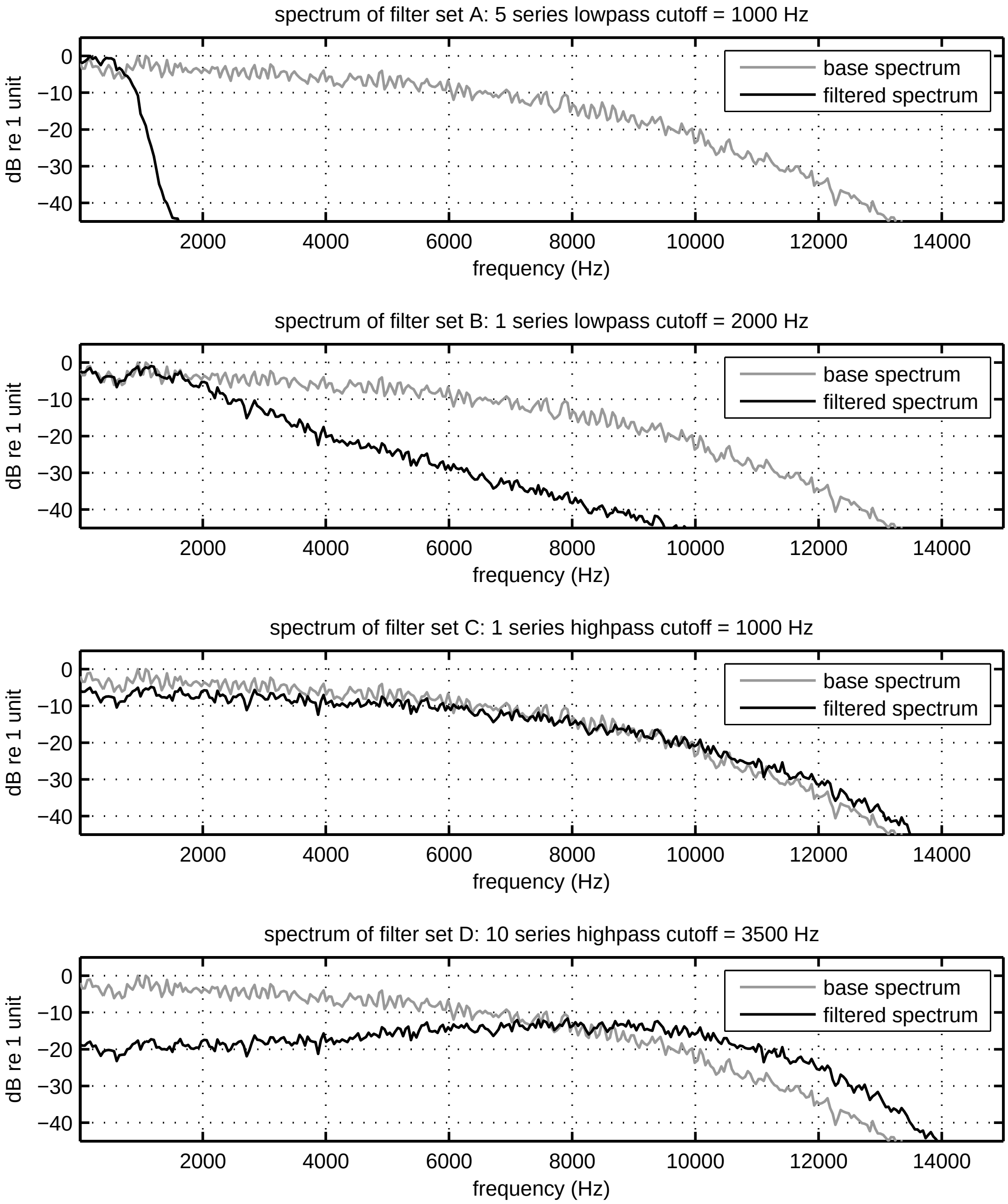
filter set	**filters**
A	5 serially-chained, first-order low-pass filters with a 1000 Hz cutoff
B	1 first-order low-pass filter with a 2000 Hz cutoff
C	1 first-order high-pass filter with a 1000 Hz cutoff
D	10 serially-chained, first-order high-pass filters with a 3500 Hz cutoff

sets are described in Table 3.2, and Figure 3.3 shows the spectrum of the base stimulus sound in the top graph and the effects of the gross spectral filtering in the bottom graph. Applying the gross spectral filtering reduced apparent loudness of the stimulus sounds, so the sound pressure level (SPL) was adjusted to present approximately equal loudness to the participant. The choices of filter parameters were based on pilot work, and were implemented using a Csound filter module (*resonx*). For stimulus set B, rise time was fixed at 40 milliseconds. The example sound "set B, stimulus 11" in Figure 3.2 shows the stimulus sound with a noise percentage of 60% and global spectral filtering implemented with filter set C; the first two harmonics are visible in the spectrum of this sound.

In order to assess whether interaction occurs between rise time and gross spectral filtering, stimulus set C was created in which the dimensions of rise time and gross spectral filtering were varied (if an interaction occurs, then the effects of varying the two dimensions are not simply additive). The rise time was varied as in stimulus set A (10–70 milliseconds), and the gross spectral filtering was varied as in stimulus set B (low–high). The noise generator of the snare-drum synthesizer again was fixed at a 100% level, which automatically sets the tone generator level to 0%. The Csound orchestra and score source code for stimulus set C can be seen in Appendix A. The example sound "set C, stimulus 8" in Figure 3.2 shows the stimulus sound with a rise time of 70 milliseconds and global spectral filtering implemented with filter set B.

3.1.2 Percussion-Judgment Collection

Percussion judgments were collected using a Linux-based desktop computer with a Sound Blaster Live audio controller (model# CD4830), MATLAB 7.0 (R14) software, and a set of Sony MDR-7506 closed-ear, "can"-style headphones. For reference, the output of the pulse-code modulation (PCM) stream and the master volume of the audio controller


spectrum of filter set A: 5 series lowpass cutoff = 1000 Hz

spectrum of filter set B: 1 series lowpass cutoff = 2000 Hz

spectrum of filter set C: 1 series highpass cutoff = 1000 Hz

spectrum of filter set D: 10 series highpass cutoff = 3500 Hz

Figure 3.3: Spectral graphs showing the gross spectral filtering of the different sets of filters. In each graph, the spectrum of the base sound is shown in grey and the spectrum of the base sound passed through the filter set specified in the graph's title is shown in black.

were set to 80% with no equalization or signal processing enabled, which resulted in SPLs of 71–83 dBA (re 20 μPa) presented at the ear for the stimulus sounds. Due to a move, the collection equipment was housed in two different acoustics laboratories at The University of Texas at Austin during the judgment collection: first in room 630 of the Engineering Science building, and then in room 4.156 of the Engineering Teaching Center. The conditions and equipment provided a high signal-to-noise ratio (approximately 95 dB for the sound card, and approximately 84 dB for the sound files) and did not introduce corruption due to electrical or acoustic noise.

For the collection of percussion judgments, a total of 29 participants came to the collection location for one, two, or three sessions in order to complete judgment collections. The participants were not selected for musical ability, and stated that they believed that they had normal hearing for someone of their age. The collection session required about 30 minutes for each stimulus set. Each participant was given $10 compensation for their time and effort. This was all done in compliance with the procedures for human research at The University of Texas and received internal review board (IRB) approval (protocol# 2004-08-0083).

For each stimulus set, the participants were presented with 150 pairs of two stimuli. According to the binomial coefficient equation [69]

$$\begin{pmatrix} n \\ k \end{pmatrix} = \frac{n!}{k!(n-k)!} \tag{3.1}$$

which reduces to the following for the case of $k = 2$ (for pairs out of n objects),

$$\begin{pmatrix} n \\ 2 \end{pmatrix} = \frac{(n)(n-1)}{2} \tag{3.2}$$

120 pairs of two stimuli are possible for $n = 16$. The first 30 pairs presented were training pairs randomly selected from the 120 possible pairs and the participant's choices for the training pairs were not recorded. The next 120 pairs represented all possible pairs and were presented in a different random order for each participant. For each pair of stimuli, the order of the two stimuli was randomly determined.

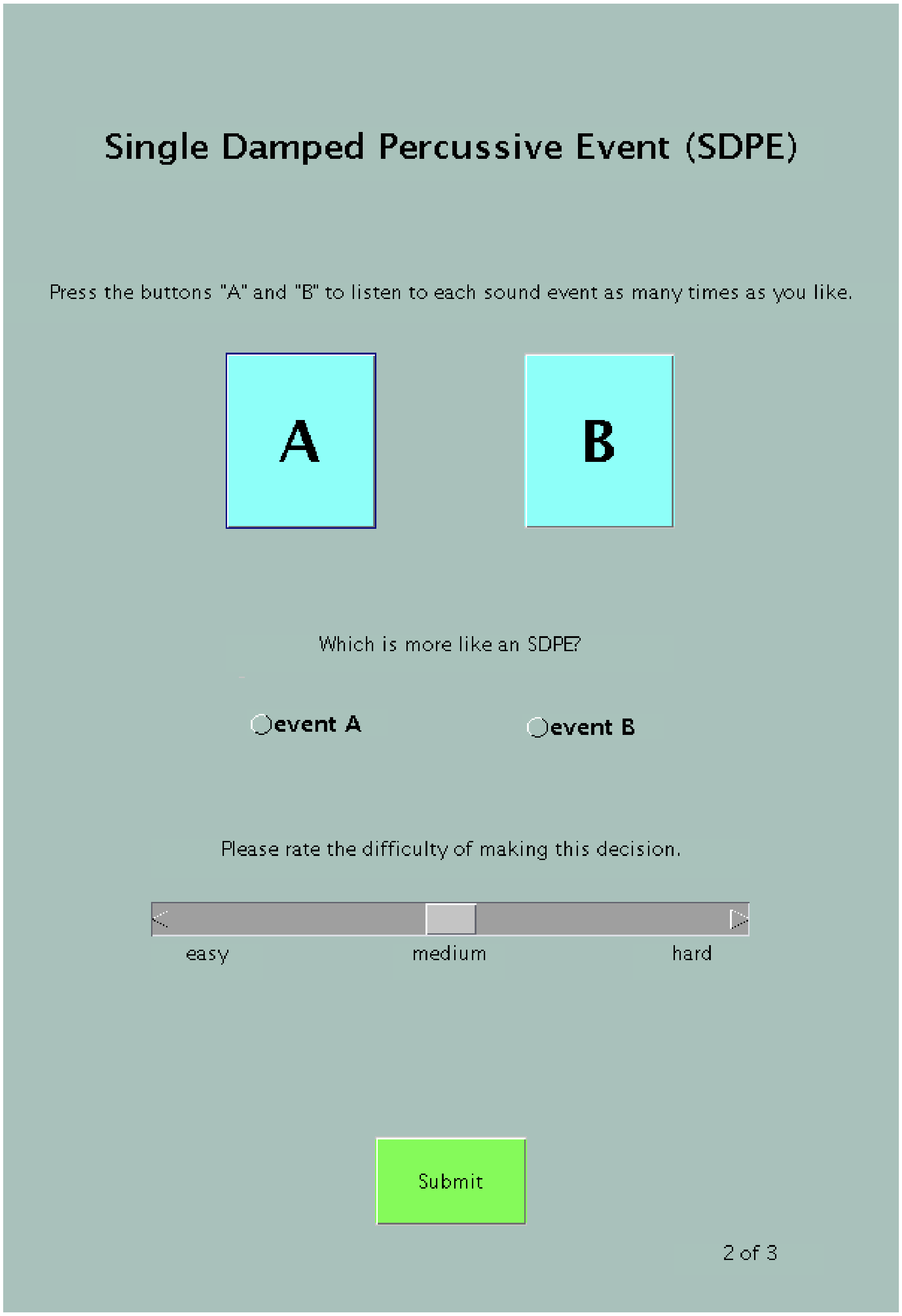
For each presentation of a stimulus pair, the participants were presented with a pair-choice window (see Figure 3.4) that allowed them to listen to each stimulus as many times as they liked, to choose the stimulus they judged to be more like an SDPE, to specify how difficult that judgment was to make, and finally to submit their choices. If the participant failed to choose a stimulus as more like an SDPE, the same pair of stimuli was presented again immediately in the pair-choice window. For each pair of stimuli, the following data were collected: the number of times the participant listened to each pair stimulus, which stimulus the participant judged to be more like an SDPE, and the difficultly rating the participant reported in making the judgment.

In order to calculate a ranked list of the 16 stimuli, a round-robin tournament algorithm was used. Each stimulus "competed" once against every other stimulus in the set of 16 with the "winner" of each "match" being determined by the participant's choice. The "*winstrength*" was determined by the difficulty rating given to the judgment by the participant; purely easy had a *winstrength* of 1 and purely difficult had a *winstrength* of 0.5. The winner of the match received *winstrength* points and the loser of the match received $1 - winstrength$. At the end of the stimulus set, the points awarded to each stimulus were summed and that sum was sorted to determine the ranks of the stimuli in a final ranked list.

That ranked list was then presented to the participant (see Figure 3.5). The list was ordered from "most like an SDPE" to "least like an SDPE," and the participant could listen to each stimulus in the list as many times as they liked. The participant was asked to choose the first stimulus from the list that was no longer an SDPE in their judgment. This threshold value was stored and the collection session then ended for the stimulus set.

3.2 Results

In the rank graphs that follow (Figures 3.6–3.11 and 3.13–3.18), the "global" dimension is the dimension that changes once every four stimulus numbers according to the numbering system shown in Section 3.1.1 and the "local" dimension value changes for each stimulus number but cycles four times throughout the stimulus numbers. The left column presents dimension 2 (from Table 3.1) as the global dimension, and the right column presents dimension 1 as the local dimension. The top two graphs show all 16 stimuli according to the two different numbering systems and their normalized mean rank in the final, ranked lists generated by the round-robin tournament algorithm. The bottom two graphs show means over the local dimensions and therefore the effect of the global dimension on judgment of percussivity of a stimulus sound. Each local-dimension mean represents the mean across one value step of the global dimension. A higher rank value indicates "more like an SDPE."



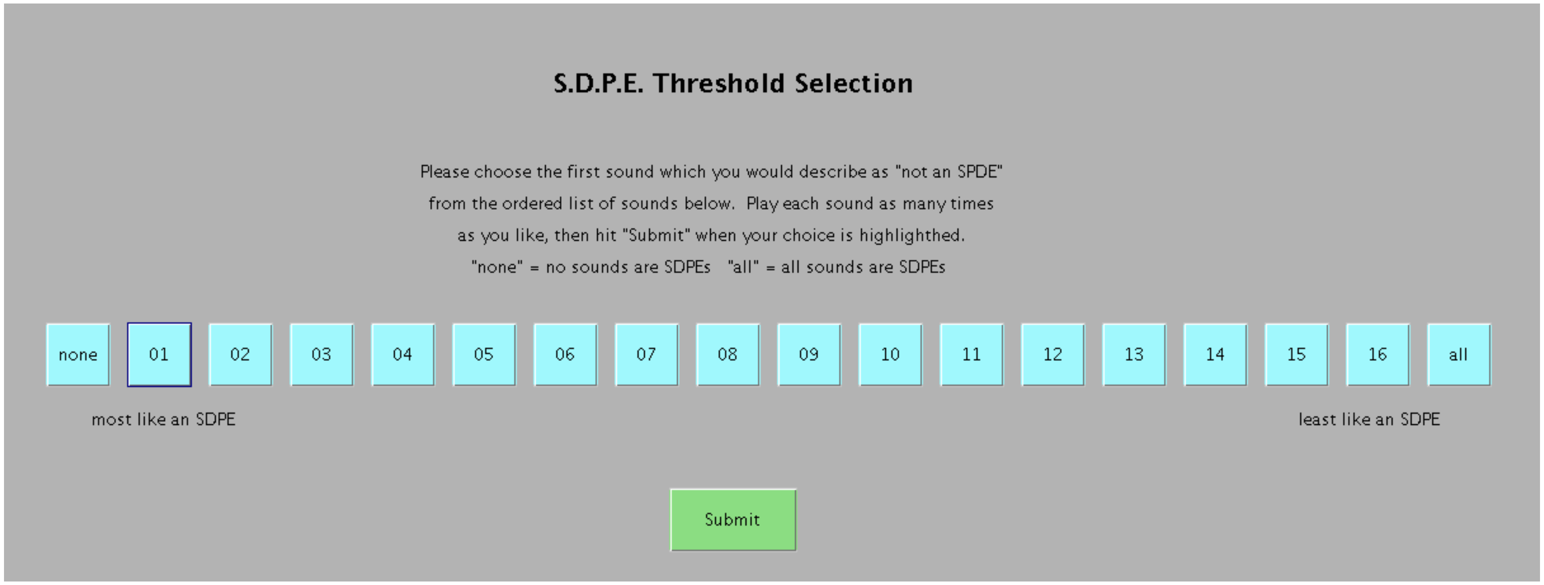
Single Damped Percussive Event (SDPE)

Press the buttons "A" and "B" to listen to each sound event as many times as you like.

Which is more like an SDPE?

Please rate the difficulty of making this decision.

Figure 3.4: The user interface for the pair choices. The user can play both sounds as many times as desired, make the stimulus choice, and rate the difficulty of the decision before clicking the "Submit" button.



S.D.P.E. Threshold Selection

Please choose the first sound which you would describe as "not an SPDE" from the ordered list of sounds below. Play each sound as many times as you like, then hit "Submit" when your choice is highlighted.

"none" = no sounds are SDPEs "all" = all sounds are SDPEs

Figure 3.5: The user interface for the threshold choice. The participant is presented with the 16 stimulus sounds ranked according to that participant's pair choices and asked to choose the first stimulus sound which is “not an SDPE.”

3.2.1 Mean Ranks

Figure 3.6 shows the mean results for stimulus set A (26 participants). For stimulus set A, the results show that, as string resonance increased, the perceived percussivity decreased. This is evident because of the monotonic decrease in mean rank for the local-dimension means in the bottom left graph. The results also show that for the values given, as rise time increased, the perceived percussivity decreased, again shown by a monotonic decrease in the local-dimension means in the bottom right graph. The larger delta ($highest - lowest$) value for rise time indicates that for the values given, it is a stronger cue for percussivity than string resonance.

Figure 3.7 shows the mean results for stimulus set B (27 participants). For stimulus set B, the results show that noise percentage has no consistent effect on the perceived percussivity, which is shown by the almost constant value of the local-dimension means in the bottom left graph. It is notable that for individual participants, the noise percentage was a cue, a negative cue, or no cue for percussivity, but for the entire group of participants, no consistent effect was seen in the displayed means. Gross spectral filtering also had little consistent effect even for individual participants except filter set D, which consistently displayed lower SDPE-likeness ratings. This was shown by the single low value of the local-dimension mean associated with filter set D.

Figure 3.8 shows the mean results for stimulus set C (24 participants). For stimulus set C, the results show that gross spectral filtering had little consistent effect, which is shown by almost constant values of the local-dimension means in the bottom left graph. The effect of gross spectral filtering was lessened in comparison to stimulus set B. As with stimulus set A, the results also show that, as rise time increased, the perceived percussivity decreased, which is shown by a monotonic decrease in the local-dimension means in the bottom right graph.

3.2.2 Mode Ranks

Another method of analyzing the results of the percussion-judgment collection using the modal choices are shown in Figures 3.9–3.11 for, respectively, stimulus sets A, B, and C. For this method, the modal choice (most common choice) for each sound stimulus pair was determined, and then ranks were assigned based on a round-robin tournament algorithm using those choices. A difficulty rating of maximum ease was assigned for any instance where the modal choice was clear, and maximum difficulty was assigned for any instance where the number of participants choosing each sound stimulus was exactly equal.

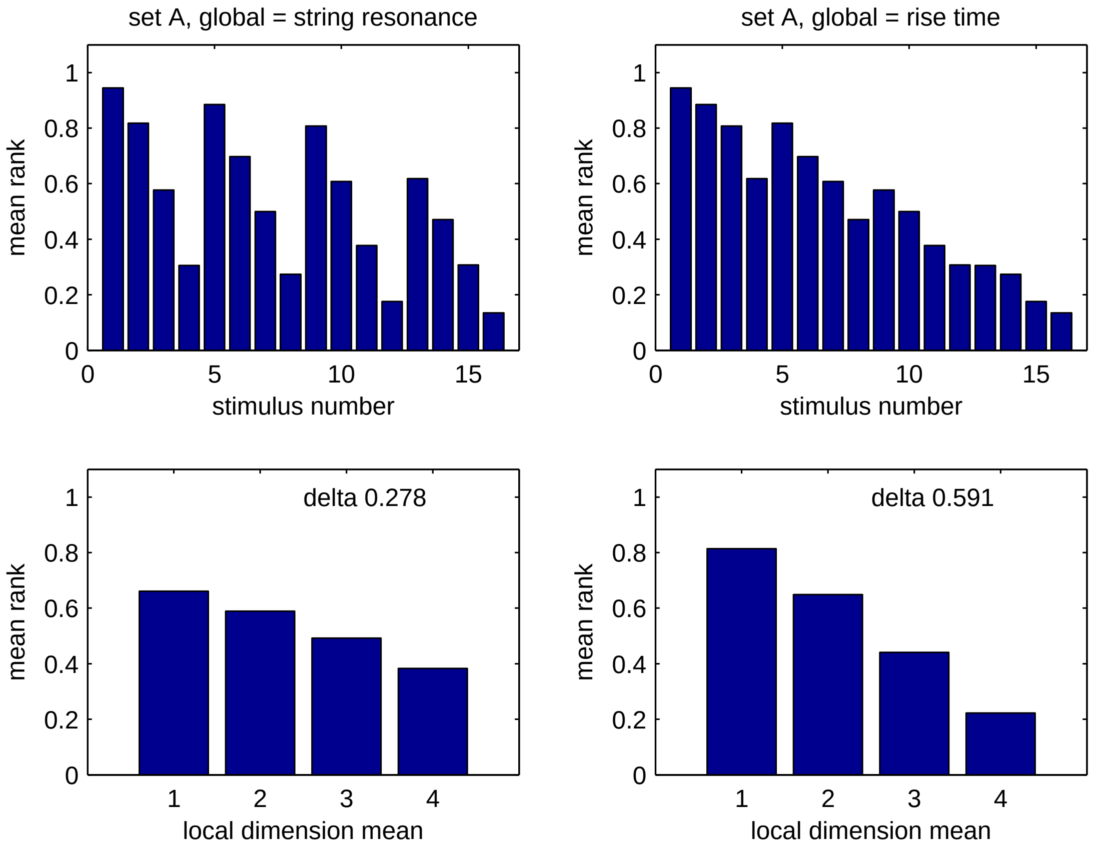


Figure 3.6: Normalized mean ranks for stimulus set A. Higher values represent "more like an SDPE." The top left graph shows the mean values for each of the stimulus sounds using number system A. The top right graph shows the same mean values using number system B. The bottom left graph shows the mean ranks averaged in groups of four using number system A. This graph shows the effect of only the global dimension from the graph above it. The bottom right graph shows the mean ranks averaged in groups of four using number system B from the graph above it. The delta values represent the difference between the maximum and minimum local-dimension means.

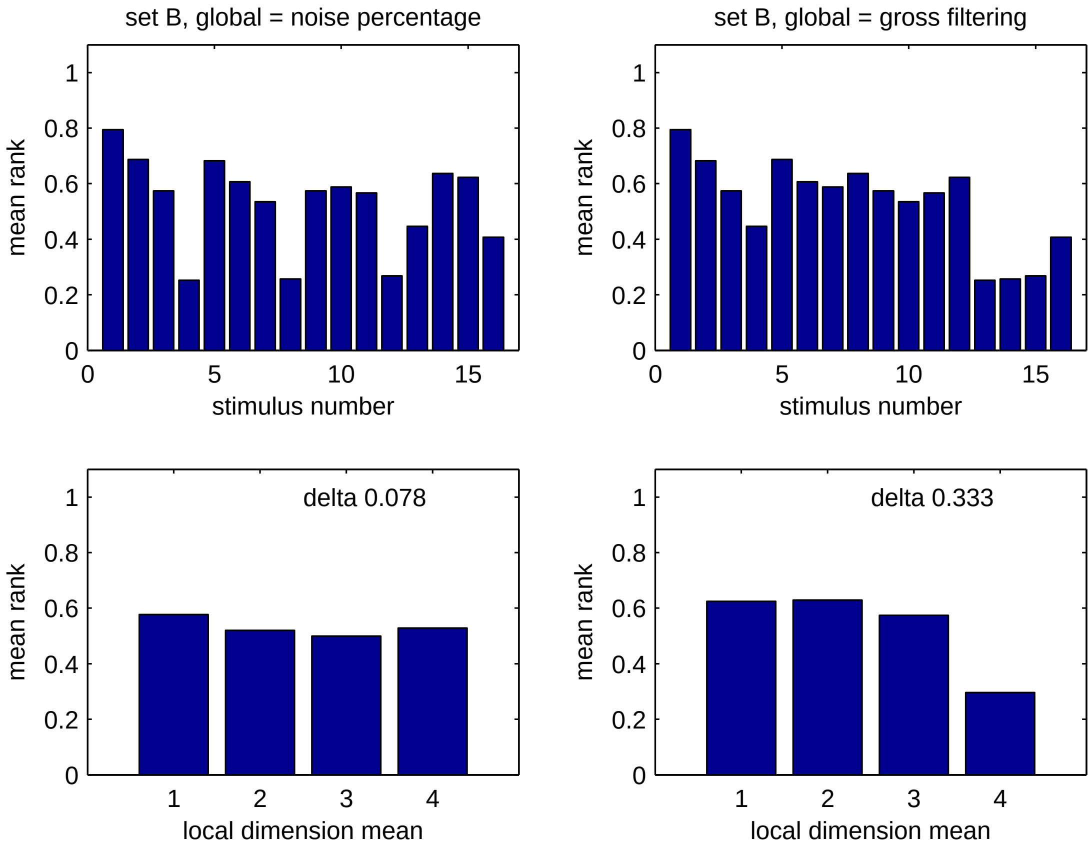


Figure 3.7: Normalized mean ranks for stimulus set B. Higher values represent "more like an SDPE." The top left graph shows the mean values for each of the stimulus sounds using number system A. The top right graph shows the same mean values using number system B. The bottom left graph shows the mean ranks averaged in groups of four using number system A. This graph shows the effect of only the global dimension from the graph above it. The bottom right graph shows the mean ranks averaged in groups of four using number system B from the graph above it. The delta values represent the difference between the maximum and minimum local-dimension means.

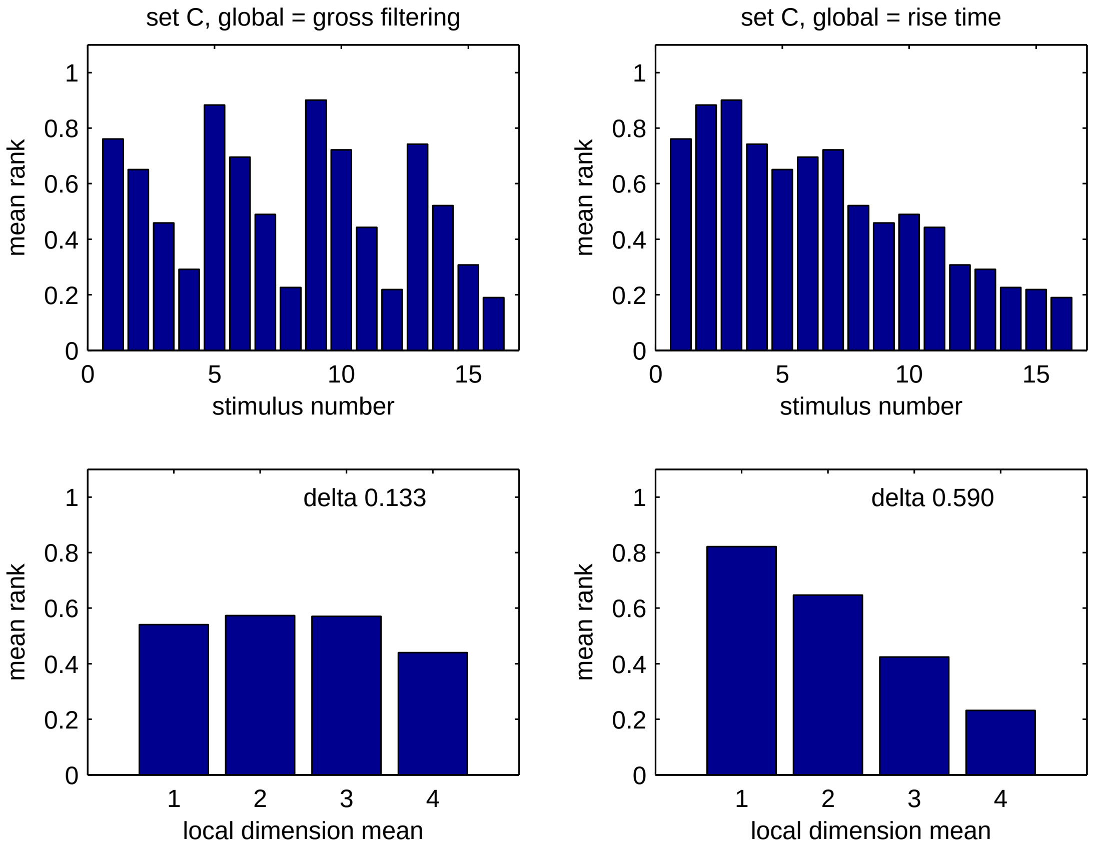


Figure 3.8: Normalized mean ranks for stimulus set C. Higher values represent “more like an SDPE.” The top left graph shows the mean values for each of the stimulus sounds using number system A. The top right graph shows the same mean values using number system B. The bottom left graph shows the mean ranks averaged in groups of four using number system A. This graph shows the effect of only the global dimension from the graph above it. The bottom right graph shows the mean ranks averaged in groups of four using number system B from the graph above it. The delta values represent the difference between the maximum and minimum local-dimension means.

The normalized mode ranks were similar to the normalized mean ranks with one notable difference. The normalized mode ranks are forced to rest at 16 discrete equal steps between 0 and 1. This was how a single participant's ranks would appear if the participant had made all of the choices most common for each stimulus sound pair.

The local-dimension means for the normalized mode ranks also were similar to the local-dimension means for the normalized mean ranks. The local-dimension means for the normalized mode ranks did show slightly higher delta values than the local-dimension means for the normalized mean ranks. The higher delta values may be due to the forcing of the normalized ranks into specific values,

3.2.3 Outliers Removed

The responses of a few of the participants seemed to be significantly different from the majority of participants. In order to find these outlier participants, each participant's choices were compared against the modal choices. Participants then were sorted according to the percentage of their choices that were the same as the modal choice. The sorted list of participants for all stimulus sets is shown in Figure 3.12.

In each of the stimulus sets, the one or two participants with the lowest percentage of choices like the modal choices represented a sudden unusual drop in this percentage. For each stimulus set, these last one or two participants were declared outliers and their data were removed from the total data set.

The mean values of participants' choices which were the same as the modal choices for stimulus sets A, B, and C were 82.60, 69.04, and 81.35%, respectively. The lower mean for stimulus set B indicates that for whatever reason, less consistency existed between participants for their stimulus pair choices for this stimulus set. The highest percentages of any single participant's choices which were the same as the modal choices for stimulus sets A, B, and C were 95.00, 85.00, and 92.50%, respectively.

The mean ranks with the outliers removed for all data sets are shown in Figures 3.13–3.15 for stimulus sets A, B, and C, respectively. The effect on the mean ranks of removing the outliers in all of the stimulus sets was minimal. The changes seen in the delta of the local-dimension means for the mean rank values were less than 6% of the maximum possible delta. For stimulus set A, the effect of string resonance decreased the delta by 1.1% while the effect of rise time increased the delta by 3.2%. For stimulus set B, the effect of noise percentage increased the delta by 1.4% and the effect of gross spectral filtering increased the delta by 4.2%. For stimulus set C, the effect of gross spectral filtering decreased the delta by 0.9% while the effect of rise time increased the delta by

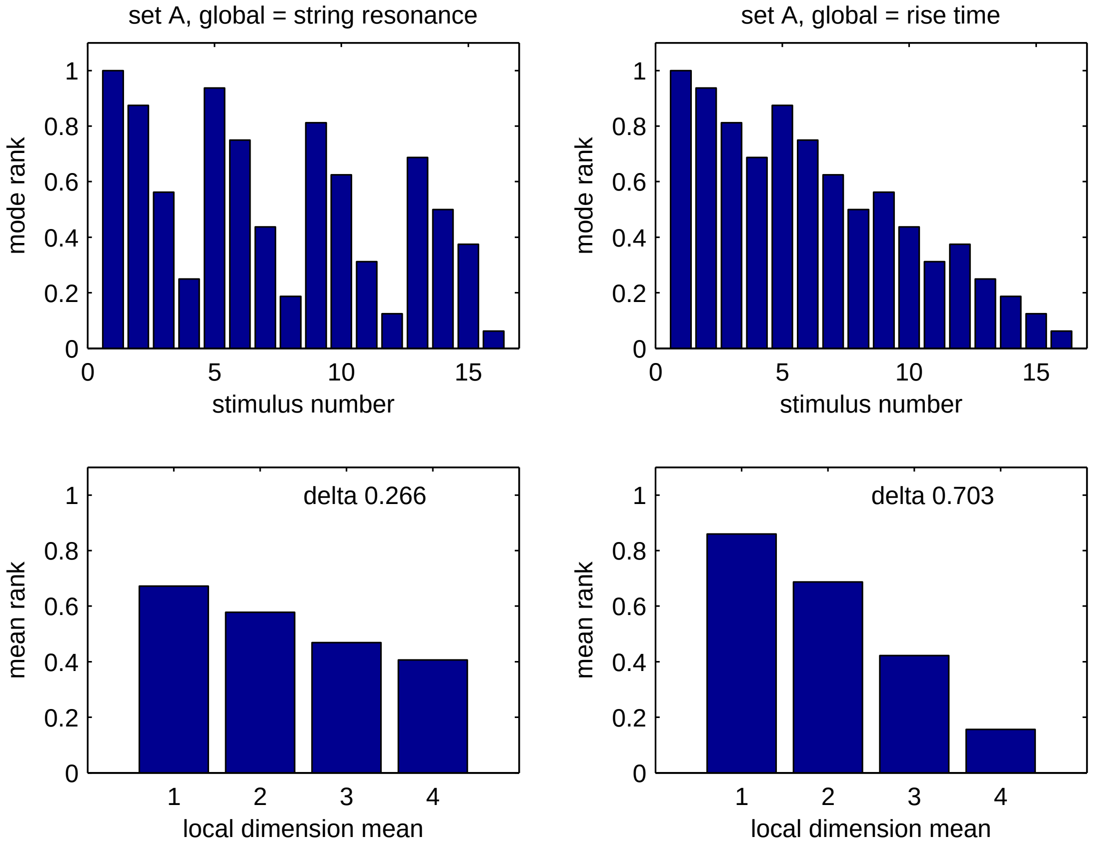


Figure 3.9: Normalized mode ranks for stimulus set A. Higher values represent "more like an SDPE." The top left graph shows the mean values for each of the stimulus sounds using number system A. The top right graph shows the same mean values using number system B. The bottom left graph shows the mean ranks averaged in groups of four using number system A. This graph shows the effect of only the global dimension from the graph above it. The bottom right graph shows the mean ranks averaged in groups of four using number system B from the graph above it. The delta values represent the difference between the maximum and minimum local-dimension means.

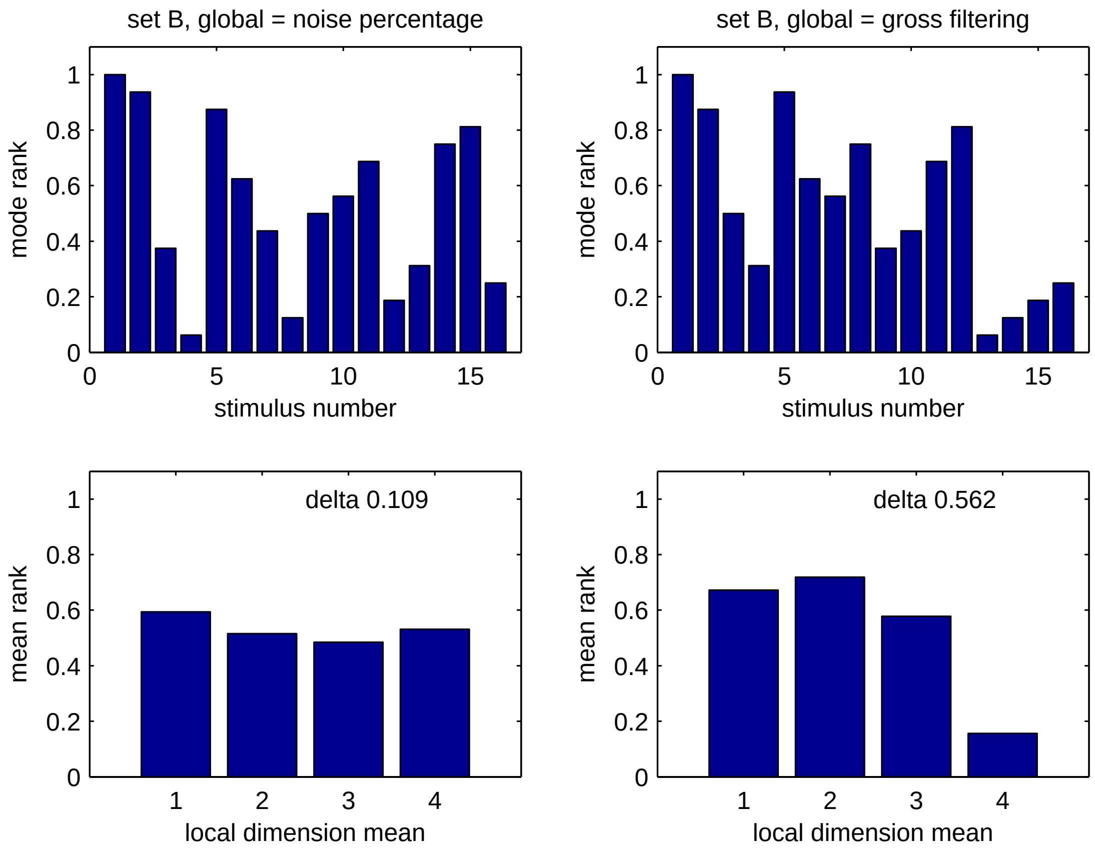


Figure 3.10: Normalized mode ranks for stimulus set B. Higher values represent "more like an SDPE." The top left graph shows the mean values for each of the stimulus sounds using number system A. The top right graph shows the same mean values using number system B. The bottom left graph shows the mean ranks averaged in groups of four using number system A. This graph shows the effect of only the global dimension from the graph above it. The bottom right graph shows the mean ranks averaged in groups of four using number system B from the graph above it. The delta values represent the difference between the maximum and minimum local-dimension means.

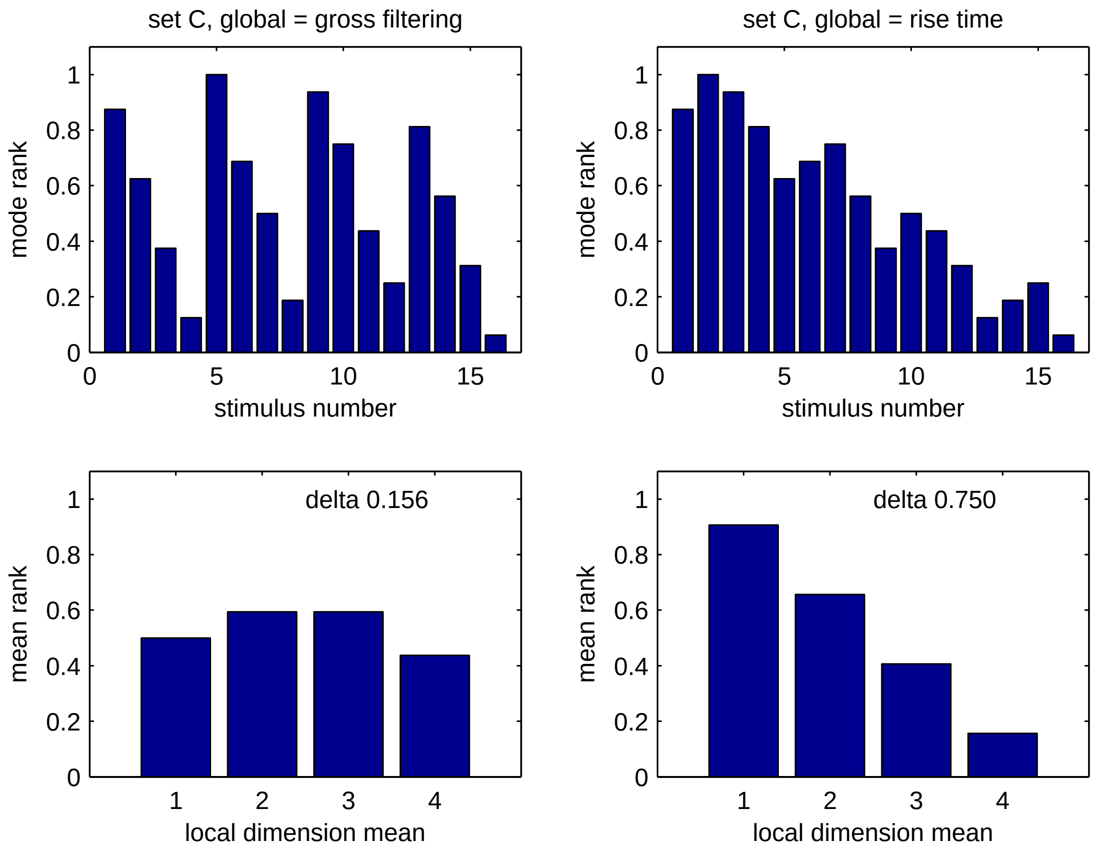


Figure 3.11: Normalized mode ranks for stimulus set C. Higher values represent "more like an SDPE." The top left graph shows the mean values for each of the stimulus sounds using number system A. The top right graph shows the same mean values using number system B. The bottom left graph shows the mean ranks averaged in groups of four using number system A. This graph shows the effect of only the global dimension from the graph above it. The bottom right graph shows the mean ranks averaged in groups of four using number system B from the graph above it. The delta values represent the difference between the maximum and minimum local-dimension means.

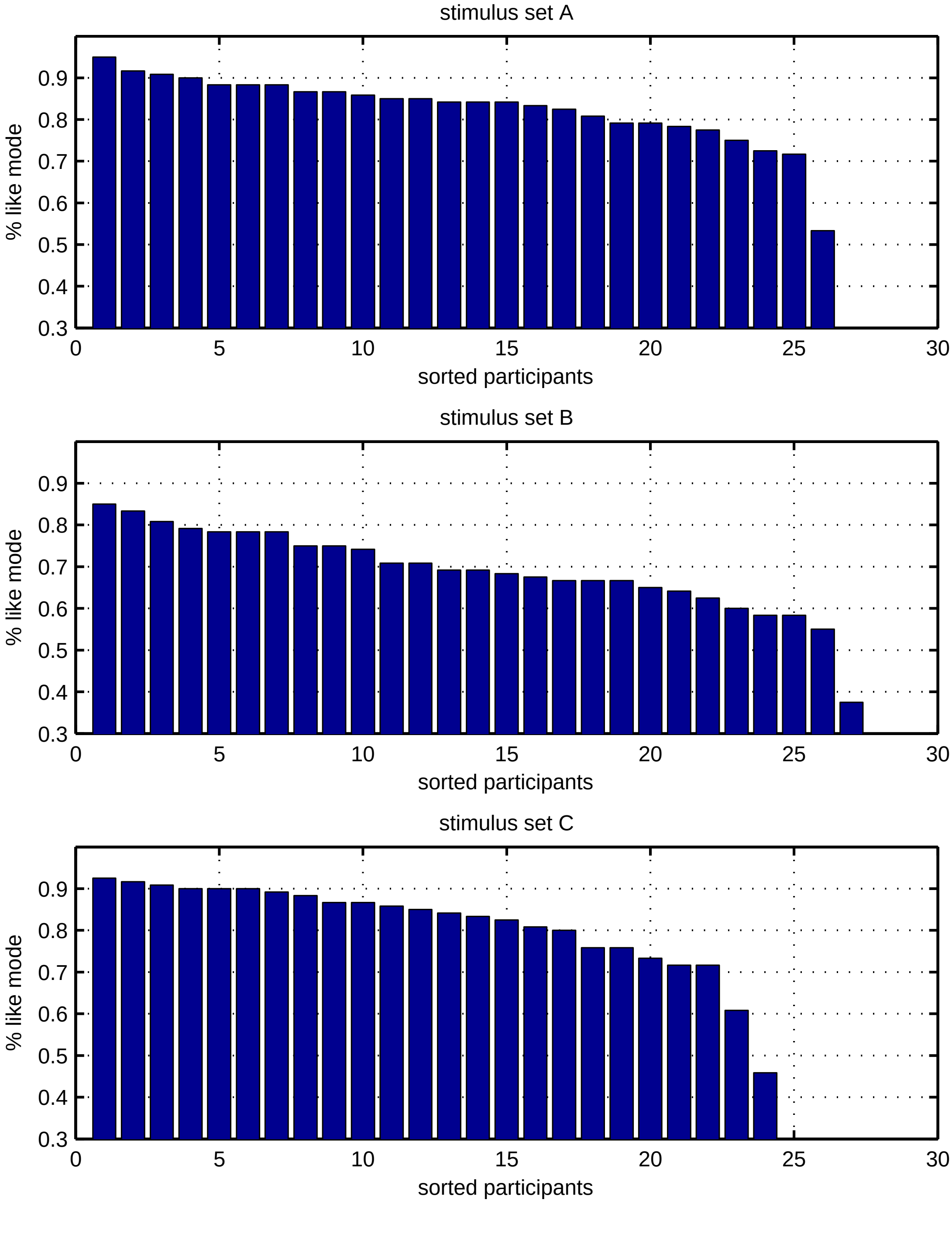


Figure 3.12: Participants sorted by the percentage of their choices which matched the modal choices for stimulus sets A, B, and C. Outliers were determined by a sudden unusual drop in this percentage between sorted participants.

the largest amount of 5.8%.

The modal ranks with outliers removed are shown in Figures 3.16–3.18 for stimulus sets A, B, and C, respectively. For stimulus sets A and C, removing the outliers had no effect on the mode ranks at all. For stimulus set B, removing the outlier changed the mode ranks more significantly. The effect on the local-dimension means for the global dimension of noise percentage was a change of 14.1% of the maximum possible delta. This large change indicates that, as stated in Section 3.2.1, the balance of participants using noise percentage as a cue for percussivity and participants using it as a negative cue was nearly equal. Removing even one participant's choices from the calculation of modal ranks had a significant effect. For gross spectral filtering, the change in the delta of the local-dimension means was only 1.5%.

3.2.4 Statistical Verification

Because the Jarque-Bera test [70] shows that 3 of the 24 sets of local-dimension means for the mean rank values are not from normally distributed data, the Friedman nonparametric test [71] was used to test the following null hypothesis: the variation of the dimensions of the stimulus sounds caused no effect on the local-dimension means. According to the Friedman test, the null hypothesis was rejected with at least a 95% confidence level for all of the local-dimension means.

3.2.5 Percussion Threshold

Table 3.3 shows the mean and mode percussion threshold points for each stimulus set across participants both with and without outliers. These threshold values represent the cross-participant means and modes of the number of stimulus sounds that participants judged to be SDPEs from their own ranked list. The mode value for percussion threshold remained the same with and without outliers.

3.3 Concluding Remarks

The goal of this collection of human percussion judgments was to find one or two fundamental dimensions of sound events that could be used as a cue to anticipate the human judgment of percussivity. Perhaps unsurprisingly, the results indicate that rise time was the strongest cue for percussivity of the dimensions tested.

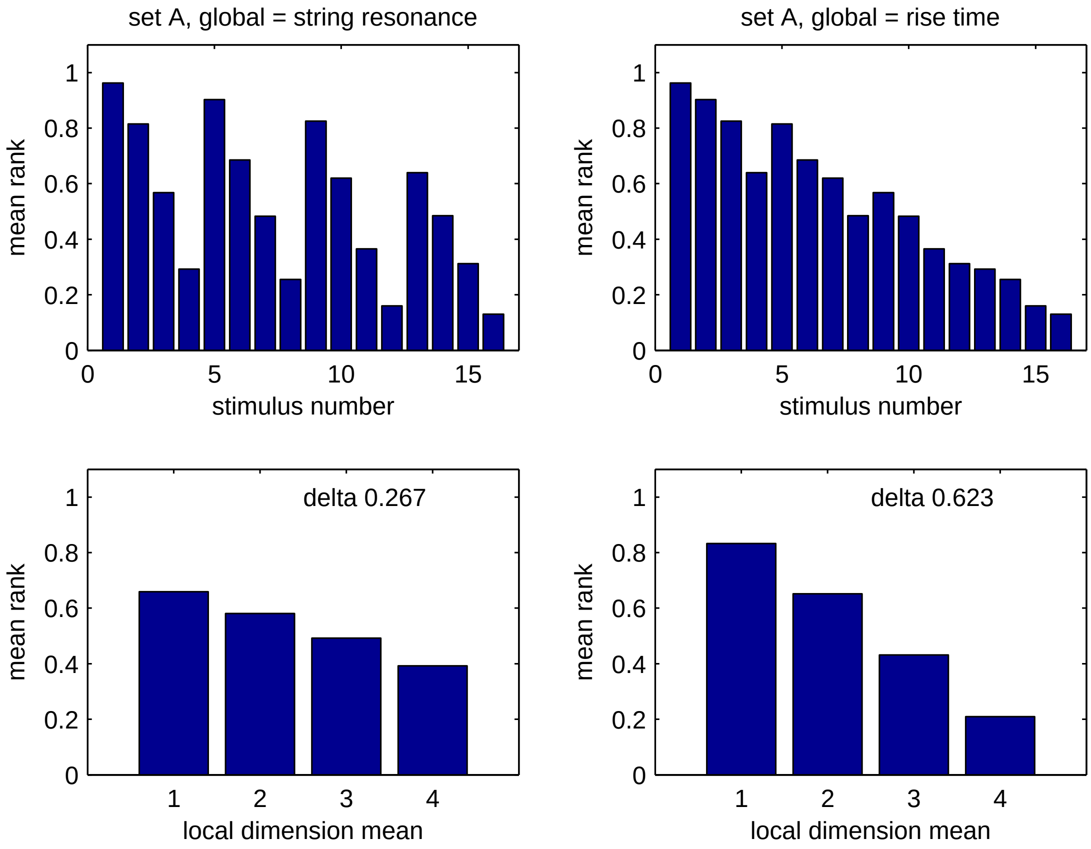


Figure 3.13: Normalized mean ranks for stimulus set A without outliers. Higher values represent "more like an SDPE." The top left graph shows the mean values for each of the stimulus sounds using number system A. The top right graph shows the same mean values using number system B. The bottom left graph shows the mean ranks averaged in groups of four using number system A. This graph shows the effect of only the global dimension from the graph above it. The bottom right graph shows the mean ranks averaged in groups of four using number system B from the graph above it. The delta values represent the difference between the maximum and minimum local-dimension means.

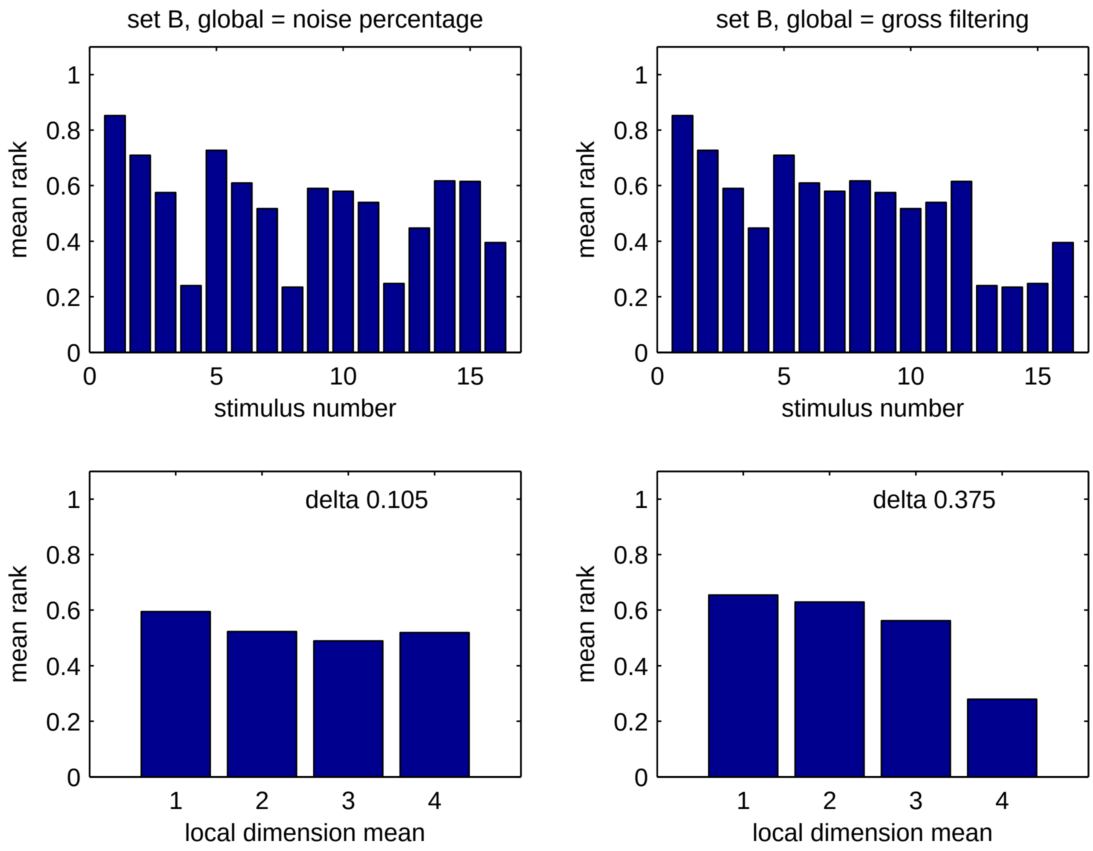


Figure 3.14: Normalized mean ranks for stimulus set B without outliers. Higher values represent “more like an SDPE.” The top left graph shows the mean values for each of the stimulus sounds using number system A. The top right graph shows the same mean values using number system B. The bottom left graph shows the mean ranks averaged in groups of four using number system A. This graph shows the effect of only the global dimension from the graph above it. The bottom right graph shows the mean ranks averaged in groups of four using number system B from the graph above it. The delta values represent the difference between the maximum and minimum local-dimension means.

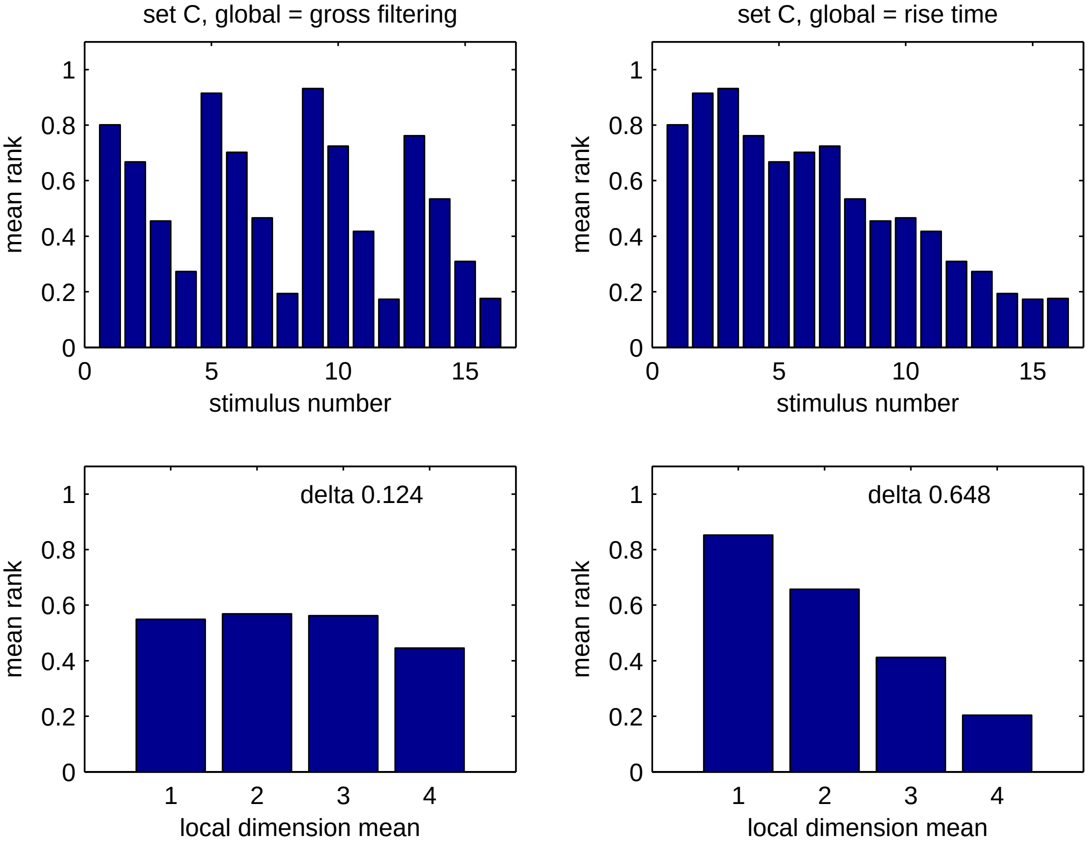


Figure 3.15: Normalized mean ranks for stimulus set C without outliers. Higher values represent "more like an SDPE." The top left graph shows the mean values for each of the stimulus sounds using number system A. The top right graph shows the same mean values using number system B. The bottom left graph shows the mean ranks averaged in groups of four using number system A. This graph shows the effect of only the global dimension from the graph above it. The bottom right graph shows the mean ranks averaged in groups of four using number system B from the graph above it. The delta values represent the difference between the maximum and minimum local-dimension means.

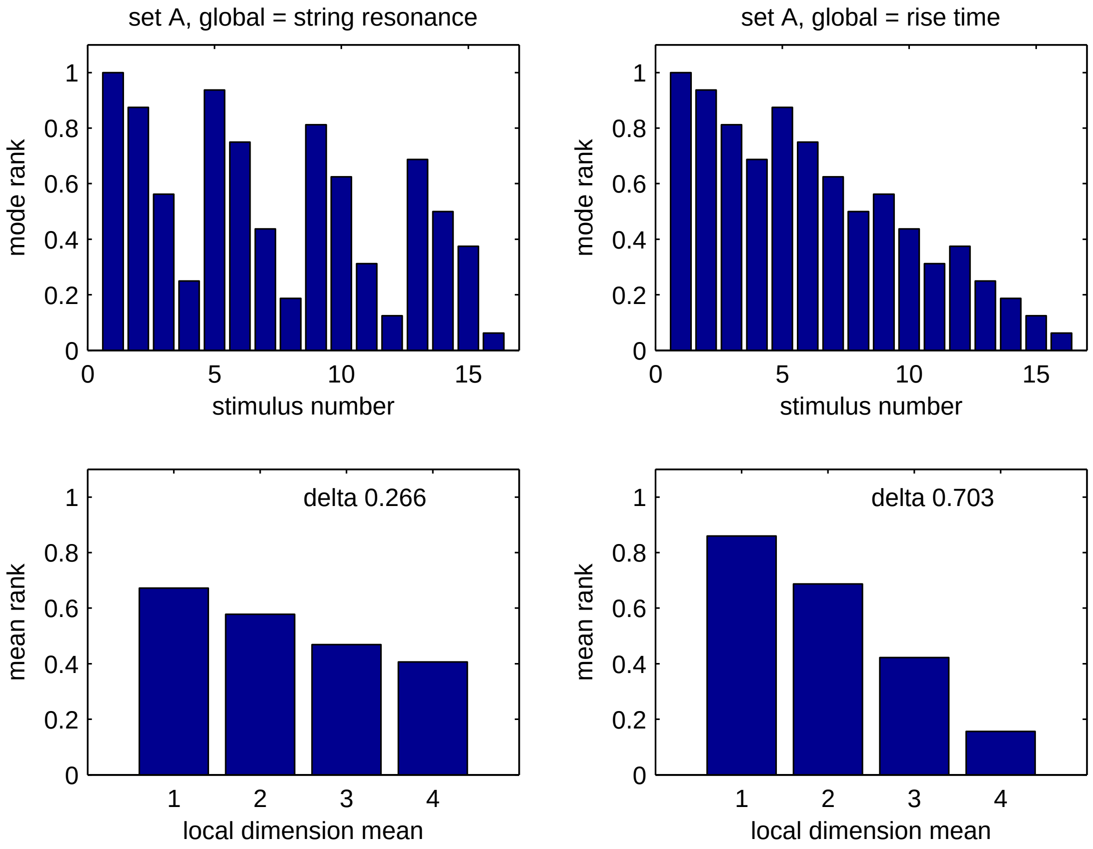


Figure 3.16: Normalized mode ranks for stimulus set A without outliers. Higher values represent “more like an SDPE.” The top left graph shows the mean values for each of the stimulus sounds using number system A. The top right graph shows the same mean values using number system B. The bottom left graph shows the mean ranks averaged in groups of four using number system A. This graph shows the effect of only the global dimension from the graph above it. The bottom right graph shows the mean ranks averaged in groups of four using number system B from the graph above it. The delta values represent the difference between the maximum and minimum local-dimension means.

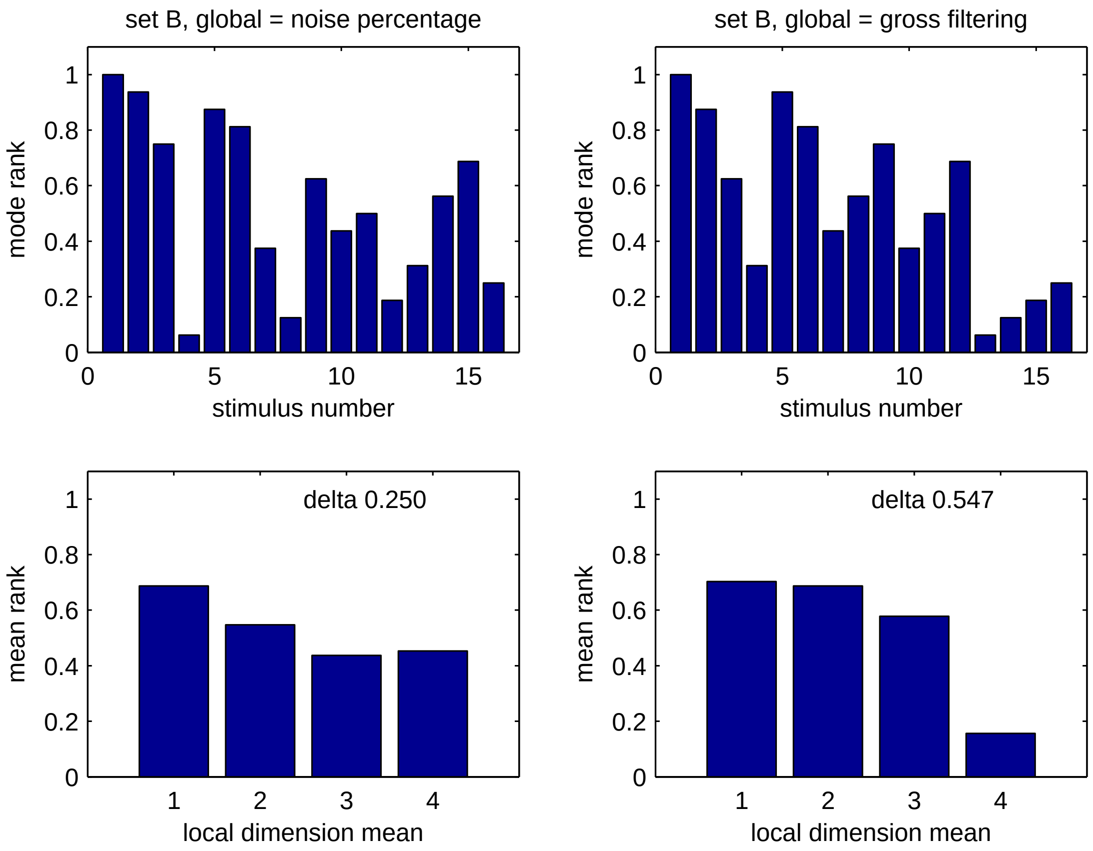


Figure 3.17: Normalized mode ranks for stimulus set B without outliers. Higher values represent "more like an SDPE." The top left graph shows the mean values for each of the stimulus sounds using number system A. The top right graph shows the same mean values using number system B. The bottom left graph shows the mean ranks averaged in groups of four using number system A. This graph shows the effect of only the global dimension from the graph above it. The bottom right graph shows the mean ranks averaged in groups of four using number system B from the graph above it. The delta values represent the difference between the maximum and minimum local-dimension means.

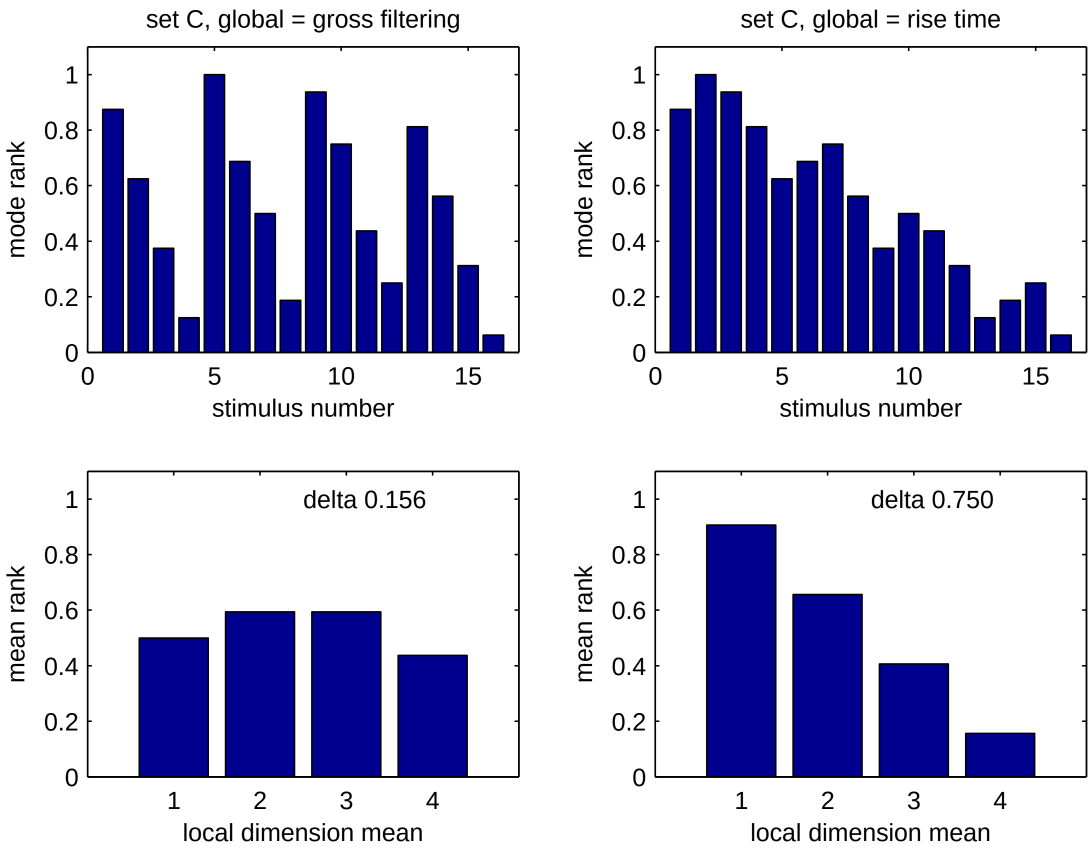


Figure 3.18: Normalized mode ranks for stimulus set C without outliers. Higher values represent “more like an SDPE.” The top left graph shows the mean values for each of the stimulus sounds using number system A. The top right graph shows the same mean values using number system B. The bottom left graph shows the mean ranks averaged in groups of four using number system A. This graph shows the effect of only the global dimension from the graph above it. The bottom right graph shows the mean ranks averaged in groups of four using number system B from the graph above it. The delta values represent the difference between the maximum and minimum local-dimension means.

Table 3.3: Mean and mode SDPE threshold values. These threshold values represent the cross-participant means and modes of the number of stimulus sounds that participants judged to be SDPEs from their own ranked list for the three stimulus sets.

stimulus set	**mean threshold**	**mean threshold without outliers**	**mode threshold**
A	10.88	10.96	10
B	10.38	10.44	11
C	10.27	10.67	8

String resonance also had a strong effect on whether humans judged a sound event to be percussive, but because of the complicated spectral and temporal effects of string resonance, it is neither a fundamental nor feasibly measured dimension of a sound event. Originally, the effect of string resonance had been intended to create a tone-like sound from pure noise, but according to the comments of the participants after the judgment collection, the addition of string resonance created complicated temporal and spectral changes to the stimuli. It may have been these other changes that were affecting the percussion judgments instead of the tonal content of the stimulus. It is impossible to determine the specific cause of the effect of string-resonance on percussion judgment from the current judgment collection.

Gross spectral filtering also had an effect on the human judgment of percussivity, although it was smaller than the effects of either rise time or string resonance. The effect of gross spectral filtering also was weaker when the much stronger cue of rise time was present. Perhaps it would be only in the absence of a stronger cue that gross spectral filtering would become an effective cue of percussivity.

These results were used to motivate the design of the percussion detection algorithm discussed in Chapter 4. How the rise time, the modal choices, and threshold values were used will be discussed there. Chapter 6 discusses some possible changes to the methodology of this collection.

Chapter 4

Percussivity-Profile Algorithm

This chapter describes the motivation, implementation, realization, and results of the percussivity-profile algorithm (PPA). *Percussivity* refers to how percussive an instant of a piece of music is. A *percussivity profile* is a percussivity rating for all instants of the entire length of a segment of sound. Because the PPA operates on a discrete-time input sound, "instants" actually refer to short time windows when discussing the PPA. The exact length of the windows will be examined as a design parameter.

The reason the PPA was created was to provide an algorithm which detects instants that humans would label as percussive in recordings of electroacoustic music. The approach used for the PPA a time-domain, channel-based algorithm using psychoacoustic models. Once the algorithm was implemented, the results of the percussion-judgment collection from Chapter 3 were used to tune the PPA. After the tuning took place, the PPA then was used to analyze a constructed example and two pieces of electroacoustic music. The pieces of electroacoustic music introduced new challenges, and the results of the analysis are described in the final section of this chapter.

4.1 Algorithm Models

In order to achieve the goal set forth for the PPA, the algorithm was designed based on time-domain onset detection, but also implemented models of the human auditory periphery. This section describes the general time-domain, onset-detection algorithm and the models of the human auditory periphery used.

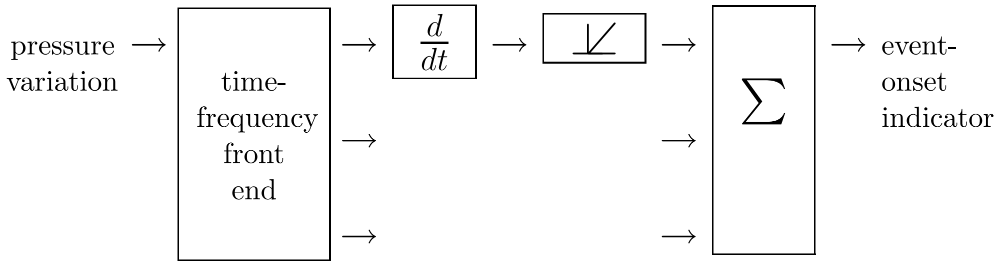


Figure 4.1: A general algorithm for time-domain onset detection. Some form of pressure variation enters a time-frequency front end where the input time waveform is split into multiple channels. The derivative of each channel is taken and then half-wave rectified. Some form of cross-channel summation is performed to create an event-onset indicator.

4.1.1 General Algorithm for Time-Domain Onset Detection

From the percussion-judgment collection, rise time was found to be the most important fundamental audio property that would indicate the percussivity of a sound event. One approach to a rise-time algorithm is to use the basic processing of a time-domain, onset-detection algorithm based on amplitude over time. The general design for such an algorithm is shown in Figure 4.1 and is seen in many of the algorithms in the reviews listed in Section 2.5.2. This approach is not inherently psychoacoustically motivated, but can be adapted to include psychoacoustic models. This general model has been previously presented by the author [72].

In the general form of the algorithm, a time-frequency front end is used to perform some initial time and frequency analysis of an incoming pressure variation represented by a sound file and then to generate many channels of output where each channel corresponds to some region in the frequency domain of the incoming pressure variation. The derivative followed by half-wave rectification of the information in each channel is then taken in order to find short time windows of increasing amplitude of the output of the time-frequency front end. The corresponding information is then collected in various ways across all of the incoming channels in order to generate an event-onset indicator.

There are some significant problems with this algorithm when using it for percussion detection. The most serious is the influence of both the analysis window length and the analysis start time. The numerical implementation of $\frac{d}{dt}$ (explained further in Section 4.2.7) requires a value $\triangle t$, which is the analysis window length. If the analysis window length is too long, then percussive events might be missed because an entire percussive attack and at least some of the sustain or release might rest within the analysis window length. If the analysis window length is too short, then too many false positives will occur due to inaccurate representation of the amplitude envelope. Even if the analysis window is somehow exactly the right length for every percussive event within the piece of music, the analysis start time will change the resulting percussivity values. The analysis start time also might cause the beginning of an analysis window to fall midway through the rise of a percussive event, effectively cutting the percussivity rating of that instant in the music by half. Techniques for avoiding these problems are presented in Section 4.2.6.

4.1.2 Human Hearing Model

The PPA makes use of the Auditory Toolbox [73] by Slaney. The Auditory Toolbox provides algorithms for examining different types of auditory time-frequency representations of sound. One of these representations uses a model proposed by Patterson et al. [74] of psychoacoustic filtering based on a critical band function implemented with a gammatone filter bank. A critical band refers to a small frequency band which describes the frequency selectivity of humans determined by the ability of wideband noise to inhibit the perception of a pure tone [75]. A gammatone filter bank is a set of psychoacoustically-designed filters which mimic human frequency selectivity and time sensitivity. An example of the spectrum of a gammatone filter bank is shown in Figure 4.2. The model proposed by Patterson et al. was extended with the Auditory Image Model software [76] and more details can be found in the documentation of that software. The number of filters in the filter bank and the frequency range that they cover are two of the PPA design parameters.

Within the time-frequency representation, a model of the neurotransmitter release at the base of inner hair cell proposed by Meddis [77] follows the gammatone filter bank. Meddis's model is used to predict the neural spike patterns, and is described in the following paragraph. Inner hair cells are the primary sensory receptor cells of the auditory system and are located in the cochlea. A neural spike (a neural firing, neurotransmission, or synaptic transmission) is the electrochemical mechanism by which neurons (nerve cells) transmit information.

The Meddis model is concerned with the initiation of a neural spike in an auditory nerve fiber by an inner hair cell. It uses a probabilistic model for neurotransmitter release from inner hair cells, auditory neural firings, and discharge patterns. The model assumes that the amount of neurotransmitter released from the hair cell is a function of the stimulus intensity, that some of the neurotransmitter is taken back into the hair cell, and that some amount of neurotransmitter is lost from the synaptic cleft. A schematic of the mechanisms can be seen in Figure 4.3. For a complete description of the model see Meddis [77].

Using several functions in the Auditory Toolbox, it is possible to implement the

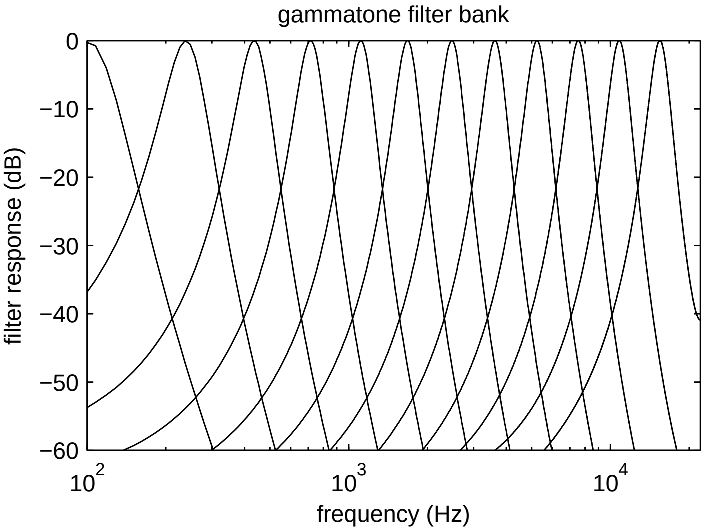


Figure 4.2: A spectral plot showing the frequency selectivity of an example gammatone filter bank.

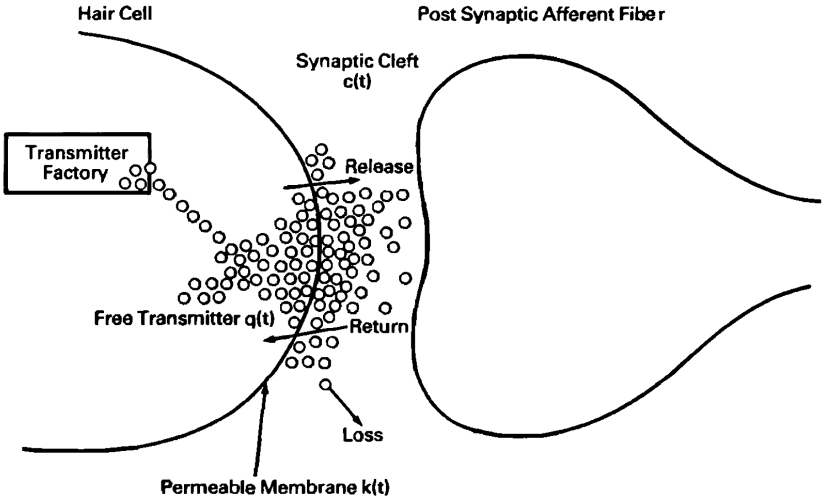


Figure 4.3: A synapse according to the Meddis model [77]. This shows neurotransmitter substance being generated and released by the hair cell into the synaptic cleft according to the stimulus intensity. Some neurotransmitter substance is assumed to be taken back into the hair cell; some is assumed to be lost.

described time-frequency representation. The chain of these functions converts a WAV sound file into a series of neural spike potentials hypothetically traveling along auditory nerve fibers of the cochlear nerve.

4.2 Description of the Percussivity-Profile Algorithm

The PPA is an analysis-only algorithm that takes as input a monophonic, WAV sound file, and returns a percussivity profile for that sound file. The parameters of the auditory processing that allow control of the PPA performance are `nChannels` (the number of divisions of the basilar-membrane model), `lowFreq` (the lowest frequency analyzed by the basilar-membrane model), `windowTimeLen` (the time division for decimation and differentiation), `hairCellScaling` (the apparent amplitude of the sound file), and `nChannelGroups` (the number of groups into which the basilar-membrane channels are combined).

The results of the PPA are two indicators of percussivity. One is a single value of percussivity over time and is called the single-value percussivity profile (SPP). The SPP is shown as a graph; the axes of the graph are time (abscissa) and percussivity (ordinate). This value is post-processed to improve visibility of percussive instants, and this post-processing is described in Section 4.4.1, and is called SPP_p.

The other indicator, called the group percussivity profile (GPP), is a matrix. The rows of the GPP indicate maximum percussivity within frequency bands and the columns indicate maximum percussivity within time windows. The GPP is shown as a two-dimensional image with larger values in the elements of the matrix displayed more darkly; the axes of the image are time (abscissa) and channel group (ordinate). The channel groups are the frequency bands mentioned, and low number channel groups contain high-frequency percussivity while high number channel groups contain low-frequency percussivity. The channel groups of the GPP are dependent on `nChannels`, `lowFreq`, and `nChannelGroups` as detailed in the following algorithm description.

The amplitude waveform, post-processed percussivity profile (SPP_p), and GPP for an electroacoustic music example are shown in Figure 4.4. This example is described and analyzed further in Section 4.5.3.

The basic processing of the PPA involves first converting an input sound file into neural spike probabilities along hypothetical neural channels, and then grouping and manipulating those neural spike probabilities into useful indicators of percussivity over time. The details of the PPA can be broken down into the following discrete steps:

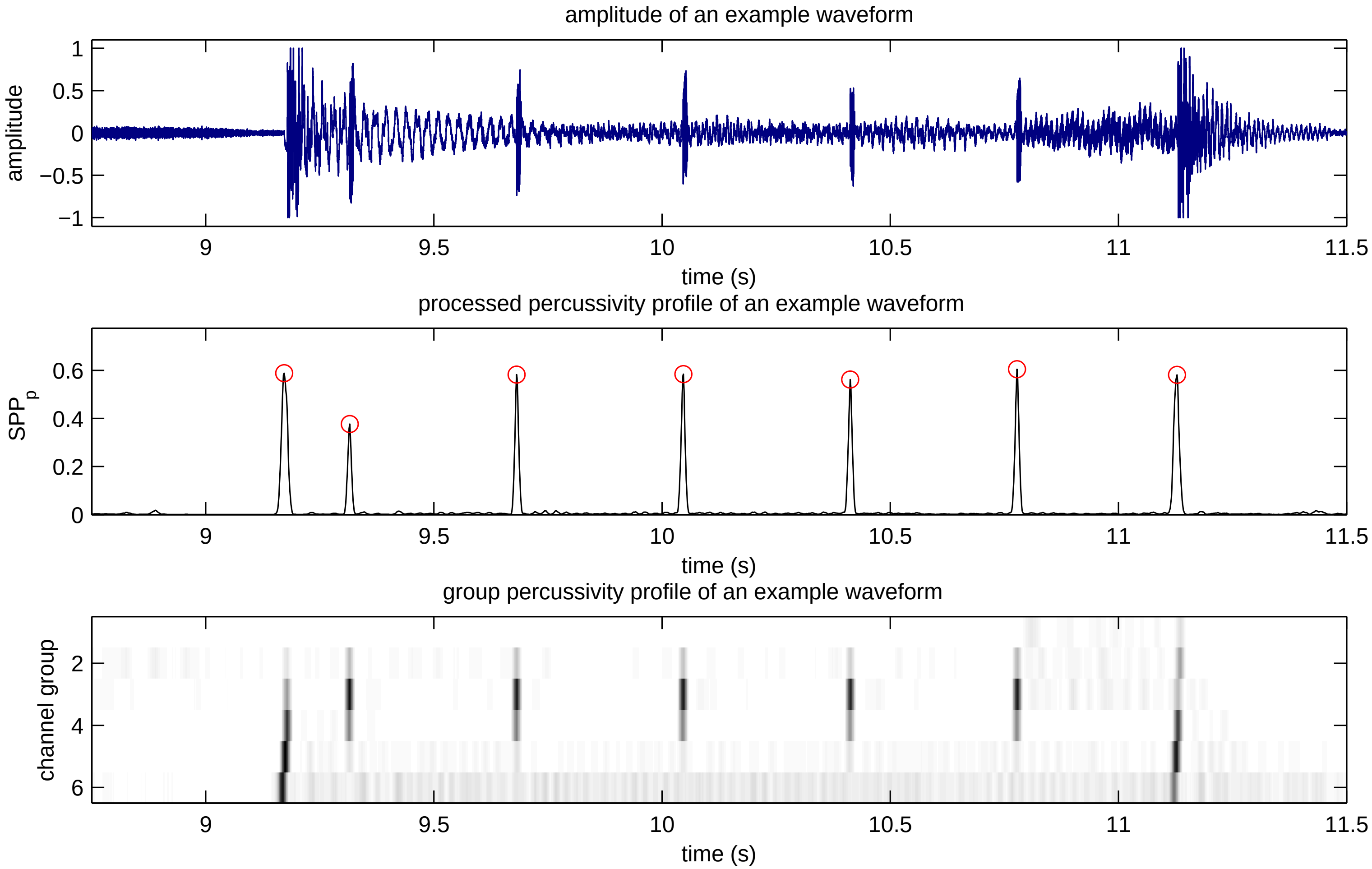


Figure 4.4: Amplitude (top plot), SPP_p (middle plot), and GPP (lower plot) of a short segment of electroacoustic music.

1. cochlear response
2. zero padding
3. hair-cell response
4. time realignment
5. low-pass filter
6. copy with time-shift and decimation
7. differentiation
8. half-wave rectification
9. channel-group means
10. upsample and unshift
11. maximums of channel-group means
12. means of copies
13. removing zero padding

Figure 4.5 shows how the steps connect with one other, and the details of the algorithm steps follow.

4.2.1 Cochlear Response

The *cochlear response* step consists of using the audio-rate (44.1 kHz for this research) sound file to predict the motion reaction of the different areas of the basilar membrane to the pressure variation represented by the sound file. The *ERBFilterBank* function of the Auditory Toolbox is used for this prediction. This step does not account for the issues of transduction to the basilar membrane, and also does not account for the limited range over which the basilar membrane can move; it is assumed that the motion of the basilar membrane is a linear scaling of the input. This step does account for the delay in maximum motion of the basilar membrane from the initial motion arriving at the oval window, and also accounts for the psychoacoustic frequency distribution along the basilar membrane.

The output of this step is a set of audio-rate, scaled representations of the basilar-membrane motion in the region of each channel. The basic mechanism of this step is a gammatone filter bank described in Section 4.1.2. The gain of each filter of the gammatone filter bank is 0 dB at the center frequency of the filter. The parameters of `lowFreq` and `nChannels` affect, respectively, the lowest frequency and number of channels of the gammatone filter bank used in this step. The highest frequency of the gammatone filter bank is the Nyquist frequency, $\frac{f_s}{2}$, derived from the sampling frequency, f_s, of the sound

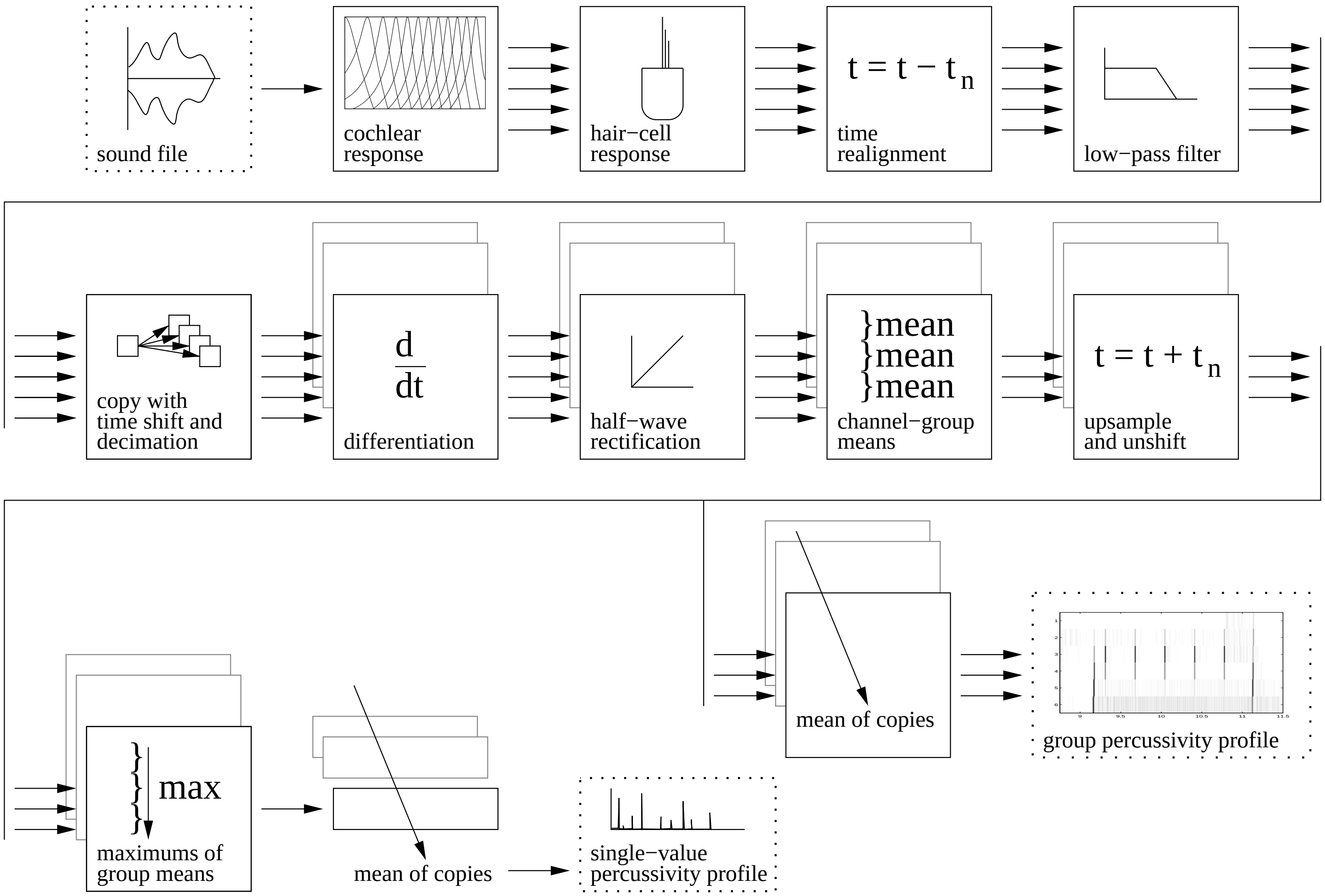


Figure 4.5: Detailed model of the percussivity-profile algorithm. The input to this model is a sound file and the output is a single-value percussivity profile and a group percussivity profile (GPP).

file being processed. Once the basilar-membrane motion for each channel is generated in this step, each channel is subsequently processed separately with no interaction between channels until the *channel-group means* step near the end of the algorithm.

4.2.2 Zero Padding

The *zero padding* step consists of adding zero values to the beginning and end of the output of the *cochlear response* step. This padding process prepares the cochlear response for subsequent processing and permits later in-channel time-shifting of the channels. Zero padding adds high frequency content to the analysis at the transition points if the input waveform is not preprocessed with a fade-in and fade-out amplitude envelope.

4.2.3 Hair-Cell Response

The *hair-cell response* step consists first of scaling the cochlear response channels according to a predetermined listening level of the sound file, and then the *MeddisHairCell* function of the Auditory Toolbox is used to predict the neural-spike probability along the channel-appropriate portion of the cochlear nerve. The scaling of the cochlear response is controlled by the `hairCellScaling` parameter. The *MeddisHairCell* function does take into account the effect of amplitude levels on the firing rates of the hair cells and the possible deficit of neurotransmitter substance in the synaptic cleft at the base of the hair cell due to previous excitation of the basilar membrane. The results of this step are non-dimensionalized by `windowTimeLen`, the window time length chosen, divided by a maximum possible firing rate for a neuron of 1000 $\frac{spikes}{s}$. This high normalization value for firing rate is chosen to extend beyond the range of any possible firing rate values from the *MeddisHairCell* function and is based on a refractory period of 1.0 millisecond [78].

The output of this step is a set of audio-rate, non-dimensionalized firing rates that indicate the rate at which neural spikes would be firing from the base of the hair cells. One firing-rate stream exists for each channel. Unfortunately, with regard to the output of the *MeddisHairCell* function of the Auditory Toolbox, “the amplitude level is arbitrary [73]” so only relative percussivity values are given. The output level is controlled both by the input level (determined by `hairCellScaling` and the amplitude level of the incoming sound file) and constants within the function (which have remained constant throughout the entirety of this research), so percussivity values can be compared between pieces of music analyzed only if the input parameter values remain constant. Differences in musical style and recording technique still may affect the percussivity values enough that direct comparison is not possible. Because percussivity values are relative, if there is no percussive event that the algorithm would label as percussive in the piece of music being analyzed, then the results of the PPA on that piece of music will not necessarily be meaningful.

4.2.4 Time Realignment

The *time realignment* step consists of shifting the channels earlier in time according to the time delay imposed by the gammatone filter bank. This time realignment is done according to both the gross features of the envelope imposed by the gammatone filter bank and by the fine structure of the phase of the resulting basilar membrane motion at the center frequency of the region of interest. Low-frequency channels are shifted further (earlier) in time than high-frequency channels. There is evidence to both validate [79] and contradict [80], [81] that this time realignment is done in the processing of the auditory system, but the model of the human auditory periphery used up to this point does not include any time realignment. Based on testing and tuning described later in Section 4.3.3, the decision was made to artificially impose this time realignment. The output of this step is a time-realigned version of the output from the *hair-cell response* step (audio-rate firing rates) with one firing-rate stream for each channel.

4.2.5 Low-Pass Filter

The *low-pass filter* step consists of using a second-order, infinite-impulse-response (IIR), low-pass filter suggested by Slaney [73] with a cutoff frequency set to a normalized value of $\frac{f_s}{2R_d}$, where f_s is the sample rate and R_d is the decimation ratio used in the next step. This prepares the channels of time-realigned hair-cell firing rates to be decimated in the next step. The output of this step is effectively channels of an audio-rate firing-rate envelope.

4.2.6 Copy with Time Shift and Decimation

The *copy with time shift and decimation* step consists of first making N_{st} copies of the channel-time-step matrix of the hair-cell response, then shifting each of those copies earlier in time an amount $\frac{n}{N_{st}}$ of the window time length, where n is the copy number and N_{st} is the number of copies. The copies are created in order to be individually processed and then averaged back together in the *means of the copies* step of the PPA. This averaging process removes the dependency on analysis start time for PPA. $N_{st} = 10$ copies were found to be a good balance between elimination of start-time dependency and computational demands. This copy-and-average process is similar to taking a fast Fourier transform (FFT) with window overlap set to $\frac{N_{st}-1}{N_{st}}$ of the number of points in the FFT (90% for $N_{st} = 10$).

The final part of this step is a decimation of the hair-cell responses. The decimation is done according to the parameter `windowTimeLen`, which is the length of time used for the following *differentiation* step. The decimation has a two-fold purpose: it both reduces the amount of data which must be subsequently processed and provides half of the mechanism by which the subsequent numerical differentiation occurs.

The output of this step is effectively N_{st} copies (with slightly differing start times) of all channels of a slower-than-audio-rate firing-rate envelope. Because of the *low-pass filter* step, these envelopes have been properly filtered for the new sampling rate of $\frac{1}{\triangle t}$ where $\triangle t$ is determined by the parameter `windowTimeLen`.

4.2.7 Differentiation

The *differentiation* step consists of taking the numerical derivative of the firing-rate envelopes with respect to time in order to determine when and how fast each envelope is rising and falling. The simple numerical formulation [82] of

$$\frac{df}{dt} \approx \frac{f(t + \triangle t) - f(t)}{\triangle t} \tag{4.1}$$

is used to calculate the slope. Differentiation also has the effect of high-pass filtering, which is a benefit; the faster the rise of the firing-rate envelope, the faster the rise time of the sound event causing the rise. The output of this step is N_{st} copies of all channels of the firing-rate envelope slopes.

4.2.8 Half-Wave Rectification

The *half-wave rectification* step consists of setting all negative values of the firing-rate envelope slopes to zero. This rectification is done in order to only detect the rise time (positive slope) of sound events. These positive slopes are an indication of percussivity, so the output of this step is N_{st} copies of the percussivity in all channels.

4.2.9 Channel-Group Means

The *channel-group means* step consists of simply grouping the channels into some number of channel groups. The number of channel groups is determined by the parameter nChannelGroups. The channels are grouped into channel groups in increasing order as evenly as possible, but for any percussivity-profile calculation, the channel groups may not be comprised of an equal number of channels due to integer division. Because the psychoacoustic frequency distribution was used in the *cochlear response* step, the frequency distribution in the channel-group means is still psychoacoustically motivated. For each channel group, the mean percussivity value is calculated from the grouped channels. At this point the percussivity values in the individual channels are no longer passed forward, and all percussivity values are contained only in the channel-group means. The output of this step is N_{st} copies of the channel-group percussivity means.

4.2.10 Upsample and Unshift

The *upsample and unshift* step consists of first duplicating each percussivity value within a channel-group N_{st} times (the number of copies created in the *copy with time shift and decimation* step). This is sample-and-hold interpolation and does introduce high-frequency noise at the transitions between hold values, though the means will be taken across copies which reduces this added noise. Each copy is then shifted n values later in time, where n is the copy number. This realigns the copies accurately according to their analysis start times. The output of this step is N_{st} copies of the start-time-realigned channel-group percussivity means at a new sampling rate of $\frac{N_{st}}{\triangle t}$.

4.2.11 Maximums of Channel-Group Means

The *maximums of channel-group means* step consists of calculating the maximum of the channel-group means across channel groups for each time window along the upsampled percussivity values. This step is only implemented for the SPP. The maximum is chosen because it appears that the gross spectral content generally does not determine the percussivity of a sound event, so high percussivity in any channel group could indicate high percussivity for the current time window in a piece of music. Choosing the mean here would have negated the effect of channel groups. (Taking the mean of means would be the same as taking the mean of the original values.) For each analysis-start-time copy, the maximums of the channel-group means over all time windows represents the percussivity profile of that copy. The output of this step is N_{st} copies of the maximum channel-group percussivity means.

4.2.12 Means of Copies

The *means of copies* step consists of averaging together the upsampled and appropriately time-shifted percussivity values across the N_{st} start-time copies. For the SPP, this averaging is done with the maximums of the channel-group means. For the GPP, this is done directly with the channel-group means before the maximums are calculated.

4.2.13 Removing Zero Padding

The *removing zero padding* step consists of removing the region of the percussivity profile corresponding to the zero values added in the *zero padding* step. The resulting output vector and matrix are, respectively, the SPP and GPP. The PPA also produces the analysis-window start times generated at the sampling rate of the percussivity profile, $\frac{N_{st}}{R_d} f_s = \frac{N_{st}}{\triangle t}$.

4.3 Tuning the Percussivity Profile Algorithm

The main motivation for creating the PPA was to produce an algorithm that would identify sound events in pieces of electroacoustic music that humans would label as percussive. In order for the PPA to return results that are guided by human choice, the PPA needed to be tuned according to a human metric of percussion. The results of the percussion-judgment collection in Chapter 3 provided that metric.

The following sections describe how the goal for human-performance matching was chosen as the measurement of success and the how the heuristic tuning method employed sought an optimal solution. It is worth noting that this ad hoc tuning method does not guarantee an optimal solution, but works well in practice.

4.3.1 Choice of Human-Performance Goal

In order to evaluate the performance of the PPA in comparison to human performance, a measurable goal for success was needed. Because the dimension of rise time was chosen as the fundamental audio dimension from the percussion-judgment collection in Chapter 3, the choice of human-performance goal needed to be focused on rise time.

Because rise time appears in both stimulus sets A and C as described in Section 3.1.1, the choice of human-performance goal for the PPA was to maximize P_{mAC}, the percentage similarity of choices with the participant modal choices for stimulus sets A and C. The sounds of stimulus set A were varied in the dimensions of rise time and string

Table 4.1: Names, variable types, and tuning ranges of the algorithm parameters in the PPA.

parameter	**type**	**minimum**	**maximum**
`nChannels`	integer	6	300
`lowFreq`	float	20 Hz	400 Hz
`windowTimeLen`	float	5 ms	200 ms
`hairCellScaling`	float	500	100000
`nChannelGroups`	integer	1	100

resonance. The stimulus sounds of stimulus set C were varied in the dimensions of rise time and gross spectral filtering. The sounds of stimulus set B did not include variation in rise time, so were not included in the human-performance goal.

The PPA was tuned using the participant's modal choices from the percussion-judgment collection. Some of the same software that was used to acquire the participant's pair choices was also used to acquire pair choices made by the PPA. The pair choice of the PPA was determined according to which of the pair-choice sounds had a higher percussivity rating. For a given stimulus set, the choices of the PPA were then compared to the participant's modal choices to determine success of the PPA for the particular set of algorithm parameters. The names, variable types, and tuning ranges of the algorithm parameters are listed in Table 4.1.

Although the ranked list of stimuli was generated just as for the human participants in the percussion-judgment collection, only the actual choices between pairs of sounds were used to evaluate the performance of the PPA. No mechanism was created to collect the threshold value for judging a sound event to be an SDPE from the PPA. This decision leads to a user-guided threshold mechanism which is described further in Section 4.5.

Two remarks should be made here. First, if human participants chose the two sounds in a pair an equal number of times, then there was no modal choice, and the PPA can never make a similar choice for that pair. This design decision avoids arbitrary matching. Second, the choice of P_{mAC} as human-performance goal does not use the does not use the participants' decision-difficulty ratings.

4.3.2 Tuning

In order to find the algorithm parameters that provided the most human-like performance from the PPA with a limited amount of computational effort, several methods from mathematical programming (optimization) were employed. Combining the methods of parameter exploration and line search [83] provided a mechanism to find consistently high values of P_{mAC}. The line-search method seeks the best result by varying only one parameter between two limits while holding the rest constant. The line-search method requires the assumption that the algorithm parameters have separable effects, which was not proven in this case, although the line-search method was still effective. The parameter exploration method employs simple parameter-space exploration guided by a user.

Table 4.1 shows the parameter values over which the tuning occurred. The parameter space was explored in several iterations. First a line search was done in turn for each of the five parameters and the results were examined. Parameter exploration then was employed to ensure that the appropriate choice was made for each parameter. Further iterations of the line-search technique then occurred, followed by parameter exploration based on those results, until a satisfactory and constant maximum for P_{mAC} was found.

Restrictions on the algorithm parameters affected the tuning procedure. The `nChannels` and `nChannelGroups` parameters must be integers with an additional restriction that `nChannels` $\geq$ `nChannelGroups`. The technique of relaxation was used, which, in this case, allows the requirements of integer values for integer parameters to be dropped while searching for parameter values, but returned to integer values for any specific calculation of P_{mAC}. The parameter space was restricted to regions where `nChannels` $\geq$ `nChannelGroups`. Also, because 120 discrete choices were made for two stimulus sets, the resulting value is not a continuous function of the independent variables (the algorithm parameters).

The best parameter set found through this tuning method is shown in Table 4.2. For this parameter set, the PPA made the same choice as the modal participant choice 95.83% of the time for stimulus set A and 95.00% for stimulus set C, with a mean of 95.42% for both sets. For comparison, the highest percentage any single participant made the same choice as the modal participant choice was 95.00% of the time for stimulus set A and 92.50% for stimulus set C, with a mean of 93.75% for both sets. The best mean for both stimulus sets for any individual, however, was 92.50%.

About 24 critical bands span the audible frequency range according to Rossing [9]. A set of 95 channels spanning the range from 115 Hz to 20 kHz is four times the

Table 4.2: Algorithm parameters tuned to provide a consistently high value of P_{mAC}.

`nChannels`	`lowFreq`	`windowTimeLen`	`hairCellScaling`	`nChannelGroups`
95	115 Hz	12.6 ms	9213	6

amount of critical bands for the same frequency band, but the channels, which follow the gammatone filters, are not intended to be critical bands. The bandwidth of the channels is a scaled version of the critical bandwidth at the same center frequency.

Example Tuning – `hairCellScaling`

Figure 4.6 shows the effect of `hairCellScaling` on P_{mAC}. This is one example of how the algorithm parameters affect the performance of the PPA. The rest of the parameters are set to `nChannels` = 95 channels, `lowFreq` = 115 Hz, `windowTimeLen` = 0.0126 seconds, and `nChannelGroups` = 6 groups.

According to Slaney [73], the value of `hairCellScaling` can be set with

$$\texttt{hairCellScaling} = 10^{\left(\frac{L_p}{20}-1.35\right)} \tag{4.2}$$

which comes from the standard formula for sound pressure level (SPL) [84],

$$L_p = 20\log_{10}\frac{p_{\mathrm{rms}}}{p_{\mathrm{ref}}} \tag{4.3}$$

where L_p is the SPL, p_{rms} is the root-mean-square (RMS) pressure, and p_{ref} is the standard reference pressure in air of 20 μPa. The normalization of -1.35 is added to the exponent in (4.2), according to the suggestion of Meddis [77] following the normalization in the work of Schroeder and Hall [85].

Because the stimulus sounds are percussive rather than continuous sounds, specifying the value of p_{rms} is not as useful as specifying the peak pressure, p_{peak}. For the formula above, p_{rms} is estimated to be $\frac{p_{\mathrm{peak}}}{2}$, because p_{rms} of a peak becomes dependent on the time window used to measure it, this procedure is also used when calculating `hairCellScaling` for pieces of music.

As stated in Section 3.1.2, the maximum SPL of the sounds presented to the percussion-judgment participants was around 83 dBA (re 20 μPa). The maximum p_{peak} of the normalized amplitudes of stimulus sets A and C was calculated to be 0.397. Using

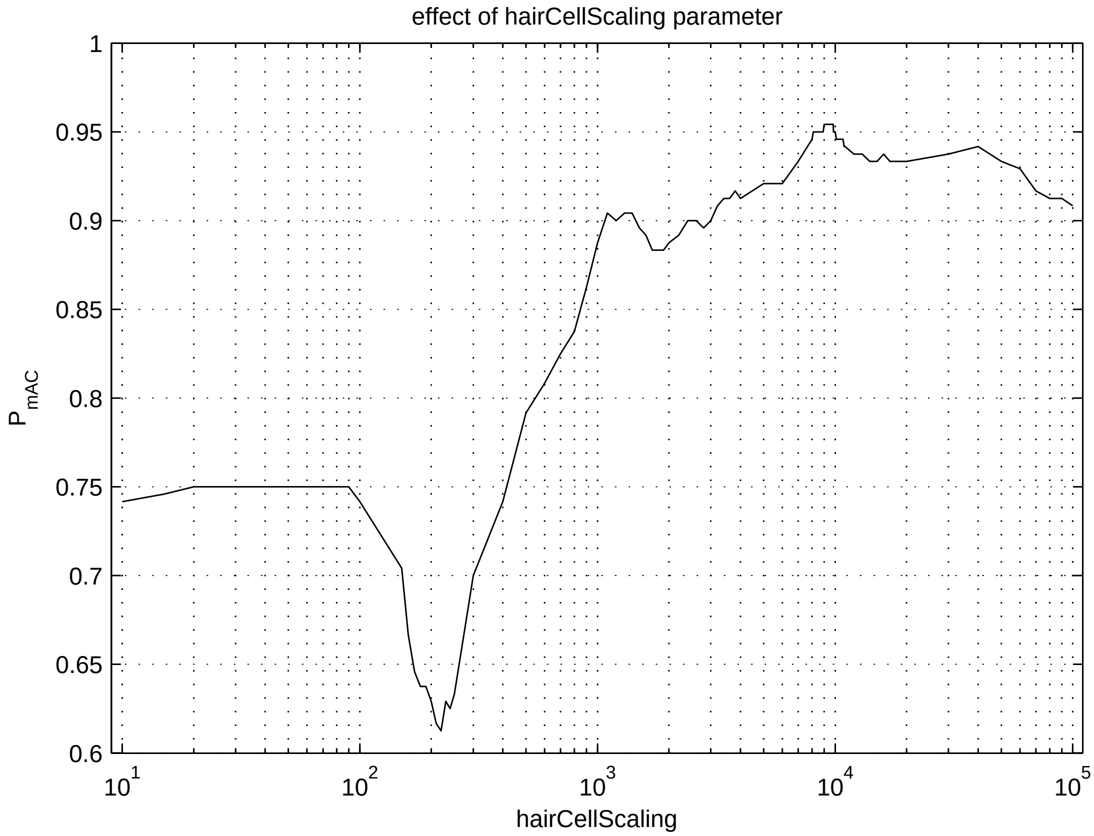


Figure 4.6: The effect of `hairCellScaling` on P_{mAC}. Other parameter values are `nChannels` = 95 channels, `lowFreq` = 115 Hz, `windowTimeLen` = 0.0126 s, and `nChannelGroups` = 6 groups.

the following equation,

$$\texttt{hairCellScaling} = \frac{10^{\left(\frac{L_{\text{peak}}}{20} - 1.35\right)}}{\frac{p_{\text{peak}}}{2}} \tag{4.4}$$

`hairCellScaling` was determined to be 3178 for the stimulus sets.

The value for `hairCellScaling` found through parameter tuning that gives the maximum P_{mAC} is 9213. The difference in values may be caused by the estimation of p_{rms} from p_{peak}.

Example Tuning – `nChannelGroups`

Figure 4.7 shows the effect of **`nChannelGroups`** on P_{mAC}. This is another example of how the algorithm parameters affect the performance of the PPA. In this case, five values centered around the best parameters are shown for each of **`nChannels`** (85, 90, 95, 100, and 105 channels). **`lowFreq`** (105, 110, 115, 120, and 125 Hz), **`windowTimeLen`** (0.0120, 0.0123, 0.0126 0.0129, and 0.0132 seconds), and **`hairCellScaling`** (7213, 8213, 9213, 10213, 11213).

There are several remarkable features of the effect of **`nChannelGroups`**. P_{mAC} increases monotonically as **`nChannelGroups`** increases to six groups in all cases. For any case in which P_{mAC} reaches the maximum value of 95.42%, that maximum value occurs at `nChannelGroups` = 6 groups. In almost all cases the value of P_{mAC} is maximum only for `nChannelGroups` = 6 groups. In most of the remaining cases, P_{mAC} is maximum at least at **`nChannelGroups`** = 6 groups. In a few cases, the values of P_{mAC} is not maximum at **`nChannelGroups`** = 6 groups. In those cases where P_{mAC} has a maximum value at a location other than **`nChannelGroups`** = 6 groups, the maximum value is found at `nChannelGroups` = 14 or 16 groups.

In this tuning process, **`nChannelGroups`** could vary in the range of 1 to 100 groups. If **`nChannelGroups`** = 1, this has the effect that the percussivity value is the maximum across all channels. While **`nChannels`** was at a value of 100 or fewer, it was possible for **`nChannelGroups`** = **`nChannels`**. This equality has the effect that the percussivity value is the mean across all channels. Because the maximum value of P_{mAC} is achieved for neither of these values of **`nChannelGroups`**, the use of the combination of the maximums and means in the PPA is validated.

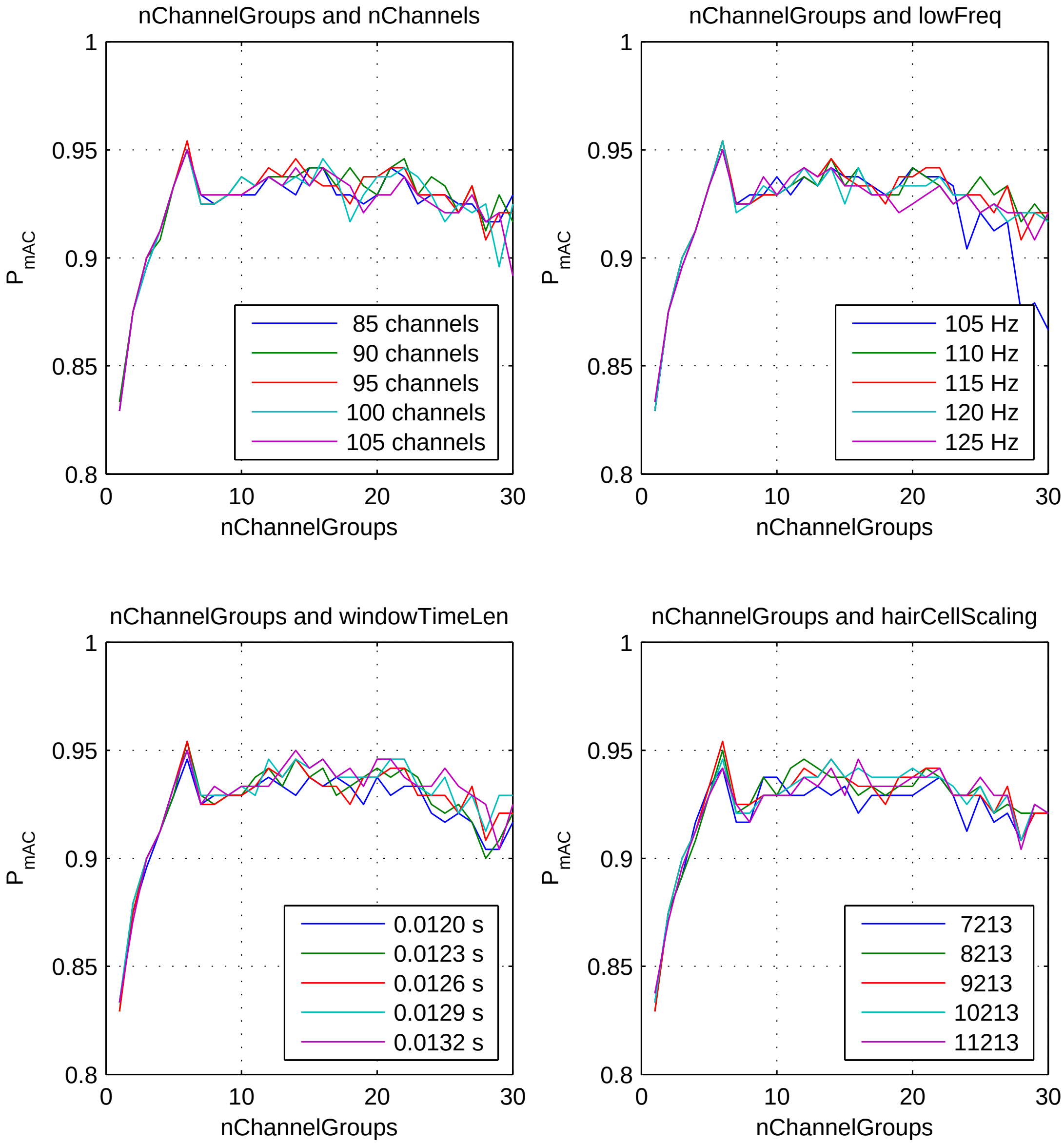


Figure 4.7: The effect of `nChannelGroups` on P_{mAC}. Unless specified, the other parameters are set to `nChannels` = 95 channels. `lowFreq` = 115 Hz, `windowTimeLen` = 0.0126 s, and `nChannelGroups` = 6 groups. The top left graph shows the effect of `nChannelGroups` on P_{mAC} for several values of `nChannels`, the top right graph for several values of `lowFreq`, the bottom left graph for several values of `windowTimeLen`, and the bottom right for several values of `nChannelGroups`.

4.3.3 Algorithm Development

The PPA is a culmination of much design and testing work. Several different choices were made during the development for the *time-realignment*, *low-pass filter*, and *copy with time-shift and decimation* steps. With each of these choices, the motivating factor was always the higher value of P_{mAC}. Each option was implemented and then the algorithm parameters were tuned. The option which provided the higher value for P_{mAC} was then chosen for the final design of the PPA.

4.3.4 Threshold

Figure 4.8 shows the percussivity profile of the stimuli from stimulus set A ordered in a sound file by decreasing SDPE-likeness according to the participant modal choices. The top plot shows the amplitude, the middle plot shows SPP_p, and the bottom plot shows the GPP. For the GPP, higher-numbered channel groups represent groups of lower-frequency channels shown at the bottom of the plot. Using number system A from Section 3.1.1, the order is 1, 5, 2, 9, 6, 13, 10, 3, 14, 7, 15, 11, 4, 8, 12, 16. As stated in Section 3.2.5, the modal choice for percussion threshold was the tenth stimulus. This threshold is indicated by the dotted line in the percussivity profile plot of Figure 4.8.

Figure 4.9 shows the percussivity profile of the stimuli from stimulus set C ordered in a sound file by decreasing SDPE-likeness according to the participant modal choices. Using number system A, the order is 5, 9, 1, 13, 10, 6, 2, 14, 7, 11, 3, 15, 12, 8, 4, 16. As stated in Section 3.2.5, the modal choice for percussion threshold was the eighth stimulus. This threshold is indicated by the dotted line in the SPP_p plot of Figure 4.9. The usefulness of these threshold values are discussed further in Section 4.4.1.

The percussivity-profile plots of both Figure 4.8 and Figure 4.9 show the general trend of higher to lower percussivity, although several of the individual stimuli in each set appear to be out of order according to the PPA. In the GPP plot of stimulus set A, little difference can be seen between adjacent stimulus sounds, but a general trend from higher values (darker) to lower value (lighter) for the stimulus sounds can be seen. In the GPP plot of stimulus set C, the differing frequency content of the stimulus sounds can be seen clearly by their GPP signature. By comparing the "spikes" of stimulus sounds with similar GPP signatures from left to right, the general trend from higher values (darker) to lower values (lighter) also can be seen (for example, the 4th, 8th, 12th, and 16th spikes show this trend).

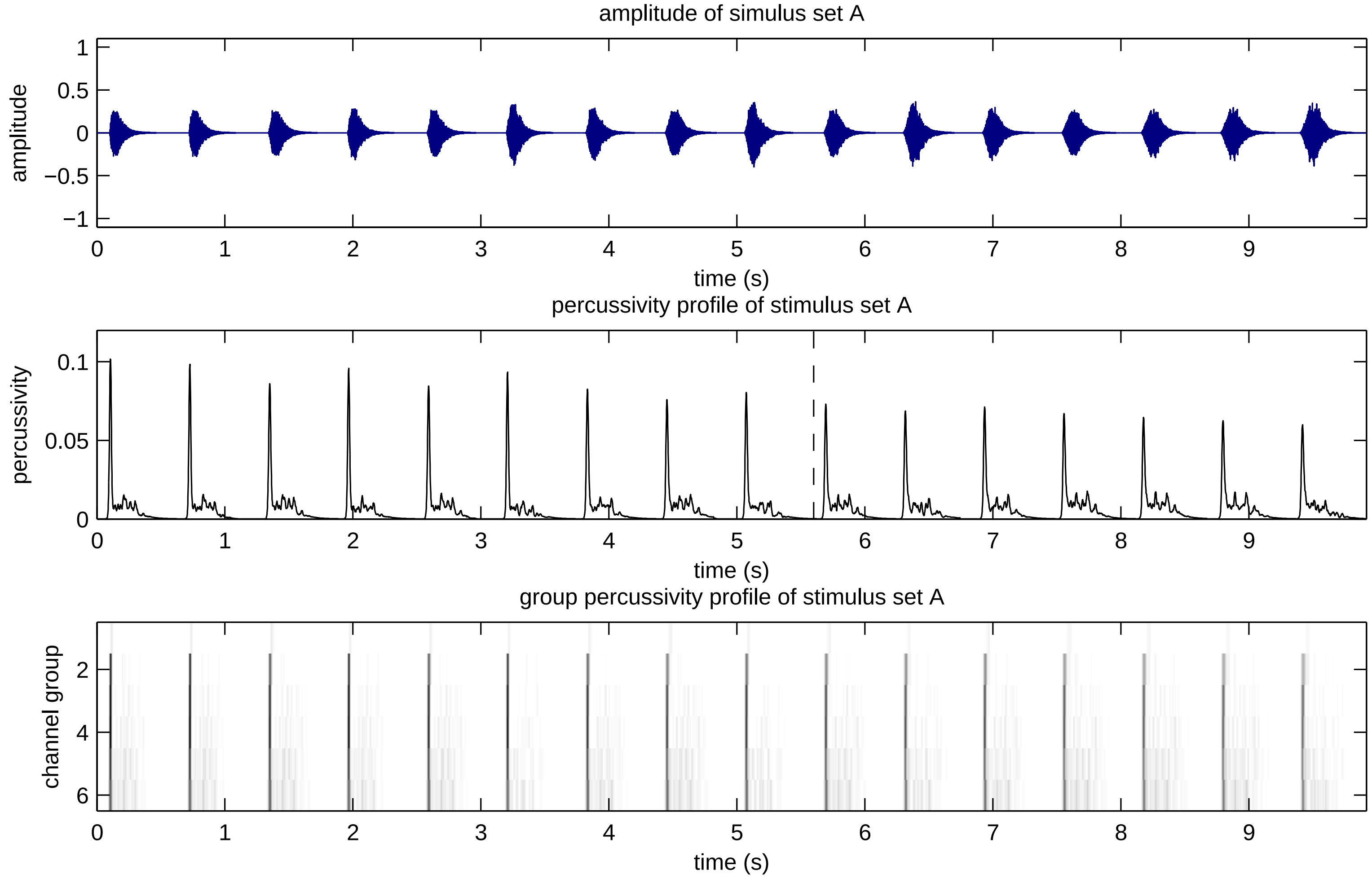


Figure 4.8: Amplitude (top plot), SPP_p (middle plot), and GPP (lower plot) of the ranked list of participant modal choices in stimulus set A.

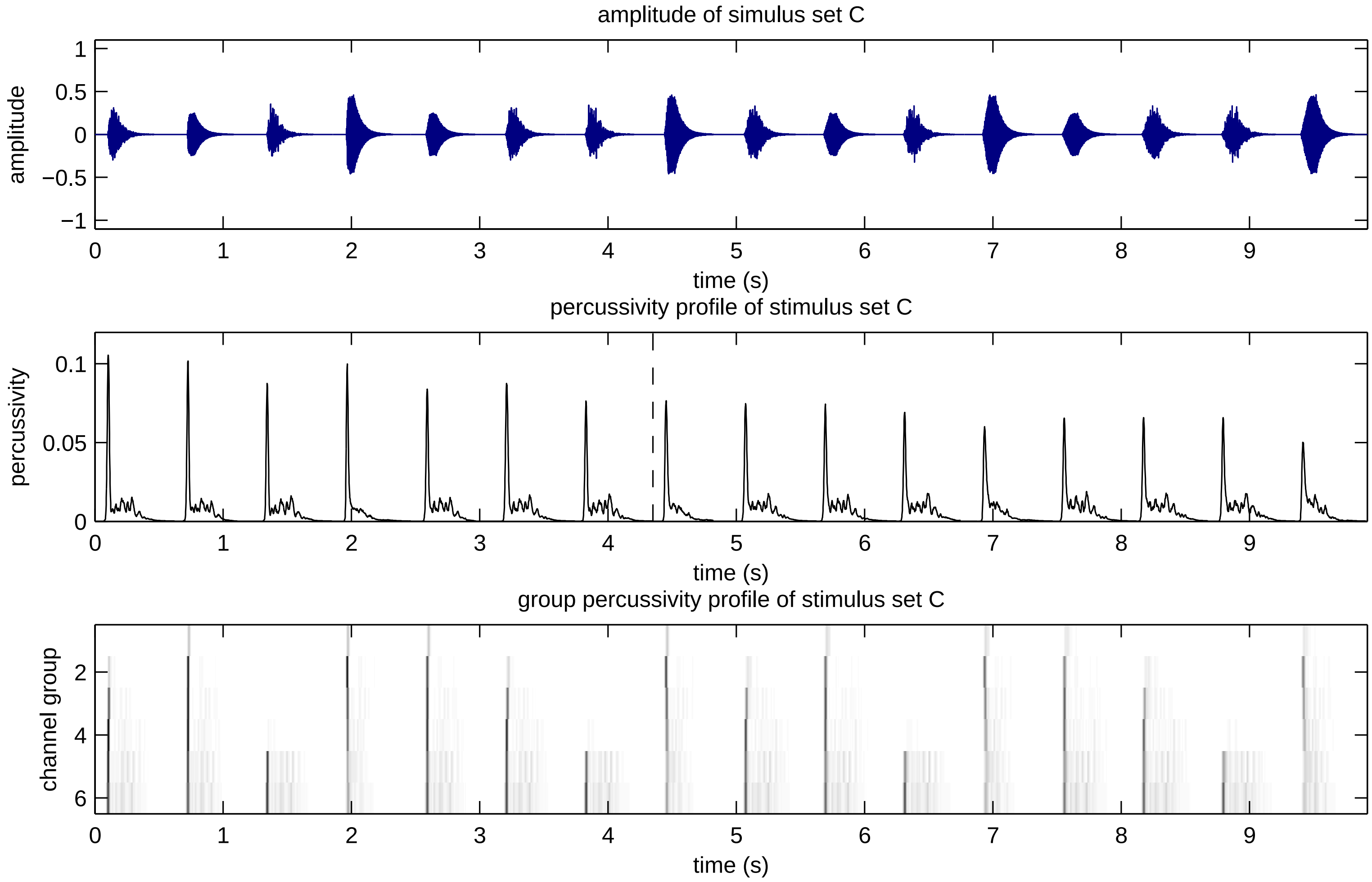


Figure 4.9: Amplitude (top plot), SPP_p (middle plot), and GPP (lower plot) of the ranked list of participant modal choices in stimulus set C.

4.4 Practical Considerations

Analyzing a real piece of music is significantly different from analyzing isolated percussive events. The presence of non-percussive sounds adds complexities to identifying sounds as percussive, so some post-processing is helpful in making the percussive sounds more obvious. The arbitrary output level of the PPA is also an issue for the choice of threshold value. The length of the sound file causes issues with memory usage and computer runtime. The next sections describe the practical considerations that affect the analysis of real music.

4.4.1 Post-Processing Manipulation

The SPP represents the grouped and averaged nerve spike potentials along the cochlear nerve. This representation is not the most visually useful representation of the sound in a piece of music. Some further processing enhances the indication of percussive sounds.

The final value of the post-processed SPP is

$$SPP_p = \frac{SPP^2}{SPP_{\mathrm{rms}}} \tag{4.5}$$

where SPP_p is the post-processed SPP, SPP is the raw SPP, and SPP_{rms} is the RMS value of SPP over a short window.

In order to make the peaks in the percussivity profile more visually apparent, the values were squared. This manipulation separates the high values from the noisiness of the non-percussive sounds represented in the SPP. In order to display the peaks in relation to the surrounding noise of non-percussive sound in the SPP, the value of SPP^2 is divided by SPP_{rms}.

4.4.2 Threshold

A sound event is labelled as percussive according to a threshold value of SPP_p, and any peaks that lay above that value are considered percussive. The threshold data were collected as shown in Section 3.1.2 and relate to the stimuli as shown in Section 4.3.4. When analyzing percussive sounds within pieces of music, however, the threshold for isolated, percussive sounds is not particularly useful. This fact is apparent in Section 4.5, where the threshold value for SPP_p is discussed further. An appropriate threshold value for SPP_p must be chosen in a user-guided fashion based on what percussive aspects the user wants to display while avoiding false positive identification.

Once again, the actual output values from the PPA are “arbitrary,” although stable for any set of given algorithm parameters. This means that for any new set of parameters and probably for any new piece of music, a new threshold value will need to be chosen. Threshold choice is discussed more in Section 4.5.

4.4.3 Block Processing

Although much effort went into keeping memory usage to a minimum, at one point the algorithm requires approximately

$$M_{max} = 17.6\, N_s N_c N_{st} \tag{4.6}$$

bytes, where M_{max} is the maximum memory required, N_s is the number of samples in the sound file, N_c is `nChannels`, and N_{st} is the number of copies from the *copy with time shift and decimation* step from Section 4.2.6. For the values of $N_c = 95$, and $N_{st} = 10$, Equation 4.6 reduces to $M_{max} = 16720\, N_s$ bytes. This translates to about 1.46 seconds of monaural music sampled at 44.1 kHz which can be processed for every 1 GB of free memory.

The following computer run times are given as a base line performance of the PPA, and can be used to anticipate performance on other computer systems. On a Dell Precision 530 workstation with two Intel 32-bit Xeon 1.80 GHz processors, 1 GB of RAM, and running Ubuntu Linux 7.10, typically 500 MB of RAM was free for processing, and one second of music could be processed in approximately 19.4 seconds of real time. On a Dell Poweredge 2950 rack mount workstation with two 3.75 GHz 64-bit Intel Dual-Core Xeon processors, 8 GB of RAM, and running Ubuntu Linux 7.04, typically 7 GB of RAM was free for processing, and one second of music could be processed in approximately 7.13 seconds of real time.

Although MATLAB can process data structures up to the sum of free RAM and free virtual memory available, doing so obviously requires the use of virtual memory. Using virtual memory incurs a significant speed reduction that can reach a slowdown of several orders of magnitude when moving beyond the boundary of free memory. Given the speed benefit of avoiding using virtual memory, it was necessary to create a function that processes longer pieces of music in smaller blocks and then reconnects the resulting shorter percussivity profiles into a single percussivity profile. The block size is determined using Equation 4.6 in order to use less than the amount of available free memory.

4.5 Results

The following sections describe how the PPA performed when analyzing one constructed example and two examples of electroacoustic music: a section of “Jeux Imaginaires” by Åke Parmerud [86] (labelled as the easy example), and a section of “Le Vertige Inconnu” by Gilles Gobeil [87] (labelled as the difficult example). The constructed example is presented in order to provide at least one objective measure of the performance of the PPA. The easy example was chosen as an example of music in which the percussive sounds are clearly percussive, and would be fairly easy for the PPA to analyze. The difficult example was chosen as an example of music in which the percussive sounds are obscured or not clearly percussive, and would point out some of the problems the PPA might have. All of these examples are monaural audio (reduced from stereo by averaging if necessary) and recorded at a 44.1 kHz sampling rate with 16 bit resolution.

4.5.1 Constructed Example

This section describes the performance of the PPA on a constructed example. This constructed example consists of a sound mixture of a short section (from 6:57.03 to 7:21.36) of “2/1” by Brian Eno [88] and a series of decreasing amplitude repetitions of stimulus 1 from stimulus set A (see Section 3.1.1). This section of “2/1” was chosen because it does not contain any percussive sounds at all. This constructed example is presented in order to provide an objective measure of the performance of the PPA. The analysis of the two sound-mixture components are presented individually first, followed by the mixture.

Figure 4.10 shows the section of “2/1.” This section of this piece of music consists of vocal sounds layered on top of one another and none of the sounds are considered percussive. A 0.5 second amplitude fade-in is used at the beginning and a 0.5 second amplitude fade-out is used at the end. The top plot of Figure 4.10 shows the waveform amplitude of the section, and that the maximum normalized amplitude of this section of music is 0.5. The middle plot shows SPP_p of the same section of music. The maximum value of SPP_p in this nonpercussive example is 0.0752; this value affects the choice of threshold level when searching for percussive sound events in the upcoming sound mixture. The bottom plot shows the GPP for the same section of music. The short time window used to calculate SPP_{rms} was 0.5 seconds.

Figure 4.11 shows 37 decreasing amplitude repetitions of stimulus 1 from stimulus set A. The top plot shows the waveform amplitude of the sound. The maximum normalized amplitude of the repetitions is 0.5 and the repetitions decrease in linear dB amplitude

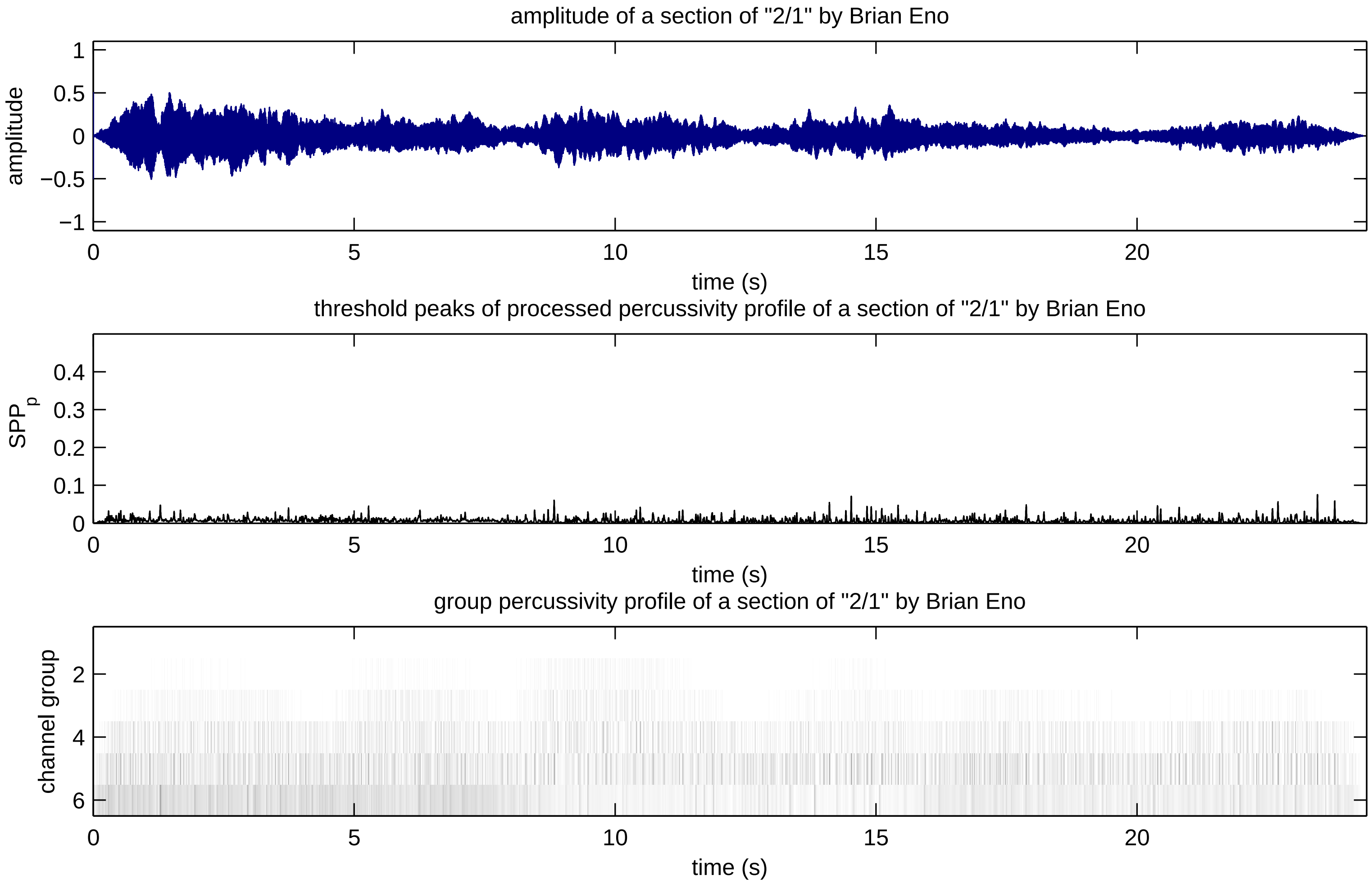


Figure 4.10: Amplitude (top plot), SPP_p (middle plot), and GPP (lower plot) of a section (from 6:57.03 to 7:21.36) of “2/1” by Brian Eno [88]. The maximum value of SPP_p in this nonpercussive mixture is 0.0752 and represents the lowest threshold value at which events would be identified as percussive. SPP_{rms} was taken over a 0.5 s window.

until repetition 37, which is 25 dB lower than repetition 1. The middle plot shows SPP_p of the same sound. The peaks that are identified as percussive events are marked with circles and are determined by a simple threshold value of 0.076 for SPP_p. The bottom plot of Figure 4.11 shows the GPP for the same sound. The short time window used to calculate $\mathit{SPP}_{\mathrm{rms}}$ was 0.5 seconds.

The threshold value of 0.076 is used in order to discriminate from the maximum nonpercussive levels (0.0752) seen in Figure 4.10. This threshold value causes the PPA to identify 26 percussive events in the stimulus repetitions. The amplitude of repetition 26 is 17.25 dB lower than repetition 1. In this isolated sound environment, a lower threshold could be used to identify more percussive events, but in context of the task of percussion detection in the upcoming sound mixture, the current threshold value is appropriate.

Figure 4.12 shows the sounds of Figure 4.10 and Figure 4.11 mixed by straightforward sample by sample addition. The top plot of Figure 4.12 shows the waveform amplitude of the sound. The middle plot shows SPP_p, the post-processed percussivity profile of the same sound mixture. The peaks that are identified as percussive events are marked with circles and were determined by a simple threshold value of SPP_p. As in Figure 4.11, a threshold value of 0.076 is used. The bottom plot of Figure 4.12 shows the GPP for the same sound. The short time window used to calculate $\mathit{SPP}_{\mathrm{rms}}$ was 0.5 seconds.

The threshold value of 0.076 causes the PPA to identify 21 stimulus repetitions as percussive events. The amplitude of repetition 21 is 13.80 dB lower than repetition 1. Identifying 21 stimulus repetitions as percussive events represents a performance decrease of 19% (21 out of 26 events found) from the isolated performance (as long as humans would identify the 5 unidentified stimulus repetitions as percussive in the new context of the sound mixture).

In Figure 4.12, if the sound of “2/1” is considered noise while trying to identify the signal of percussive events, then stimulus repetition 1 in the sound mixture has a signal-to-noise ration (SNR) of 1:1 or 0 dB. Repetition 21 has an SNR of approximately 1:5 (0.204) or -13.80 dB. Because repetition 26 of the isolated stimulus repetitions was able to be identified at a level of -17.25 dB compared to the level of event 1, the noise of “2/1” is blocking the ability of the PPA to identify percussive events at a level of 3.45 dB.

4.5.2 Easy Example

Figure 4.13 shows a section (from 3:12.41 to 3:43.26) of the piece of music “Jeux Imaginaires” by Åke Parmerud [86]. This section of this piece of music was chosen because the percussive sounds in it are clearly percussive and therefore should be easily identified by

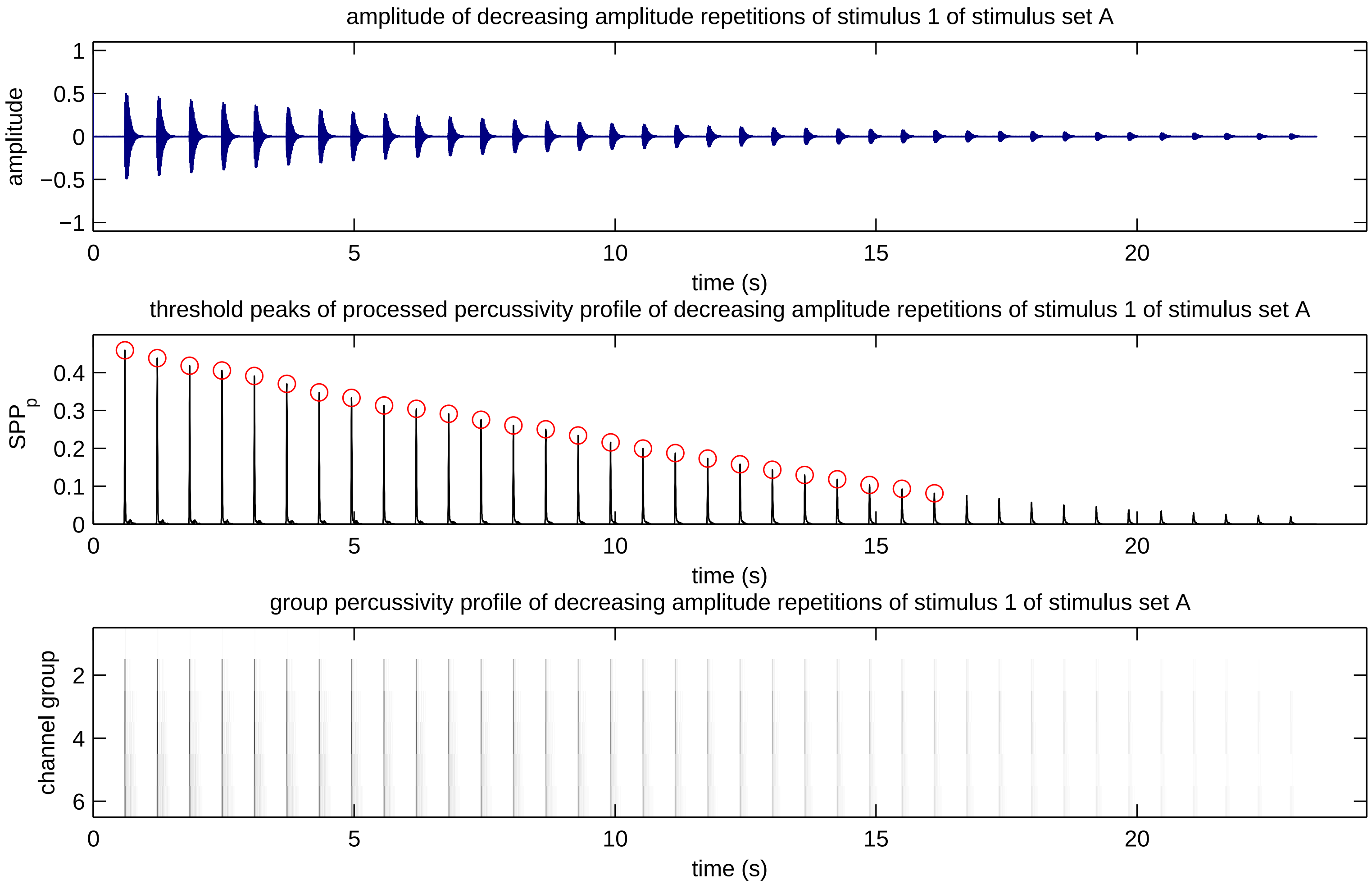


Figure 4.11: Amplitude (top plot), SPP_p (middle plot), and GPP (lower plot) of repetitions of stimulus 1 from stimulus set A at decreasing amplitude. The circled peaks in the SPP_p plot represent percussive instants according to a threshold value of 0.076. SPP_{rms} was taken over a 0.5 s window.

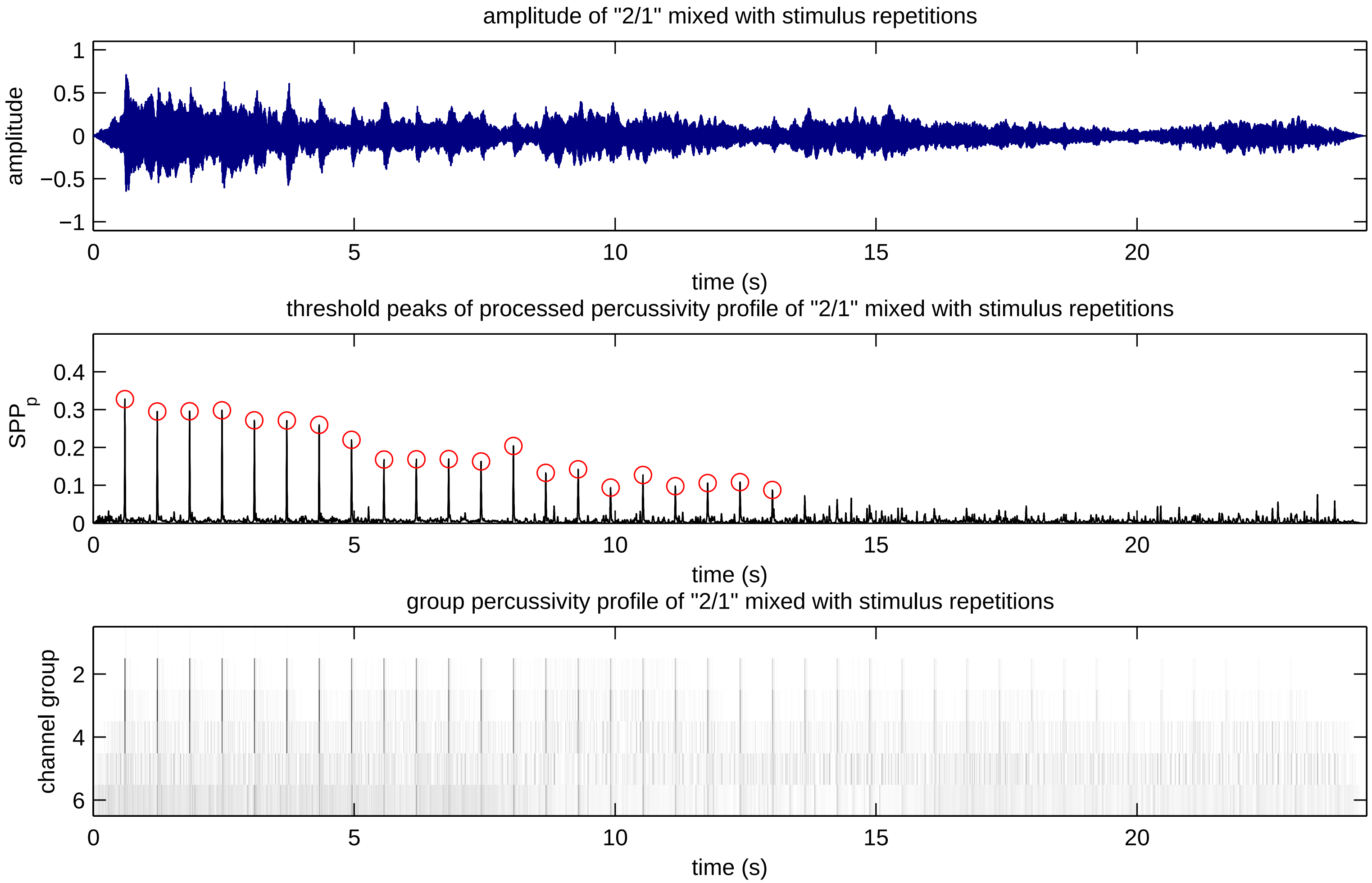


Figure 4.12: Amplitude (top plot), SPP_p (middle plot), and GPP (lower plot) of a mixture of a section "2/1" by Brian Eno and repetitions of stimulus 1 from stimulus set A. The circled peaks in the SPP_p plot represent percussive instants according to a threshold value of 0.076. SPP_{rms} was taken over a 0.5 s window.

the PPA. Real-world examples often provide more difficulties than anticipated, however, and this section of real electroacoustic music is no exception. The beginning of the example was modified by a fade-in amplitude ramp in order to avoid the sudden onset of sound at an arbitrary cut being identified as a percussive sound.

The top plot of Figure 4.13 shows the waveform amplitude of the section of "Jeux Imaginaires." The middle plot shows SPP_p of the same section of music. The peaks that were identified as percussive events are marked with circles and were determined by a simple threshold value of SPP_p. A threshold value of 0.065 was used for this example. The bottom plot shows the GPP for the same section of music. The short time window used to calculate SPP_{rms} was 0.5 seconds.

Most of the sudden amplitude increases in the top plot represent percussive sounds according to the percussivity profile. The two shaded regions in Figure 4.13 will be discussed in more detail in the following paragraphs. It is worth noting that some of the sudden amplitude increases are not identified by the PPA as being as strongly percussive or even percussive at all, and are discussed in the following paragraphs as well.

Figure 4.14 shows more detail of Region 1 (from 0:05.00 to 0:07.00) in the amplitude plot of Figure 4.13. Within this section of music five percussive sound events are clearly identified. The sound events occur at a very regular interval and sound as if perhaps they are an artificially truncated strike of a drum stick on a wood block.

At roughly halfway between the second and third, the third and fourth, and the fourth and fifth identified percussive sound events are sudden amplitude increases that are not identified as percussive sound events. These amplitude increases sound as if they are perhaps artificially truncated strikes of marbles against one another, and might be identified as percussive by a listener. These sound events, however, are approximately 0.0025 seconds in length, which is only 20 percent of the time window (0.0126 seconds) being used to analyze the section of music and is only 25 percent of the shortest rise time (0.010 seconds) tested in the percussion-judgment collection. These sound events were clearly beyond the capabilities of the PPA to identify as percussive, and are perhaps sound events that press the limits of what humans define as percussive.

Figure 4.15 shows more detail of Region 2 (from 0:08.75 to 0:11.50) in the amplitude plot of Figure 4.13. Within this section of music, seven percussive sound events are clearly identified. The sound events at the beginning and end of this section sound like a drum being struck, and the five sound events in the middle are similar to the truncated wood-block strikes from Region 1 (see Figure 4.14).

A few important traits of the algorithm are highlighted in Region 2. It is notable

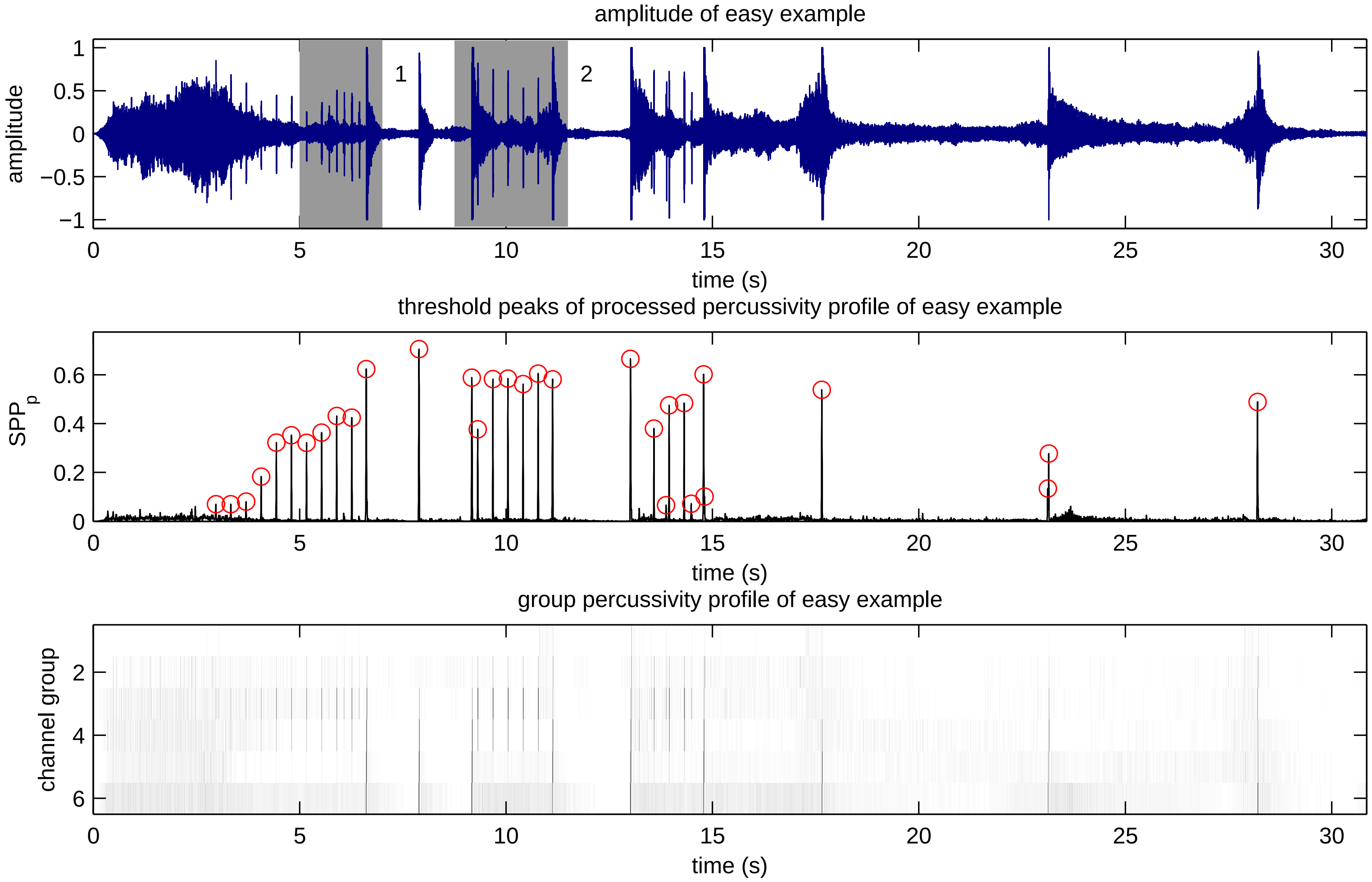


Figure 4.13: Amplitude (top plot), SPP_p (middle plot), and GPP (lower plot) of a section (from 3:12.41 to 3:43.26) of "Jeux Imaginaires" by Åke Parmerud [86]. The circled peaks in the SPP_p plot represent percussive instants according to a threshold value of 0.065. SPP_{rms} was taken over a 0.5 s window.

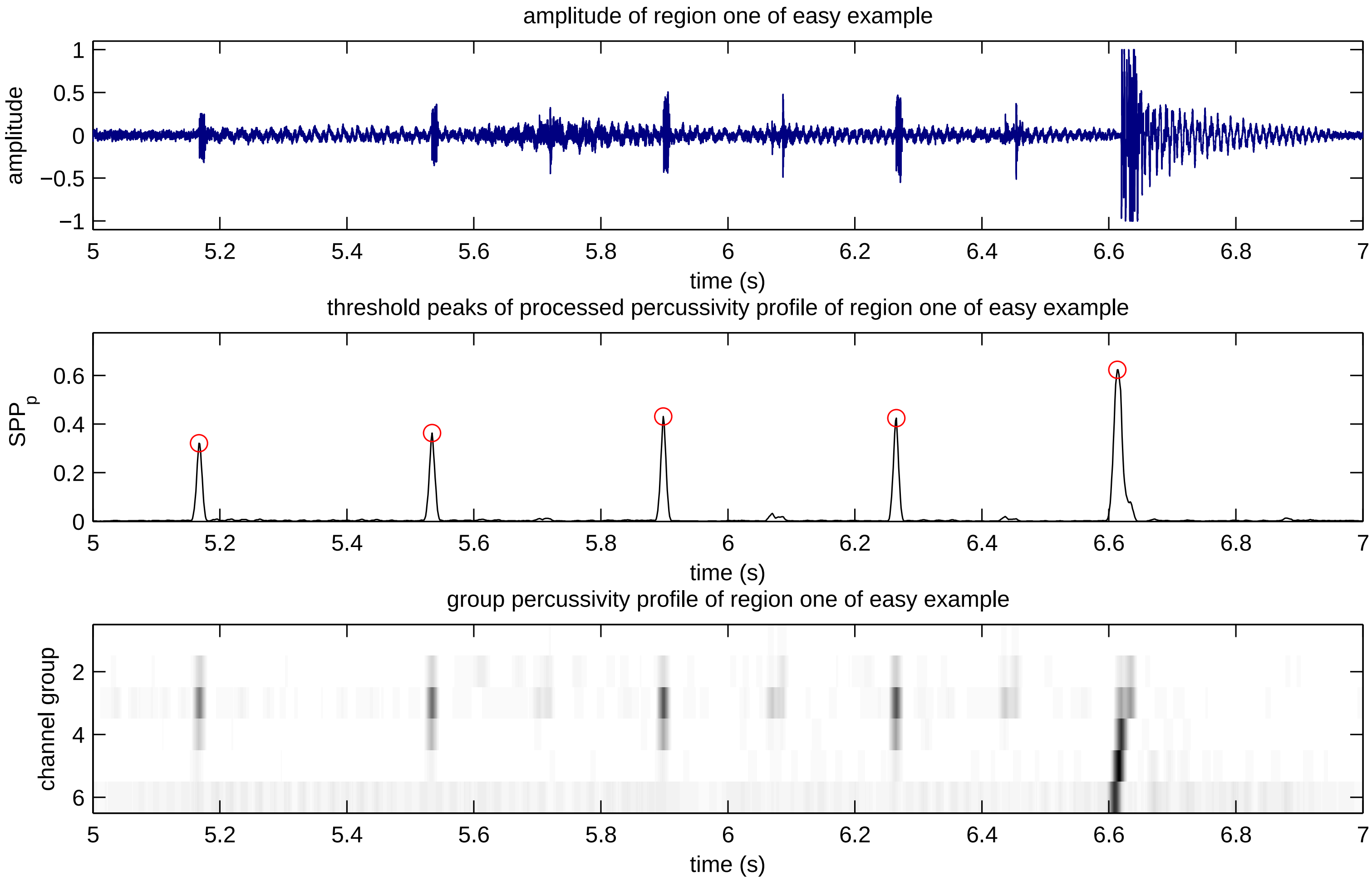


Figure 4.14: Amplitude (top plot), SPP_p (middle plot), and GPP (lower plot) of Region 1 of a section (from 3:12.41 to 3:43.26) of “Jeux Imaginaires” by Åke Parmerud [86]. The circled peaks in the SPP_p plot represent percussive instants according to a threshold value of 0.065. SPP_{rms} was taken over a 0.5 s window.

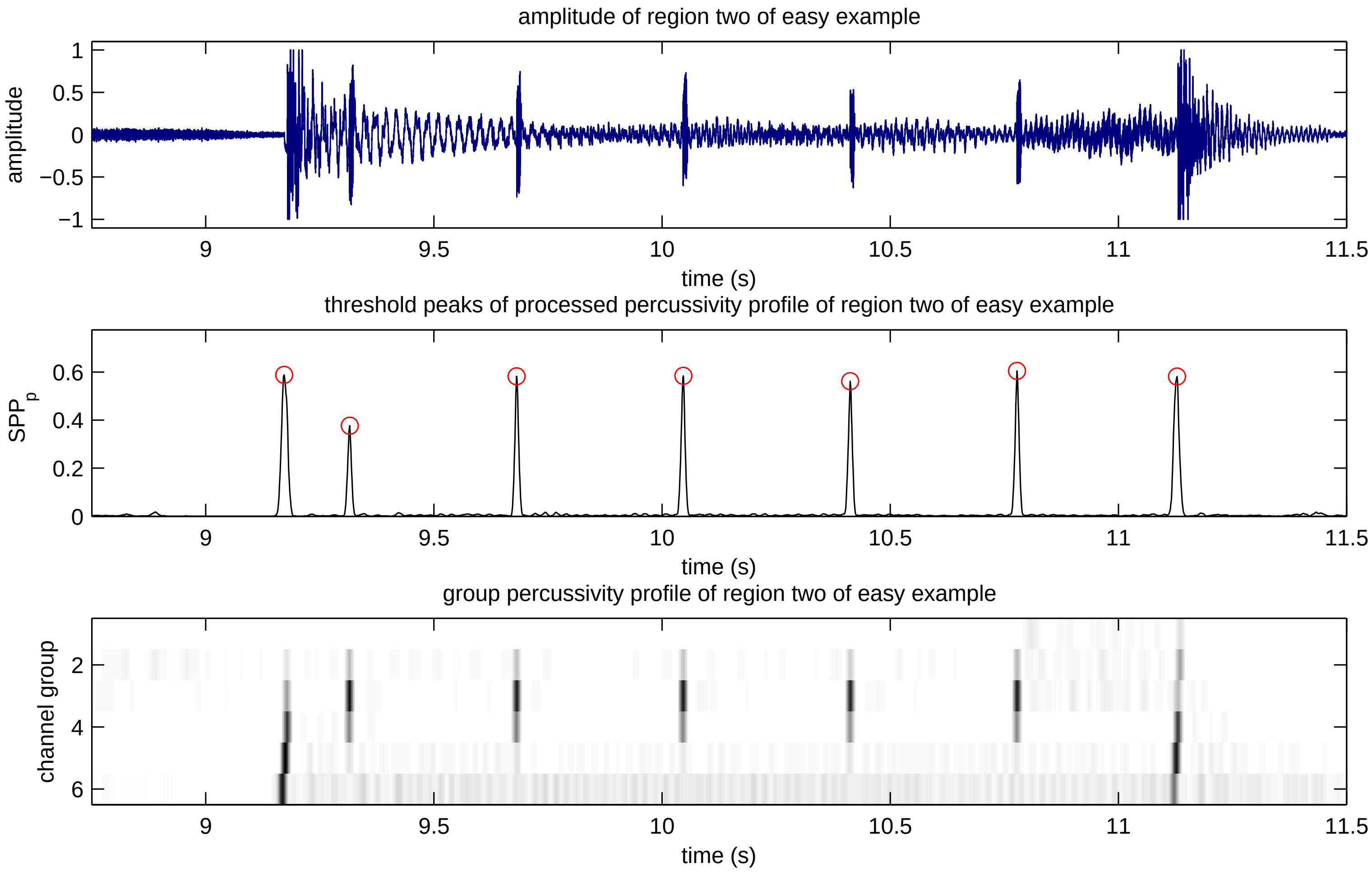


Figure 4.15: Amplitude (top plot), SPP_p (middle plot), and GPP (lower plot) of Region 2 of a section (from 3:12.41 to 3:43.26) of “Jeux Imaginaires” by Åke Parmerud [86]. The circled peaks in the SPP_p plot represent percussive instants according to a threshold value of 0.065. SPP_{rms} was taken over a 0.5 s window.

that the first wood-block strike (the second sound event identified as percussive in this example) is found to be percussive even though it is situated in other noise. It is also notable that the last percussive sound event in this example has noise leading up to it. Because the PPA processes channel groups separately, if a percussive sound event contains frequency components in a different group from the other noise in which it is situated, then the percussive sound event is still identified.

4.5.3 Difficult Example

Figure 4.16 shows a section (from 4:22.93 to 5:11.30) of "Le Vertige Inconnu" by Gilles Gobeil [87]. This section of music contains sound events and sound mixtures that press the limits of what is percussive, both in terms of the stimulus sounds used to tune the PPA and what humans might label as percussive. This section of music is intended to demonstrate some of the problems associated with identifying percussive sound events in real electroacoustic music.

In Figure 4.16, the top plot represents the waveform amplitude, the middle plot shows SPP_p, and the bottom plot shows the GPP for the piece of music. No fade-in amplitude ramp was necessary in this example because the initial amplitude of the example is so low. The peaks in SPP_p that have been identified by the PPA as percussive are marked with a circle, and this identification was done by simple thresholding SPP_p. A threshold value of 0.044 was used for this example. The short time window used to calculate SPP_{rms} was 0.5 seconds.

Most of the sudden amplitude increases in the top plot represent percussive sound events according to the percussivity profile; however, there are some significant counterexamples. The two shaded regions in Figure 4.16 will be discussed in more detail below. Once again, the PPA identifies some of the amplitude increases as not being strongly percussive and some as not percussive at all.

Perhaps most significantly, the large-amplitude sound between 10 and 12.5 seconds in the amplitude plot was not identified by the PPA as percussive. This section of sound could be described as the spinning up of a motor, along with some knocks and rattling, and finally the slam of a heavy door all taking place in a reverberant environment. The last component of this section, the door slam, seems as if it should be identified as a percussive event.

There are several possible reasons that the PPA does not identify the door slam as a percussive event: there may be too much competing noise across the frequency spectrum for the door slam to be identified separately, the tuning of the algorithm parameters may

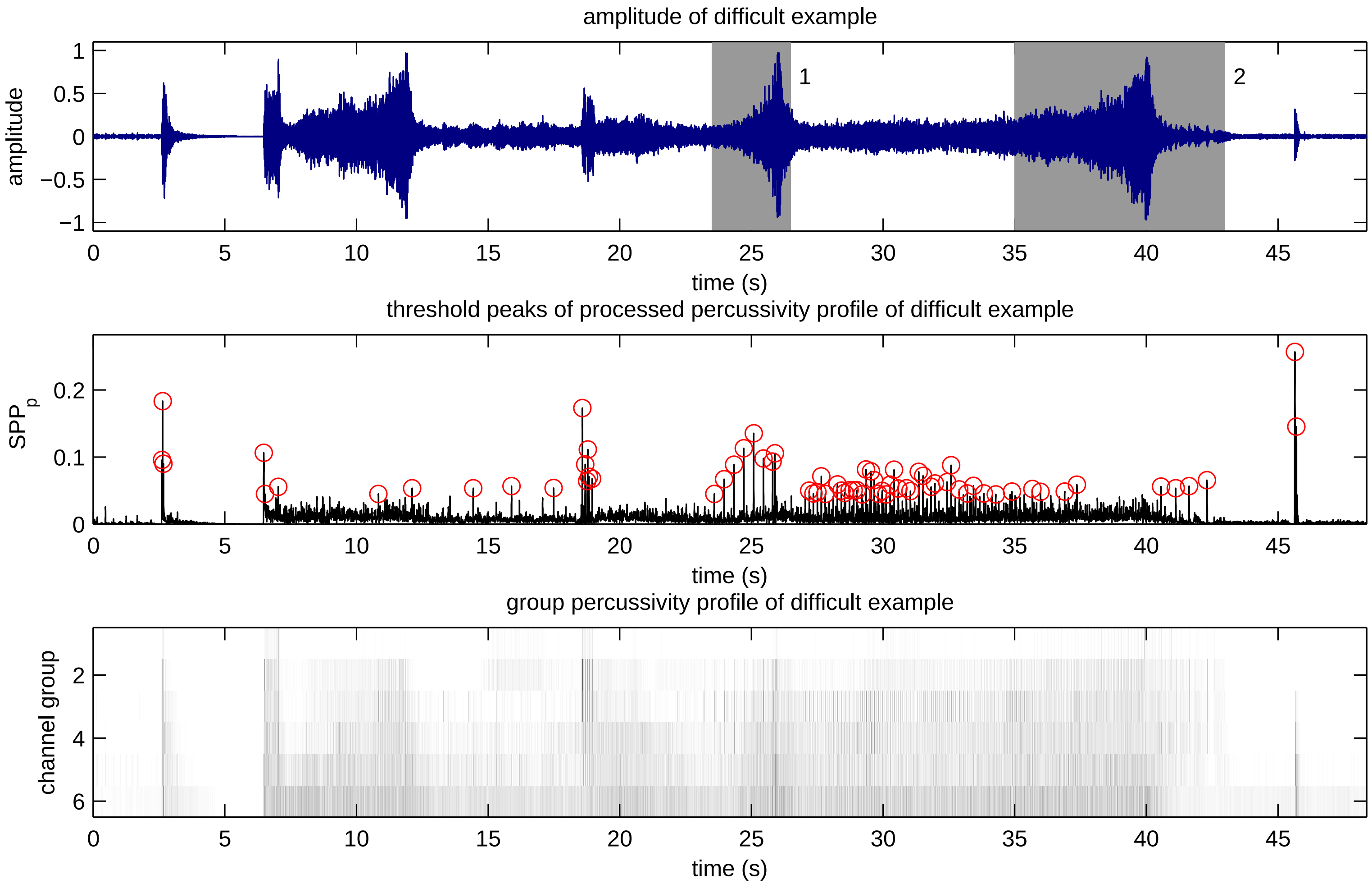


Figure 4.16: Amplitude (top plot), SPP_p (middle plot), and GPP (lower plot) of a section (from 4:22.93 to 5:11.30) of "Le Vertige Inconnu" by Gilles Gobeil [87]. The circled peaks in the SPP_p plot represent percussive instants according to a threshold value of 0.044. SPP_{rms} was taken over a 0.5 s window.

be inappropriate for this specific sound to be identified as a percussive event, it might be a completely different mechanism (for example, dynamic filtering) that would cause humans to identify this sound as percussive, or this sound event might not be identified as percussive by humans at all. Regardless of the reason, this sound event does point out challenges for the PPA.

Figure 4.17 shows more detail of Region 1 in the amplitude plot of Figure 4.16. In this region, eight percussive sound events are identified. The first six are perhaps strikes of a bell and the next two are perhaps truncated balloon pops. This region also is full of other non-percussive sounds, so it is remarkable that the PPA is able to extract the bell sounds as percussive amidst the milieu of other sounds occurring. This is an example of the effectiveness of using channel groups to provide percussion detection in spectral regions.

Figure 4.18 shows more detail of Region 2 in the amplitude plot of Figure 4.16. In this region, eight percussive sound events are identified. The analysis of this region of music is divided into four subregions.

Subregion A occurs from 35.0 until about 38.0 seconds. During this subregion many different sources of sound are contributing to the general noise. The identifiable sources are a buzzing horn, crickets, and what seems to be several machines tapping out a rhythm of some sort. Certainly some of the sounds of the machine rhythm would be considered percussive if heard in isolation, but the quick repetition of these sounds creates the effect of static and mechanical noise. Within this intense jumble of sounds it is difficult to focus on any one percussive event. In this subregion, however, the PPA identifies four percussive sound events.

Subregion B occurs from around 38.0 until around 39.8 seconds. In this subregion first the sound of the buzzing horn becomes louder, precluding the PPA from finding any more percussive sound events in the machine rhythm. At around 39.0 seconds, a new sound event, which sounds somewhat like a subway car approaching, increases the musical intensity. In this region no percussive sound events are found.

Subregion C occurs from around 39.8 until around 40.5 seconds. In this subregion a sound mixture occurs which is arguably a percussive event, but is not identified in the least as a percussive event by the PPA (no percussive sound events are identified by the PPA in this subregion). The sounds in this subregion appear to be pressurized air being released and then the closing of a metal gate mixed with a chorus of human voices saying the syllable, “cho.” The pressurized air sound is mixed with all of the previous sounds until the sound mixture of the gate closing and choral ”cho” occurs. At this point, many

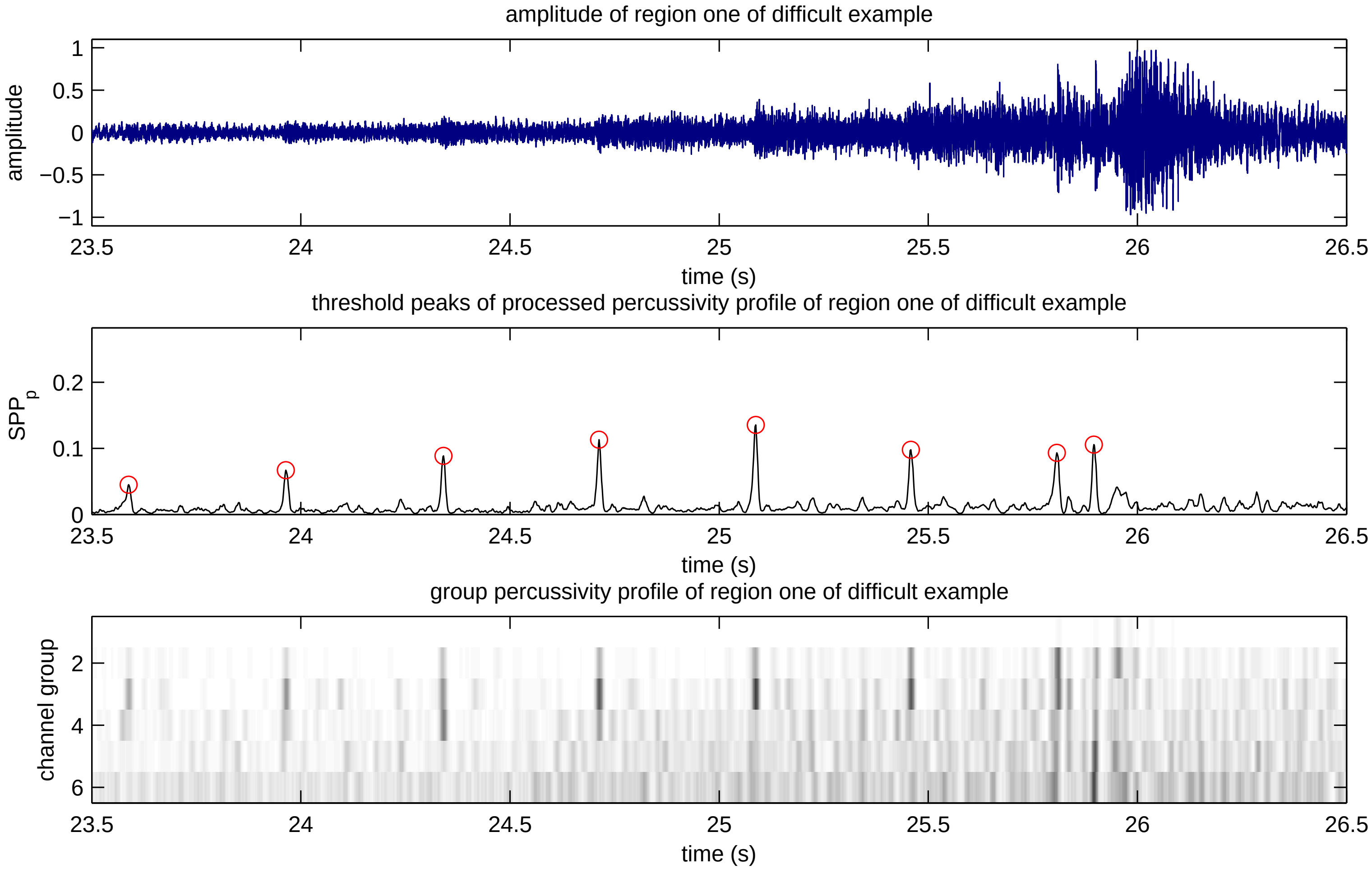


Figure 4.17: Amplitude (top plot), SPP_p (middle plot), and GPP (lower plot) of Region 1 of a section (from 4:22.93 to 5:11.30) of "Le Vertige Inconnu" by Gilles Gobeil [87]. The circled peaks in the SPP_p plot represent percussive instants according to a threshold value of 0.044. SPP_{rms} was taken over a 0.5 s window.

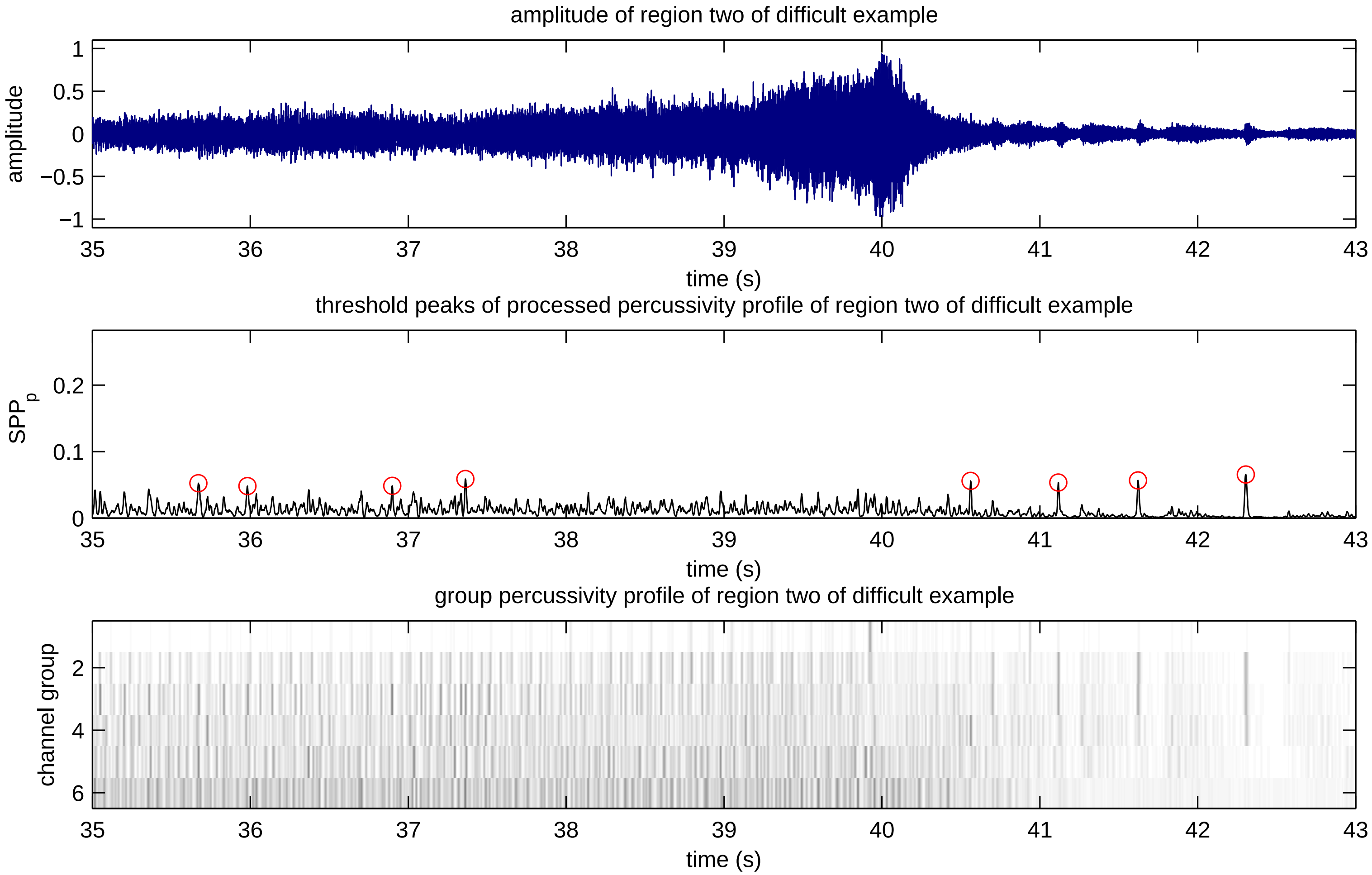


Figure 4.18: Amplitude (top plot), SPP_p (middle plot), and GPP (lower plot) of Region 2 of a section (from 4:22.93 to 5:11.30) of "Le Vertige Inconnu" by Gilles Gobeil [87]. The circled peaks in the SPP_p plot represent percussive instants according to a threshold value of 0.044. SPP_{rms} was taken over a 0.5 s window.

of the other sounds stop, and it is this point in the music that is of interest regarding being labelled as percussive by humans.

This point in the music does have several characteristics of percussive sound events: it has a sharp decrease in musical energy (a fall) immediately following it, it has a sharp rise in audio energy (a rise) at the closing of the gate, and it does appear to be a combination of two sounds, each of which fits one of the definitions of an SDPE (see Section 2.1). The timing, spectra, and mixture of the different sounds may interfere with one another so as to prevent the PPA and possibly humans from identifying this point as a percussive event. Certainly this point in the music is not a simple SDPE, but it does represent a sudden change in musical context.

Subregion D occurs from around 40.5 until 43.0 seconds. In this subregion, the dénouement of the intensity plays out with the general dampening of the sounds. There are four sound events that appear to be releases of steam. Each of these is identified by the PPA as a percussive event even though its amplitude is not significantly greater than the surrounding noise, especially the first release.

4.6 Concluding Remarks

This chapter described the details of a percussivity-profile algorithm (PPA) that was created to identify instants in pieces of music that humans would similarly identify as percussive. The implementation details are described fully. Tuning over the algorithm parameters was performed to maximize the PPA's performance at matching the most common choices made by participants in the percussion-judgment collection. With this tuning, the PPA was used to analyze one constructed example and two electroacoustic music examples. The results show that the PPA appears to perform well at the percussion-detection task, although the ambiguity of certain sound events and sound mixtures cause the PPA to not identify some sound events as percussive that might be heard as such by humans.

The stimulus sounds used in the percussion-judgment collection certainly had a significant effect on the sounds that can be identified as percussive by the PPA. An example of this is the truncated marble strikes in Region 1 of the easy example (see Figure 4.14). These sounds have a total length less than the shortest rise time used for the stimulus sounds in the percussion-judgment collection. Because the PPA is tuned from the results of the percussion-judgment collection, the tuning goal is incongruous with identifying those truncated marble strikes as percussive. Suggestions for how to choose new stimulus sounds are given in Section 6.2.

In listening to real electroacoustic music, like Region 2 of the difficult example (see Figure 4.18), a person can hear massive shifts in musical context. These context shifts seem to have a musical effect similar to a percussive event. The electroacoustic composer is probably conscious of this effect and is composing with this phenomenon in mind. Discussion as to whether these context shifts should be considered percussive requires further exploration of the word "percussive" and other sound events besides single, damped, percussive events (SDPEs) which would be considered percussive by humans. It is unsurprising that a PPA based on a collection of human percussion judgments that is subsequently based on a definition of an SDPE has some trouble performing clearly with these ambiguous sounds.

Chapter 5 discusses a method to use the results of the PPA in order to illuminate the self-similarity in the percussive sounds of electroacoustic music. Chapter 6 discusses, amongst other things, several ways in which the performance of the PPA might be measured and improved.

Chapter 5

Percussive Self-Similarity

This chapter describes the similarity matrix used by Foote [57] to visualize self-similarity in music. This chapter also describes how the similarity matrix can be combined with the results of the percussivity-profile algorithm (PPA) from Chapter 4 in order to find self-similarity in the percussive sounds of a piece of music. Practical limitations of such an analysis are considered, and the performance of this type of analysis for one constructed example and two examples of electroacoustic music are shown.

Self-similarity in this context refers to a property of a piece of music. A piece of music is said to have high self-similarity if there are many sections of the piece which are similar to many other sections. Depending on the application, the sections under consideration may be longer than half the length of the piece of music or as short as a percussive event.

The percussivity similarity matrix (PSM) is one mechanism by which the results of the PPA can be used in order to provide more information about a piece of music than might otherwise be available. The PSM shows where a piece of music exhibits self-similarity in the percussive sounds and helps to identify structural characteristics of a piece of music.

5.1 Algorithm

This section describes Foote's [57] similarity matrix and how the PPA can be integrated with it to provide a PSM. A diagonal summation of the PSM is also shown as a helpful indicator of percussive self-similarity.

5.1.1 Similarity Matrix

Foote [57] suggests that a two-dimensional similarity matrix can be used to visualize pieces of music in order to emphasize sections of the music which are similar to other sections. An example similarity matrix [89] for “The Magical Mystery Tour” by The Beatles [90] is shown in Figure 5.1. In the similarity matrix, time runs from left to right and from top to bottom. The time difference, or time lag, between two points being compared is indicated by the horizontal or vertical distance (always the same) from the main diagonal. The brightness of a point (i, j) in the matrix is proportional to the similarity at instants i and j of the piece of music. In reality, the “instants” are created by dividing the piece of music into short time windows.

Foote and Cooper [61] suggest that any appropriate similarity metric, $s(i, j)$, may be chosen. In their analyses, they most commonly use the cosine distance between mel-frequency cepstral coefficients (MFCCs) to make the comparison for time windows of 100 milliseconds [61]. MFCCs are coefficients which represent a short-term audio power spectrum. Cosine distance of MFCCs and a 100 millisecond time window are used in Figure 5.1. Figure 5.2 shows how the similarity matrix is constructed.

Because the main diagonal of the similarity matrix indicates a time offset of 0 seconds, the similarity matrix has a white stripe running along the main diagonal of the matrix. This white stripe indicates that each time window is similar to itself (autocorrelation is maximum at a time lag of 0 seconds). For a piece of music that maintains the same tempo throughout its length, other white stripes running parallel to the main diagonal indicate similarity found at a time offset. Note that the similarity matrix is symmetric across the main diagonal when $s(i, j) = s(j, i)$.

5.1.2 Percussive Similarity Matrix

By choosing a similarity metric that is based on percussivity, the similarity matrix can be turned into a PSM. Issues arise when trying to create a PSM, however. Percussion is, by its nature, short and often comprises little of the total time of a piece of music, so most of a piece of music will be strongly similar to itself (no percussion is similar to no percussion). Obviously, exceptions exist depending on musical style.

It is only during short time periods of the music that the presence of a percussive event will be compared to the presence of another percussive event. If these percussive events are rated as similar by the percussive similarity metric chosen, then the PSM indicates similarity by plotting a light pixel at the appropriate location. If percussive

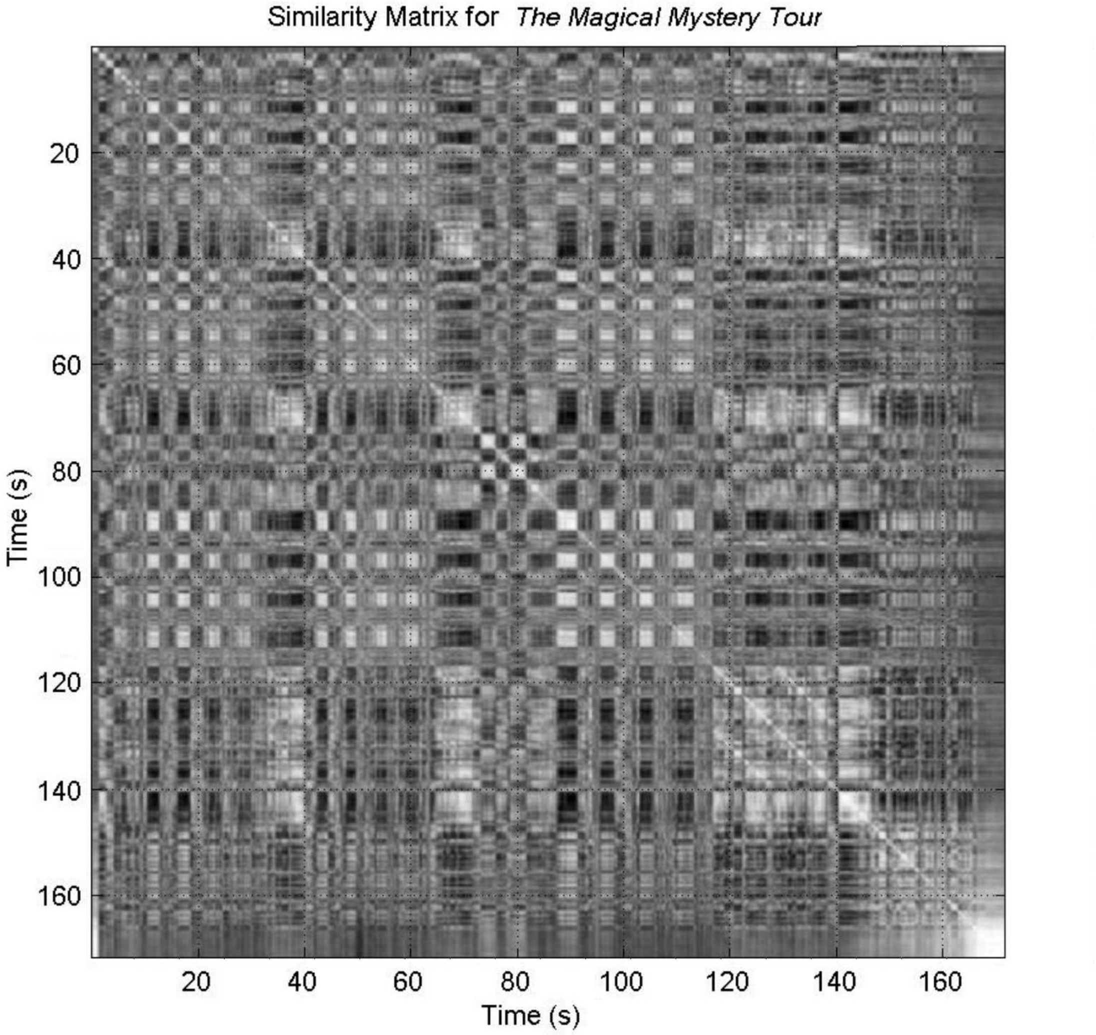


Figure 5.1: Example of a similarity matrix from Cooper and Foote [89] (modified slightly). Time runs horizontally to the right and vertically down. Light regions indicate similarity and dark regions indicate less or no similarity. The time difference, or time lag, between two points being compared is indicated by the horizontal of vertical distance (always the same) from the main diagonal.

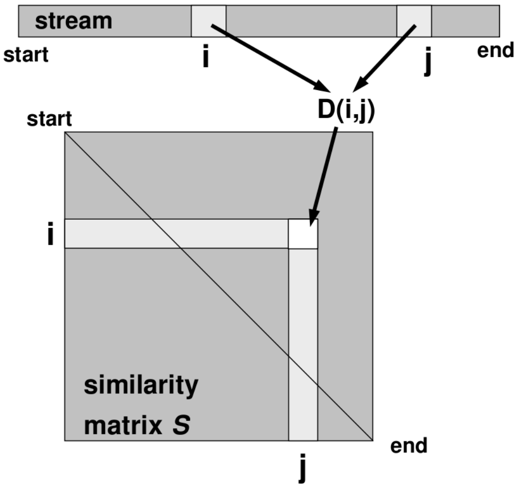


Figure 5.2: Creation of the similarity matrix from Cooper and Foote [62]. $D(i, j)$ here is the cosine distance between MFCCs at points i and j in the audio. $D(i, j)$ is one choice of similarity metric, $s(i, j)$. "Stream" here refers to the audio stream of the piece of music. The value of similarity for any point in the similarity matrix is value of $D(i, j)$.

comparisons line up at the same time offsets, then a white line appears parallel to the main PSM diagonal. This white line passes through the intersections of the horizontal and vertical dark lines created by percussive events, Figure 5.3 shows a PSM from a constructed example for immediate reference. This example is more fully explained in Section 5.2.1.

In order to quantify the self-similarity in a piece of music, the values of mean similarity are taken along the PSM diagonals. If percussive similarity is regularly present at a specific time offset, that similarity appears in the comparison of the PSM diagonal means. This work follows Foote and Cooper's [61] work on "beat spectrum," but that nomenclature is inappropriate when analyzing electroacoustic music because the lack of musical beat is common in the genre.

5.1.3 Percussive Similarity Metric

The percussive similarity metric used is given by

$$s(i, j) = 1 - \frac{d(i, j)}{d_{max}} \tag{5.1}$$

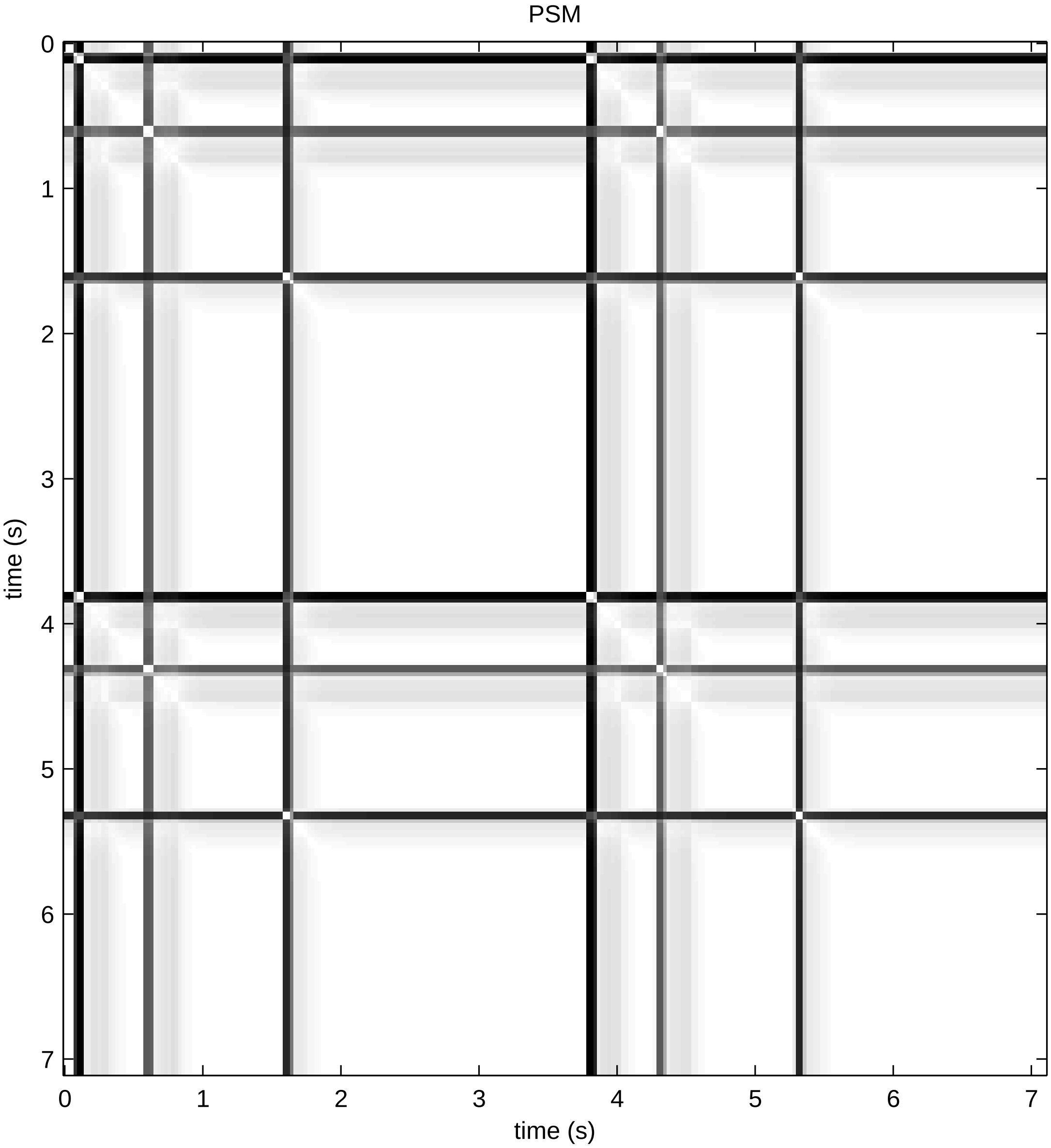


Figure 5.3: PSM for a constructed sound example using the GPP in the similarity metric. At the intersections of percussive events (dark lines), light pixels indicates high similarity and dark pixels indicate low similarity.

where

$$d(i,j) = ||\mathbf{x}_i - \mathbf{x}_j|| = \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{j,k})^2} \tag{5.2}$$

is the Euclidean distance [91] between the vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ and d_{max} is the maximum Euclidean distance between the group percussivity profile (GPP) vectors of any two time windows in the piece of music (see Section 4.2 for an explanation of the GPP). In Equation 5.2, $\mathbf{x}_i$ represents the vector of GPP values at time window i, n is the number of values at a given time window in the GPP (and is equal to the PPA parameter `nChannelGroups`), and $x_{i,k}$ is a value of the GPP at time window i and channel group k. The quantity is normalized so that absolute percussive similarity gives a percussive similarity value of 1.0, and the least-similar point on the matrix has a value of 0.0. This normalization is appropriate because the percussivity values from the PPA are arbitrary (see Section 4.4.1), and because the amount of percussion in the piece of music will have a significant effect on the PSM diagonal means.

Basing the percussive similarity metric on the GPP has the effect of discerning different types of percussive sounds from one another if the differences appear in the frequency discernment of the GPP. In order to search for percussive self-similarity based only on the fact that any percussive event occurred, a percussive similarity metric based on only the single-value percussivity profile (SPP) could be used (see Section 4.2 for an explanation of the SPP). An example of a PSM using this similarity metric is shown in Section 5.2.1.

5.1.4 Practical Considerations

The memory requirements for the PSM and associated manipulation are given by Equation 5.3,

$$M_{max} = 17.6 \left(\frac{l_m}{l_w} \right)^2 \tag{5.3}$$

where l_m is the length in seconds of the piece of music to be analyzed, and l_w is the time window length in seconds. From Equation 5.3, if 7 GB of real memory are available, then the value of 20700 is the maximum for $\frac{l_m}{l_w}$ without moving into virtual memory and significantly slowing down the calculations. Given the values of 0.0126 seconds for the `windowTimeLen` PPA parameter (see Section 4.3.2) and 10 start times distributed throughout the PPA time window (see Section 4.2.6), GPP values exist at time windows of 0.00126 seconds. Using this value for l_w gives an l_m of only 26.1 seconds.

In order to analyze pieces of music longer than 26.1 seconds, some form of decimation must occur. Using 15 minutes as the longest l_m, a value of 0.05 seconds for l_w is calculated. In order to avoid the problems that arise from such a long l_w (compared to the rise time of percussive events), decimation is performed according to the maximum percussivity value within the channel group over l_w with an overlap of $\frac{l_w}{2}$. The overlap is necessary to avoid splitting a single percussive event across two windows, but unfortunately reduces the longest l_m by a factor of four. An l_w of 0.05 seconds is used for all examples in this chapter under 3.75 minutes. An l_w of 0.1 seconds is used for all examples over 3.75 minutes.

5.2 Results

The following sections describe how the PSM indicates percussive self-similarity. The first example is a somewhat artificial, constructed example and is intended to display percussive self-similarity with a minimum of musical interference. Two examples of electroacoustic music follow the constructed example: a section of “La Ou Vont les Nuages...” by Gilles Gobeil [87] (labelled as the simple example) and a section of “Associations Libres” also by Gilles Gobeil [87] (labelled as the complex example). The simple example was chosen to demonstrate the ability of the PSM to display percussive self-similarity for a musical selection with almost completely regular percussive time intervals. The complex example was chosen to demonstrate the ability of the PSM to display percussive self-similarity for a more musically complex selection. All of these examples are monaural audio (reduced from two channels by averaging if necessary) and are recorded at a 44.1 kHz sampling rate with 16 bit resolution.

5.2.1 Constructed Example

In order to better explain the PSM, an example exhibiting significant percussive self-similarity with no other musical interference was constructed using three different percussive sounds from stimulus set B of the collection of human percussion judgments (see Section 3.1.1). Figure 5.4 shows the amplitude, SPP_p, and GPP of the constructed example. All of the percussive sounds exhibit the fastest rise time of the stimulus set (0.01 seconds), and the three individual sounds were the stimulus sounds processed using filter sets A, B, and C.

The percussive sounds were placed in the order B, A, C, B, A, C at the following times in the sound file: 0.1, 0.6, 1.6, 3.8, 4.3, and 5.3 seconds. There is no other noise

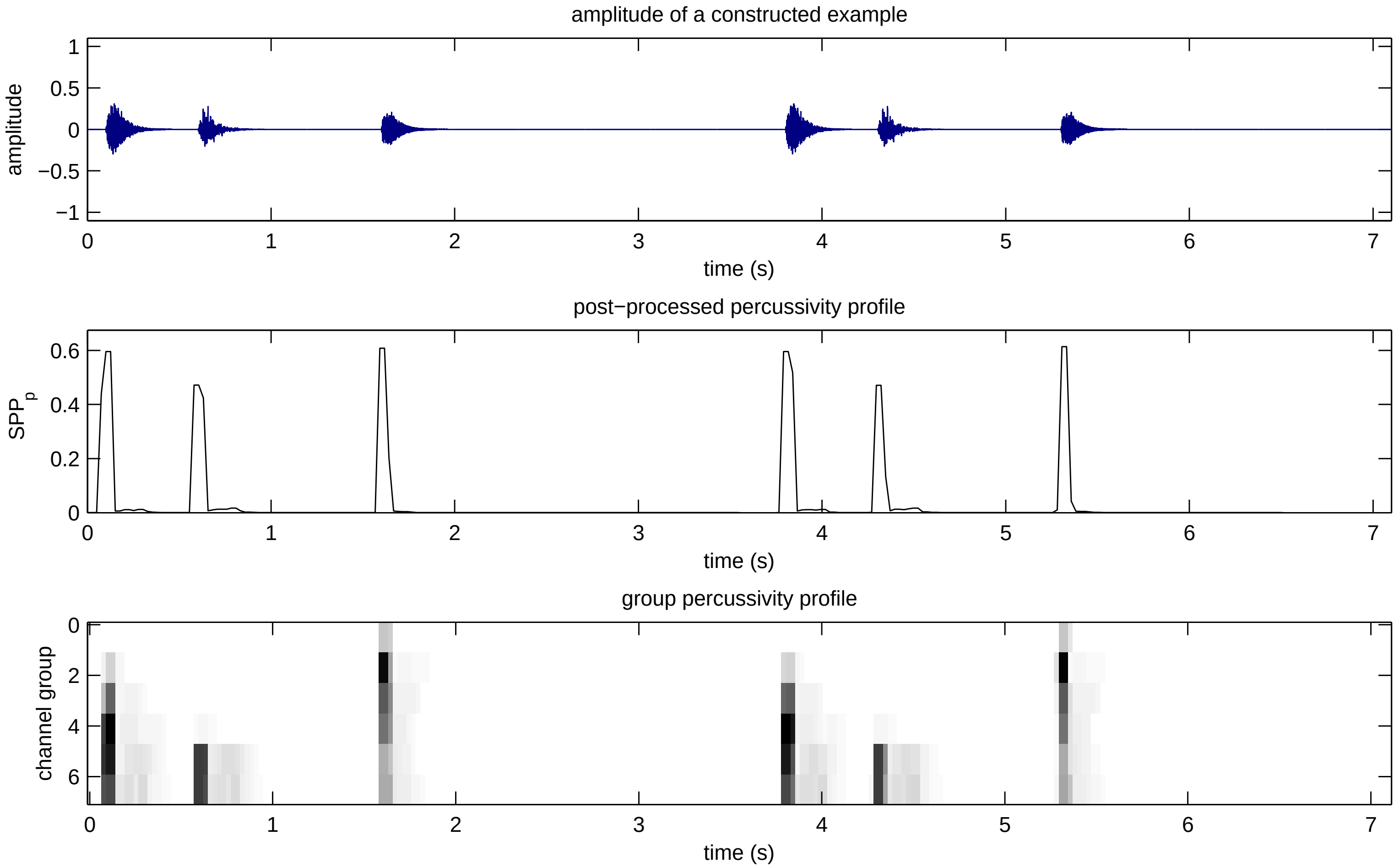


Figure 5.4: Amplitude (top plot), SPP_p (middle plot), and GPP (lower plot) of the constructed sound example. These values are decimated using $l_w = 0.05$ s.

or sound added to the example and it ends at 7.1 seconds. This particular percussion pattern generates strong self-similarity at a time offset of 3.7 seconds as well as weaker self-similarity at other time differences between the percussive sounds (at a time offset of 0.5 seconds, for example). The SPP_p plot shows that these percussive events are recognized as percussive, and the GPP plot shows that each of the filter sets creates a unique channel-group signature. The SPP_p and GPP plots also show the effects of decimation.

Figure 5.3 (in Section 5.1.2) shows the PSM for the constructed example. The expected white line along the main diagonal (upper left to lower right) is present showing that the entire piece of music is similar to itself. Parallel white lines can be seen in the bottom left quadrant (through points (0.1, 3.8), (0.6, 4.3), and (1.6, 5.3)), and in the upper right quadrant (through points (3.8, 0.1), (4.3, 0.6), and (5.3, 1.6)). These parallel lines indicate similarity between two sections of the piece of music at a time offset. The time offset is the vertical or horizontal time offset from the main diagonal for any specific time window, and in this case has a value of 3.7 seconds. It is important to note that not every percussive event is considered similar to every other percussive event. This fact is evidenced by the display of low similarity (dark pixels) at many of the intersections of percussive events, for example at (0.1, 0.6).

Figure 5.5 shows the PSM diagonal means from the upper triangle of the PSM for the constructed example. The upper triangle of a matrix is all matrix values above and including the main diagonal in the matrix layout. Only the upper triangle is used because of the symmetry of the PSM across the main diagonal. The value of 1.0 at a time offset of 0 seconds is the mean value of the main diagonal. After the main diagonal, the highest peak in the PSM diagonal means occurs at a time offset of 3.7 seconds, indicating that the highest amount of similarity across this example occurs at that time offset. Smaller peaks represent some of the other time differences between the percussive sounds (at a time offset of 0.5 seconds, for example).

It is worth noting that every PSM diagonal is of a different length, and PSM diagonals near the upper right corner have few values over which the mean is taken. Because the average of a small number of observations of a random variable is not guaranteed to be near its expected value according to the law of large numbers [92], the PSM diagonal means are not presented where the number of diagonal values is 50 or fewer.

In order to show a PSM when every percussive event is calculated to be similar to every other percussive event, Figure 5.6 shows the PSM for the constructed example using the normalized difference of the SPP as the similarity metric. In this case the GPP signature of each percussive event has no effect, and simply the SPP is important. This

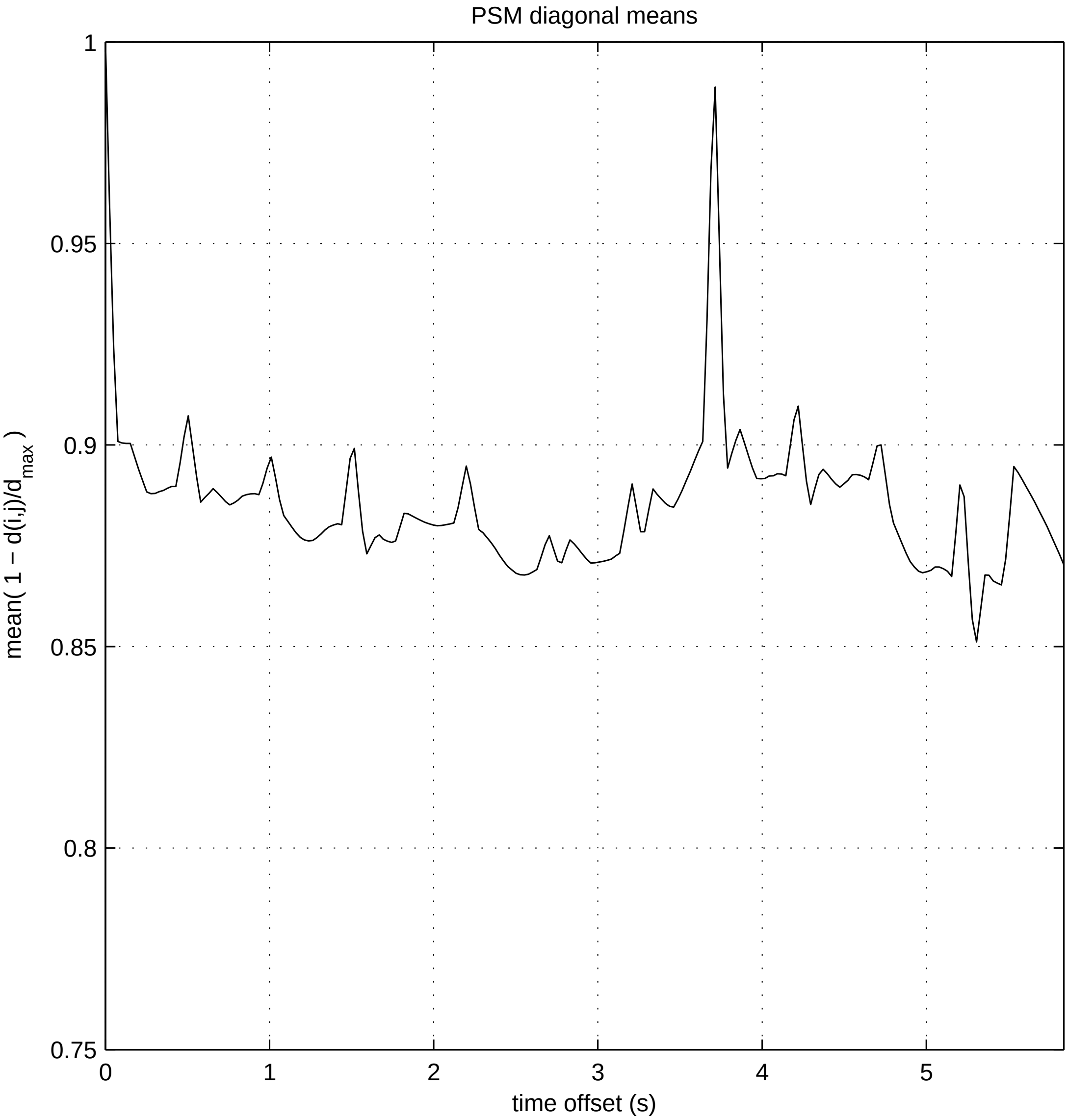


Figure 5.5: PSM diagonal means of the constructed sound example using the GPP in the similarity metric.

is evidenced in Figure 5.6 by the high similarity (light pixels) at the intersection of every percussive event with every other percussive event. Figure 5.7 shows how the change in similarity metric affects the PSM diagonal means. The smaller peaks from Figure 5.5 become more prominent.

5.2.2 Simple Example

The first electroacoustic music example used for the application of the PSM is a section (from 2:40.98 to 3:25.94) of "La Ou Vont les Nuages..." by Gilles Gobeil [87], and was chosen to demonstrate the ability of the PSM to display percussive self-similarity for a musical selection with almost completely regular percussive time intervals. The amplitude, SPP_p, and GPP of this example are shown in Figure 5.8. There are many sounds presented in the complex sonic environment of this example: shuffling and announcements in a large space reminiscent of a train station, several tones reminiscent of a distant fog horn or of feedback, the tick of a clock, and most important to this analysis, what sounds like the regular percussive strike of a metal box with the strike of a bell on every other box strike.

This example was chosen as the simple example because, although it has a complex sonic milieu, the percussive sound events are prominent and regular so there is significant percussive self-similarity. The percussive strikes occur at regular intervals of approximately 4.6 seconds. Of the 10 box strikes, the bell sound is combined with the box sound on the first, third, fifth, seventh, and ninth strikes.

The PSM for this example is shown in Figure 5.9. The regular percussive sound events create a regular grid in the PSM. The white line along the main diagonal is once again visible. Wherever two percussive sound events interact on the PSM, percussive similarity is also shown by the white points at the intersection. Because these interactions are at such regular intervals, they line up in diagonals parallel to the main diagonal.

Figure 5.10 shows the diagonal means of the upper triangle of the PSM for this example. At a time offset of 0 seconds, the diagonal mean is 1.0 as expected. The peaks, seen at every multiple of 4.6 seconds, indicate percussive self-similarity at every multiple of 4.6 seconds. Because the bell sound combines with the box sound at every other box sound, a more significant increase of the second peak was expected, but did not materialize. Because the PPA only recognizes the attack of a percussive sound, it may be that the bell sound had little effect on the percussivity signature of the percussive sound in the GPP, and therefore had little effect on the percussive similarity metric. The slight increase in the fourth peak may be due to subtle differences in the percussive sound timings that lead to groups of four aligning slightly better than other groupings.

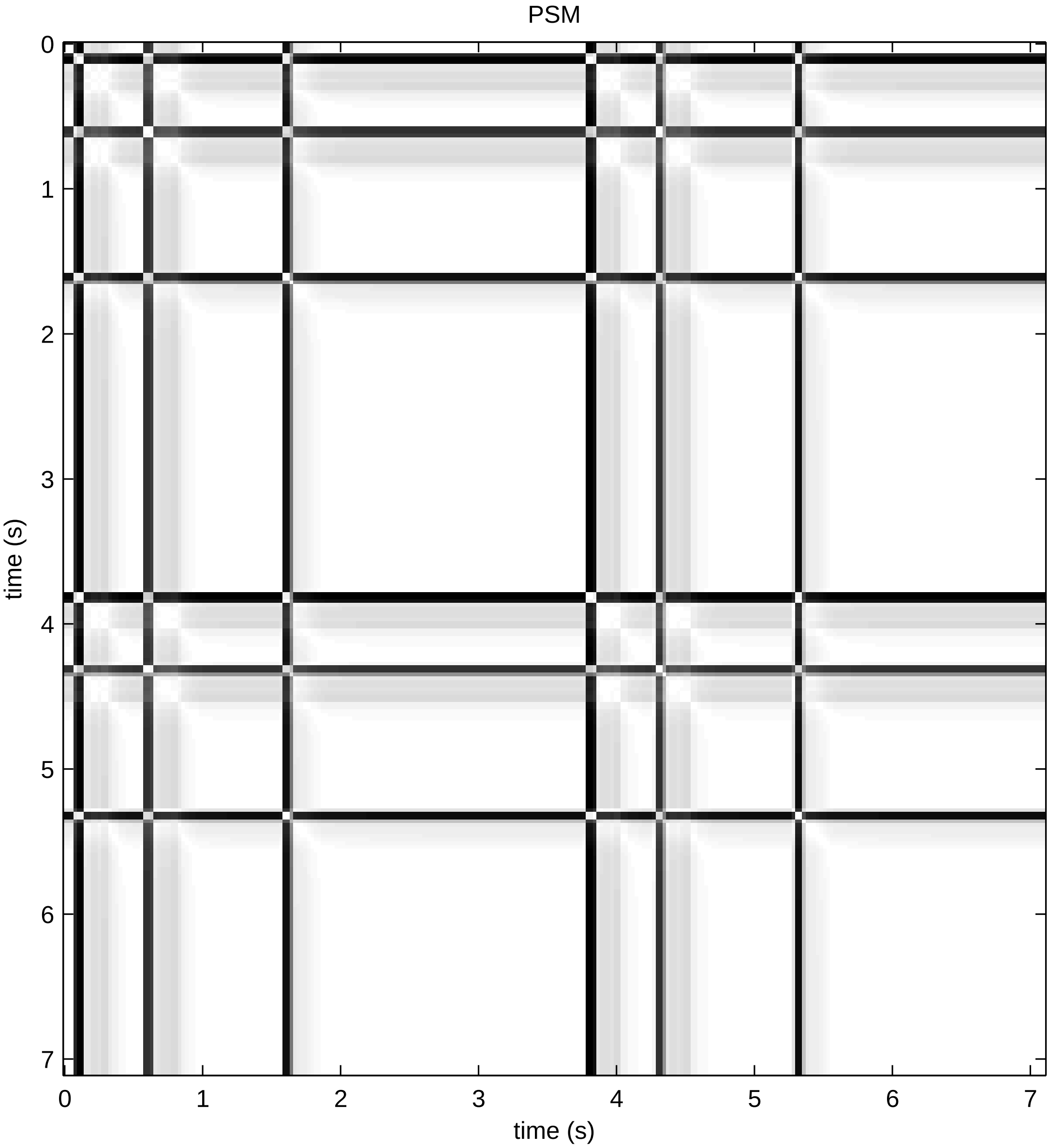


Figure 5.6: PSM for a constructed sound example using the SPP in the similarity metric. Every intersection of percussive events (dark lines) indicates high similarity (light pixels).

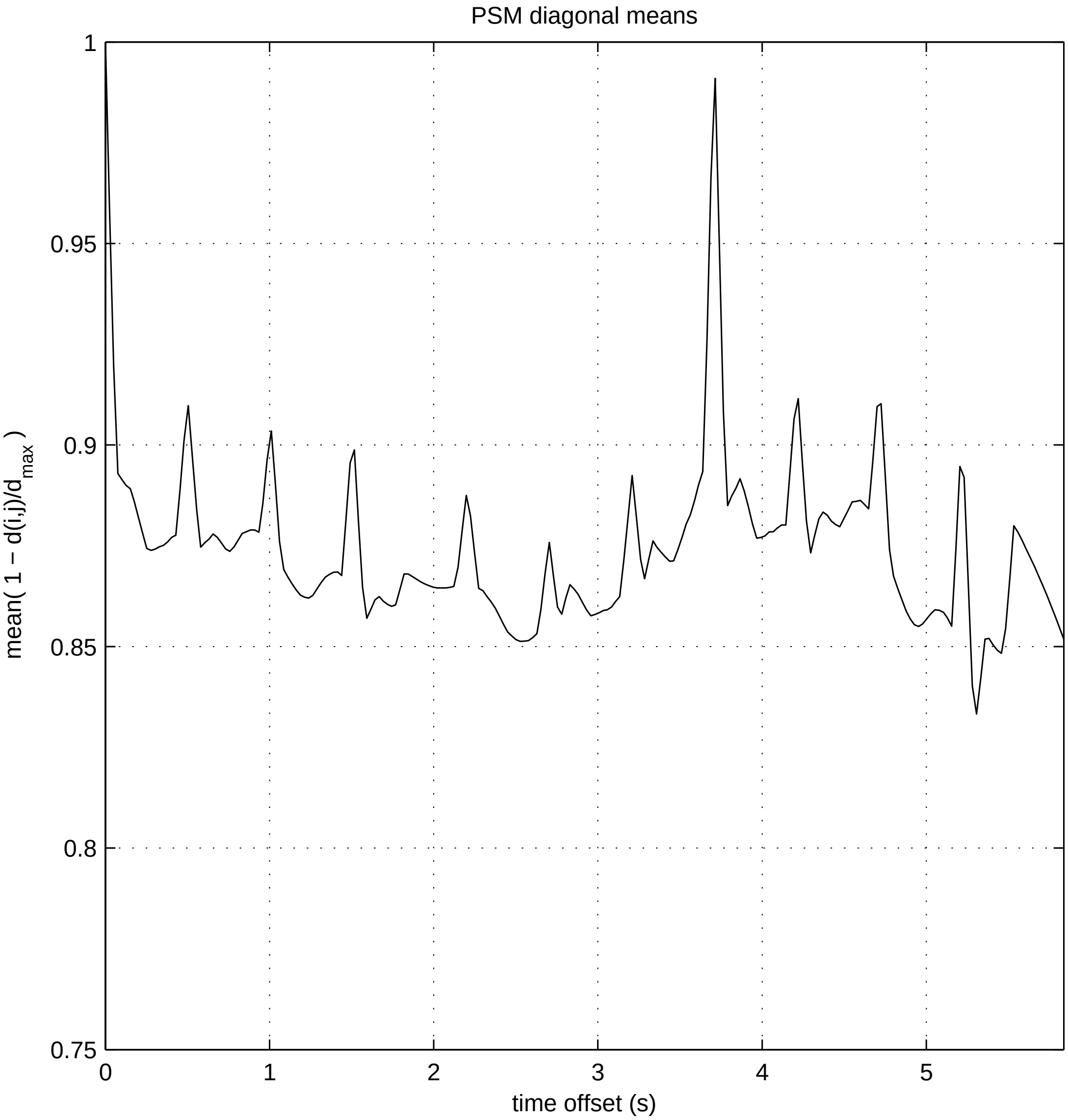


Figure 5.7: PSM diagonal means of the constructed sound example using the SPP in the similarity metric.

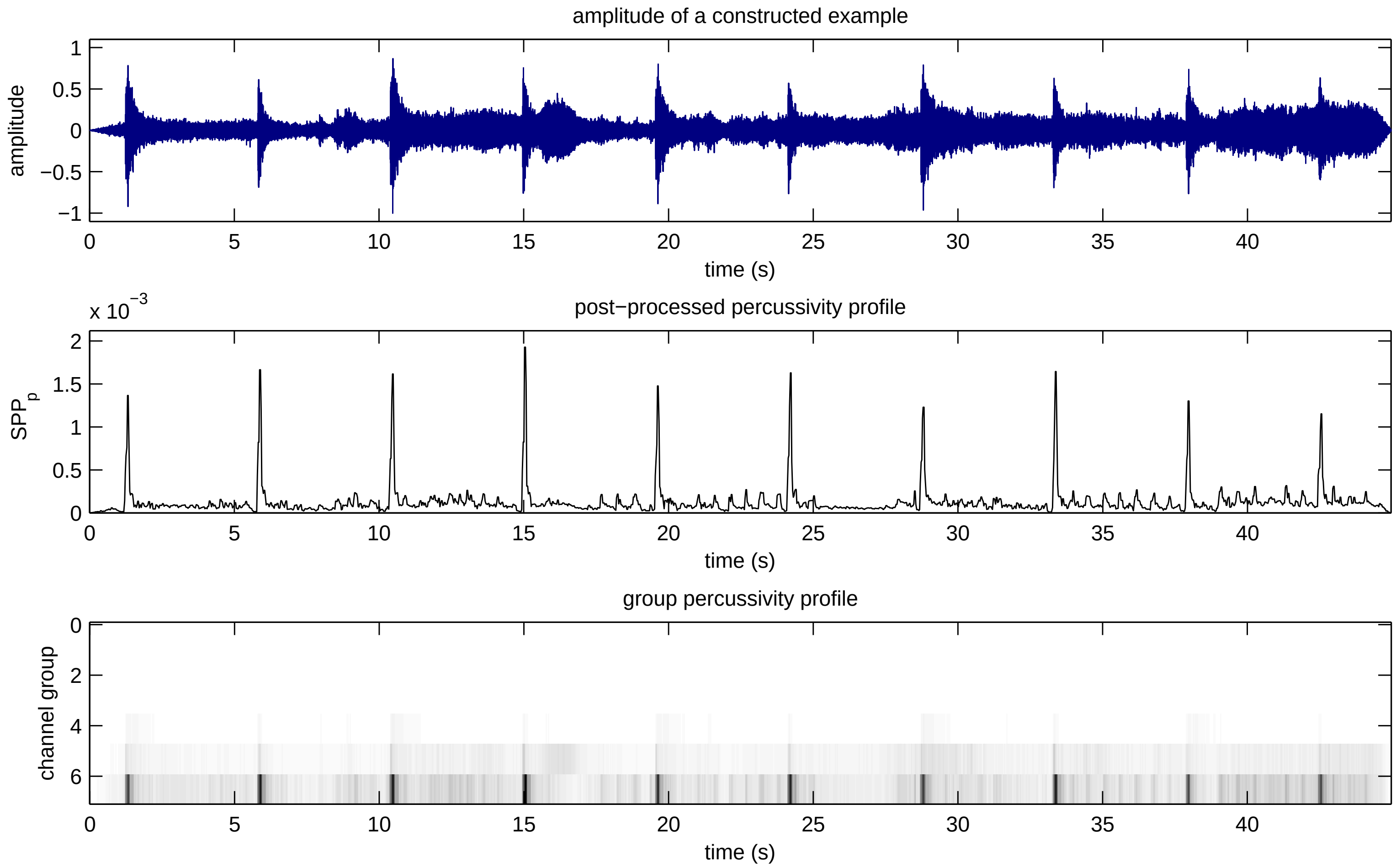


Figure 5.8: Amplitude (top plot), *SPP*$_p$ (middle plot), and GPP (lower plot) of a section (from 2:40.98 to 3:25.94) of “La Ou Vont les Nuages...” by Gilles Gobeil [87]. These values are decimated using $l_w = 0.05$ s.

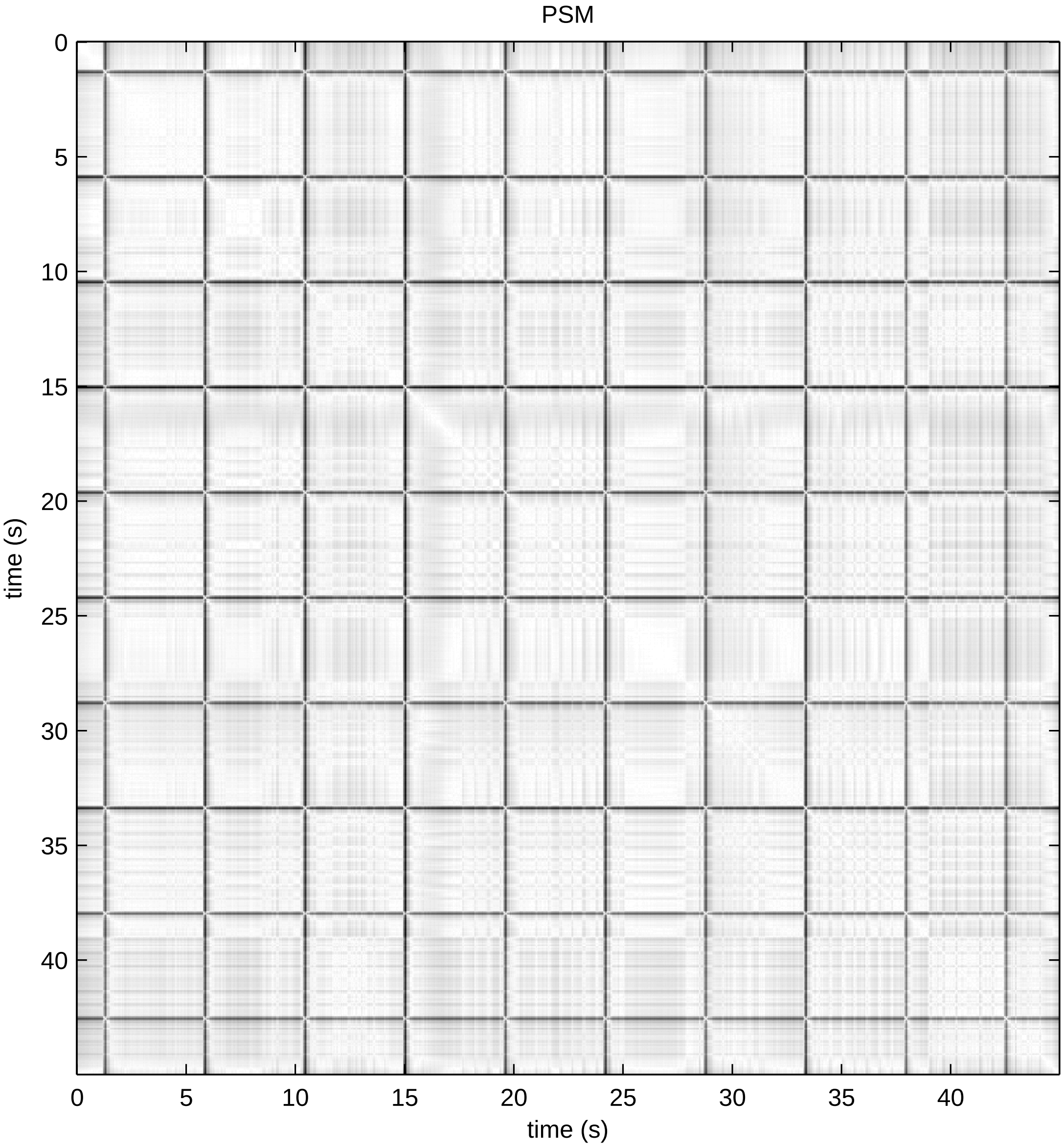


Figure 5.9: PSM of a section (from 2:40.98 to 3:25.94) of "La Ou Vont les Nuages..." by Gilles Gobeil [87].

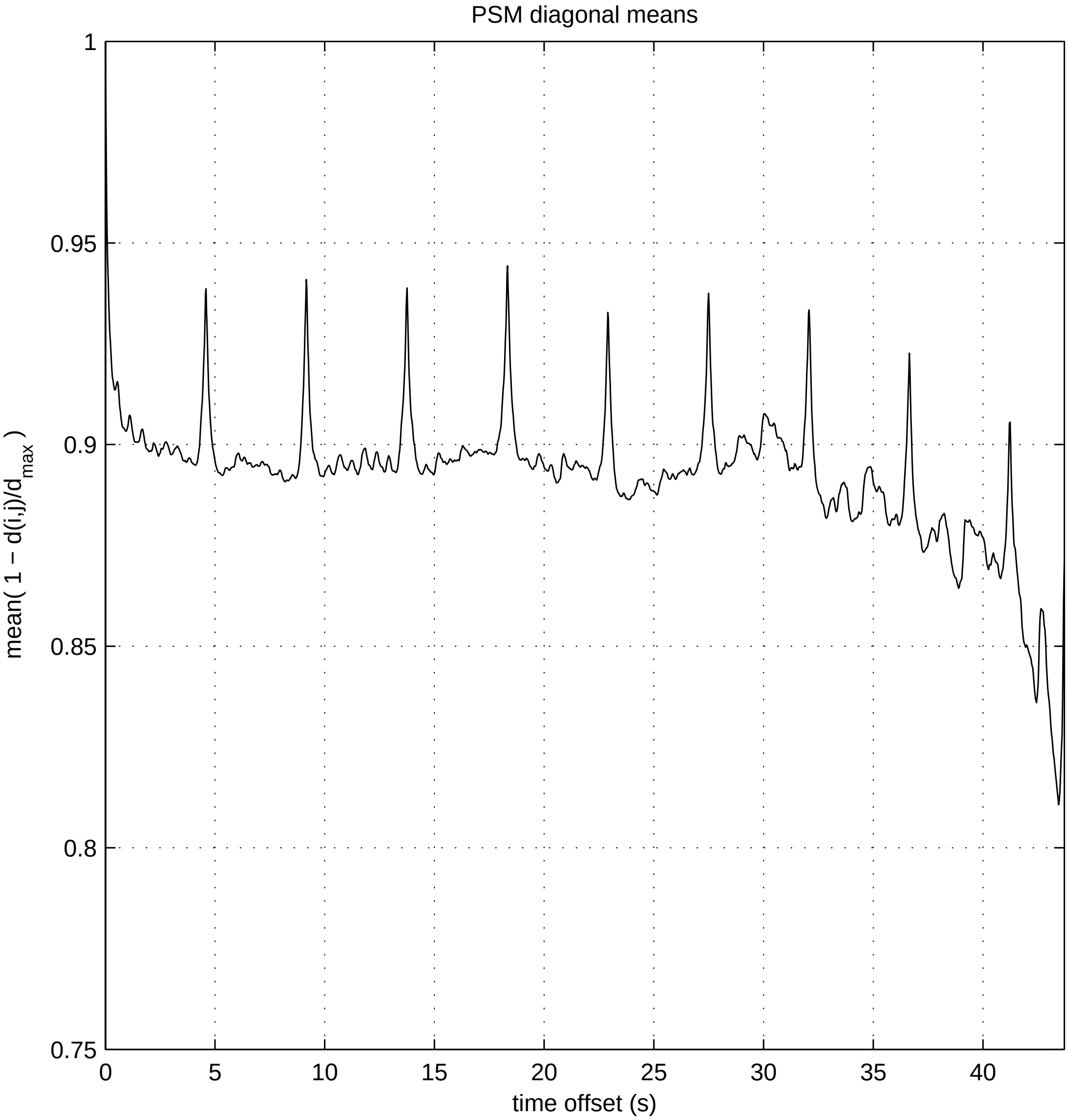


Figure 5.10: PSM diagonal means of a section (from 2:40.98 to 3:25.94) of “La Ou Vont les Nuages...” by Gilles Gobeil [87].

5.2.3 Complex Example

The final example used to show the application of the PSM is a section (from 1:48.08 to 2:34.24) of “Associations Libres” by Gilles Gobeil [87], and was chosen to demonstrate the ability of the PSM to display percussive self-similarity for a more musically complex selection. The amplitude, SPP_p, and GPP of this example are shown in Figure 5.11. This example is musically complex and aggressive. The sounds in this example include sounds reminiscent of automatic gunfire, the cocking of a gun, a gong, a distorted and amplified spring being struck and processed vocally, a drum roll played with a bell, and the tick of a clock.

The arrangement of the piece can be seen in the amplitude and SPP_p plots of Figure 5.11. There are a total of eight musical swells in the piece. The first four swells are separated from the final four by a musical pause. Each swell generally consists of a drum roll played on a bell, a gun-cocking sound, and a gong strike, although the fourth swell removes the gong strike, and the fifth swell uses automatic gun fire instead of the drum roll played on a bell as the rise of the swell. The distorted and amplified spring sounds occur throughout both groups of swells, but during the second group, the spring sounds appear even more distorted, processed, and aggressive. The musical pause starts with the missing gong sound of the fourth swell, and includes quiet processed vocal sounds and the tick of the clock. The pause ends when a gun-cocking sound leads into the sound of the automatic gunfire of the fifth swell.

The percussive self-similarity in the piece can be somewhat extrapolated from the SPP_p plot of Figure 5.11. The gong strikes appear in two groups of three (at times 1.8, 7.3, and 13.0 seconds; and 26.7, 32.4, and 37.9 seconds), and all of the swells are visible in the SPP_p plot. Self-similarity is expected to arise at the time offset between the swells within a group (at about 5.6 seconds) and also at the time offset between the groups (at about 25.0 seconds). Weaker self-similarity should appear at the time offsets between groups of less than four sounds; for example, there exists weaker self-similarity at a time offset of 30.6 seconds between the first three swells of the first group and the last three swells of the second group.

The PSM for this example is shown in Figure 5.12. The main diagonal is, of course, white again. In the upper left quadrant and lower right quadrant, lighter lines of similarity parallel to the main diagonal can be seen at time offsets that are multiples of the time between the swells (approximately 5.6 seconds). In the bottom left and upper right quadrants, parallel light lines can be seen that are at time offsets related to the

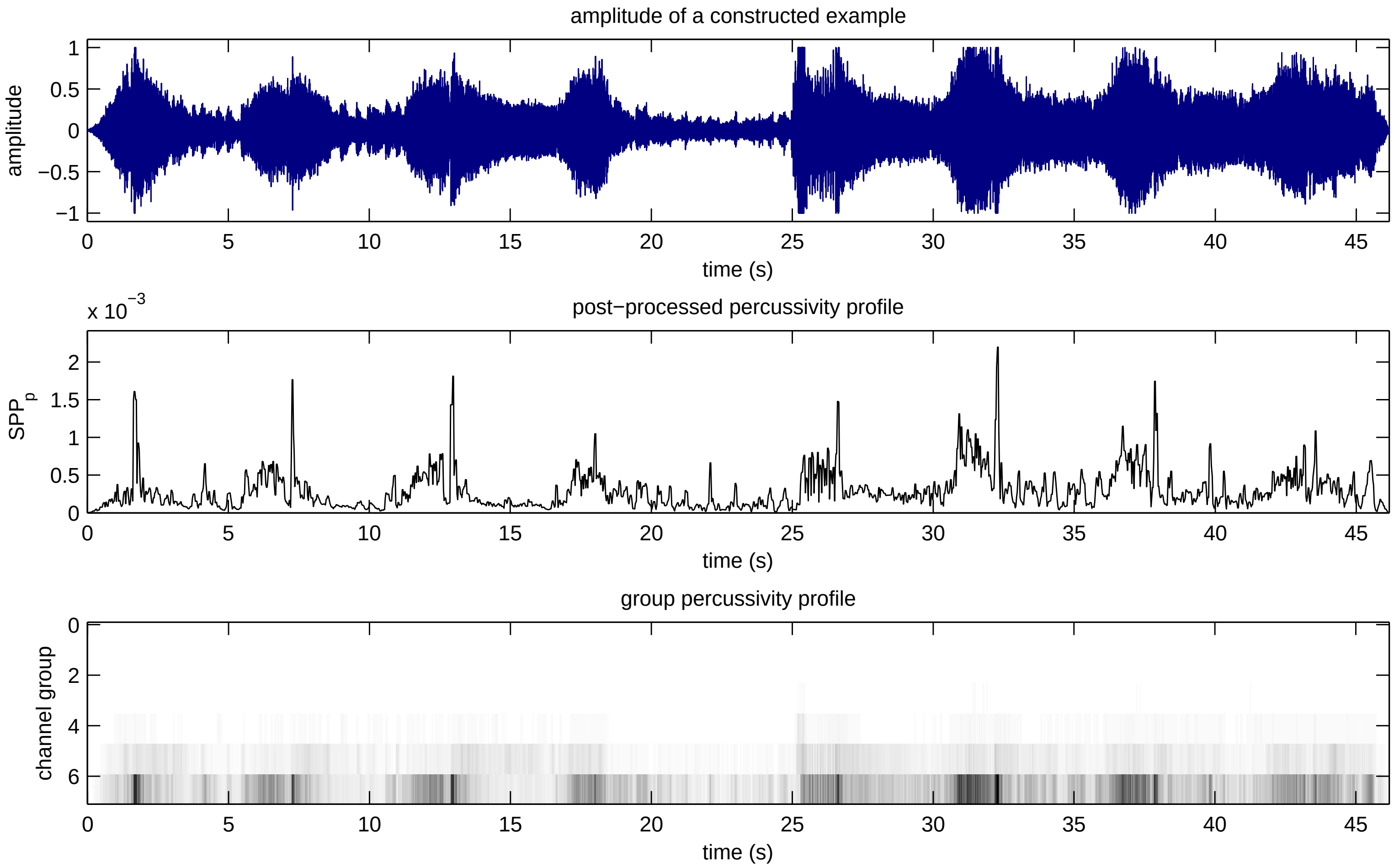


Figure 5.11: Amplitude (top plot), SPP_p (middle plot), and GPP (lower plot) of a section (from 1:48.08 to 2:34.24) of "Associations Libres" by Gilles Gobeil [87]. These values are decimated using $l_w = 0.05$ s.

interval between the two groups of swells (approximately 25.0 seconds) with multiples of the interval between swells within a group (approximately 5.6 seconds) added and subtracted.

Figure 5.13 shows the diagonal means of the upper triangle of the PSM for this example. The two highest peaks, aside from the peak at an offset of 0 seconds, are the most significant. The highest peak of these is at 5.6 seconds and represents the percussive self-similarity between swells within a group. Because four swells appear in each group, a second, weaker multiple of this peak is seen at 11.2 seconds. The significant width of the peaks is caused by the similarity of the percussivity within a swell. The second highest peak is at 25.0 seconds and represents the percussive self-similarity between the two groups of swells. Other, weaker peaks seen at 19.4 and 30.6 seconds are caused by the interactions of the groups as mentioned previously.

The PSM for the full piece of music is shown in Figure 5.14. It does not appear that any other sections of this piece of music exhibit the self-similarity seen in the previously chosen section. The PSM of that segment can be seen from approximately 108 to 154 seconds.

5.3 Concluding Remarks

The percussivity similarity matrix (PSM) and the diagonal means of the PSM do point out self-similarity in the percussion of pieces of music. These tools are able to indicate easily when a regular pulse or beat occurs in the music, but also are able to indicate repetitions of percussive events in music that simply have the same timing.

One aspect of the PSM not yet mentioned is the issue of tempo changes. Tempo changes here refer specifically to the repetition of a section of a piece of music faster or slower than the original section. This type of tempo change would create a white (or lighter) line in the PSM that would not be exactly parallel to the main diagonal but at an angle to it. A new technique for finding this type of line would need to be employed rather than a simple averaging of the diagonals.

Another aspect of the PSM not yet specifically mentioned is that the strength of the peaks of the PSM diagonal means is dependent on how the section was chosen from the piece of music. In the examples given, the sections of music exhibited strong self-similarity, and the sections were chosen in order to display that effectively. If longer sections of the pieces of music had been analyzed, the peaks of the PSM diagonal means might not have been as strong due to the lack of self-similarity across the entire section.

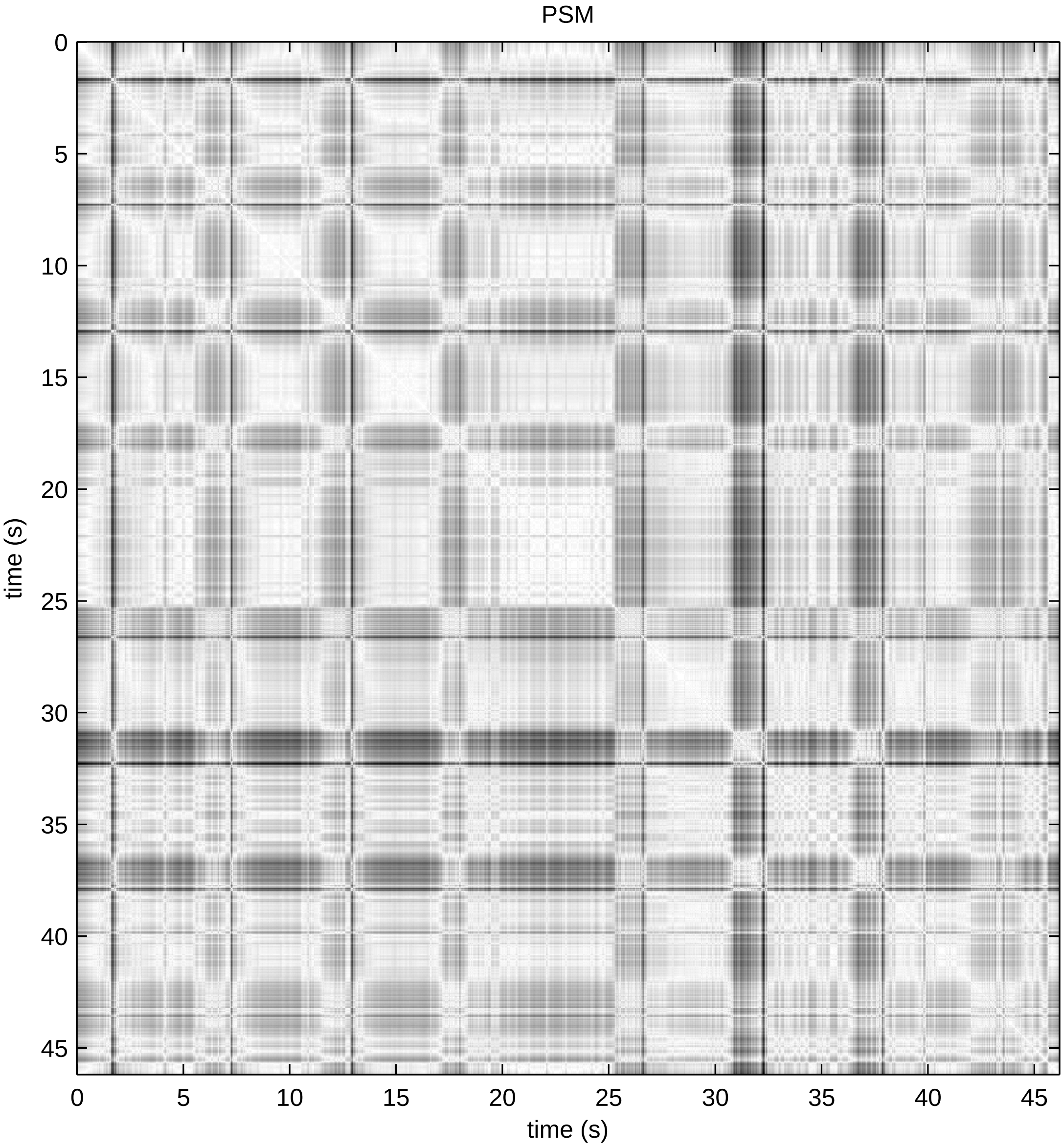


Figure 5.12: PSM of a section (from 1:48.08 to 2:34.24) of "Associations Libres" by Gilles Gobeil [87].

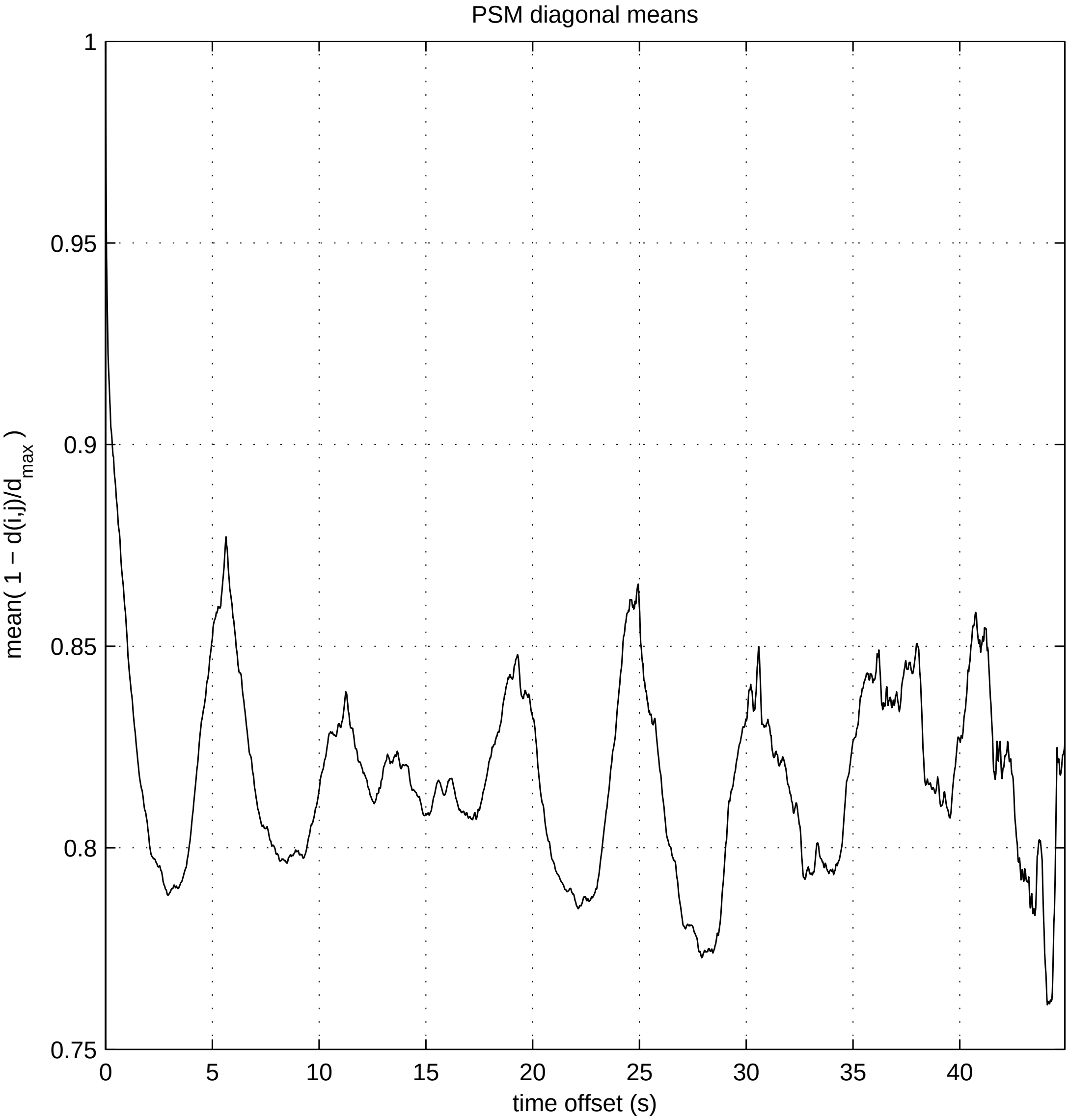


Figure 5.13: PSM diagonal means of a section (from 1:48.08 to 2:34.24) of “Associations Libres” by Gilles Gobeil [87].

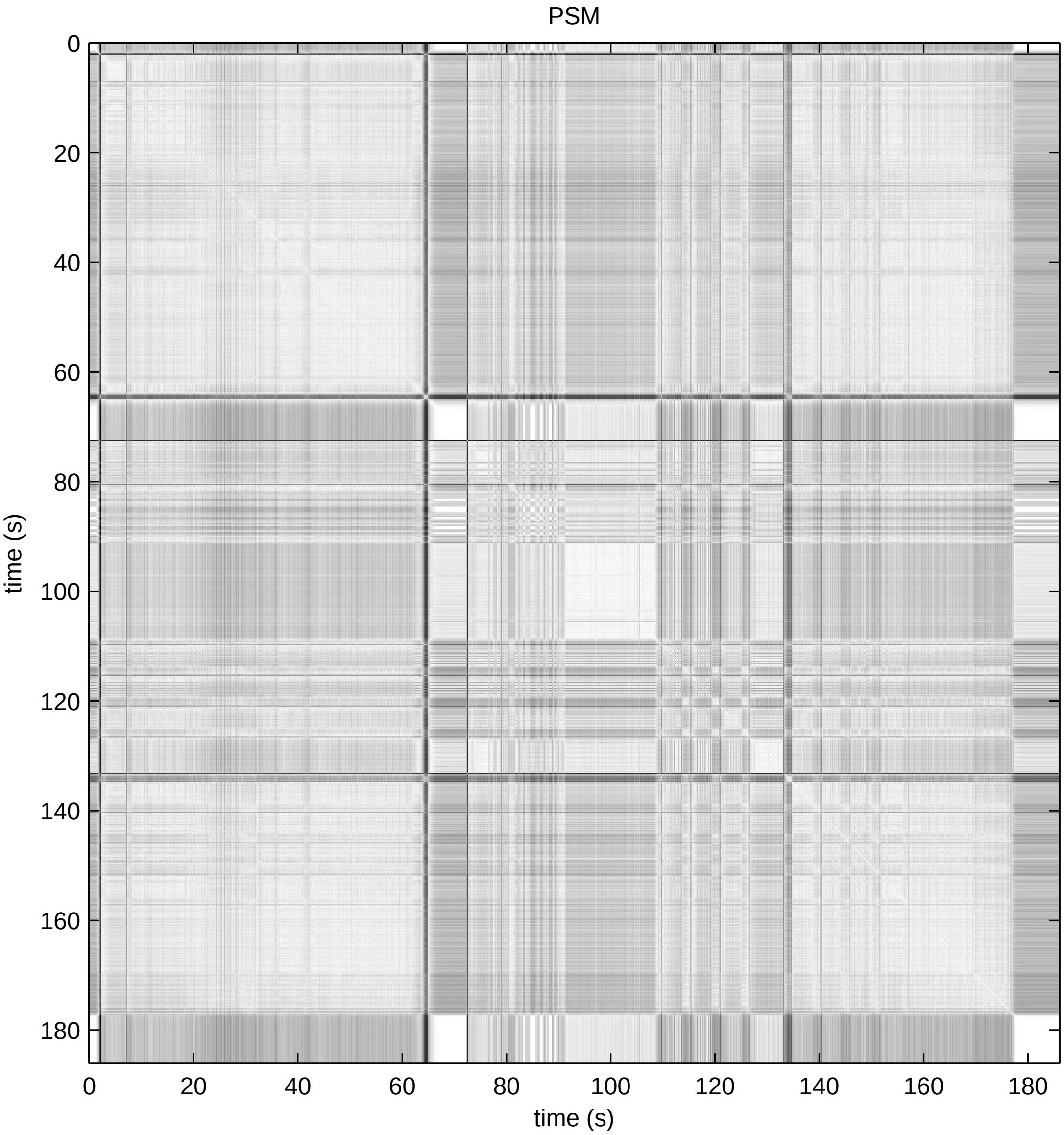


Figure 5.14: PSM of the full piece of music “Associations Libres” by Gilles Gobeil [87]. Decimation for this example was $l_w = 0.10$ s.

Foote and Cooper [61] utilize the presence of the white lines across small diagonal sections of their similarity matrix and scan through pieces of music with strong beat or pulse in small chunks to find the time signature of such pieces of music.

At this point, it is left to a user of the PSM to choose the sections that best display percussive self-similarity. The results of the PSM can be used to corroborate a human musical analysis or discover new facets of the musical structure. A new algorithm possibly could search for white (or lighter) lines in the PSM and choose an appropriate section of the music or an appropriate direction for taking a diagonal mean in order to automatically find percussivity self-similarity.

Chapter 6 summarizes this dissertation and points out its unique contributions. Further study is also considered.

Chapter 6

Summary and Concluding Remarks

Following the presentation of this dissertation's motivation and the background necessary for context, Chapter 3 described the collection of human percussion judgments. Synthesized snare-drum sounds were used as stimulus sounds for this collection, and the sounds were varied in two different dimensions for the three sets of stimulus sounds. The collection involved 29 participants assessing at least one of the three sets of 120 pairs of stimulus sounds. For each pair of stimulus sounds, a participant judged which sound was more like a single, damped, percussive event (SDPE). The grouped results of this collection show that rise time and string resonance could be used as a primary cue for sound percussivity. However, due to the complex nature of string resonance, it is not a feasibly measurable nor fundamental dimension of a sound. Gross spectral filtering also appears to affect the percussivity of a sound, but not as strongly as, nor in the presence of, rise time. Threshold judgments also were collected. These judgments specify a point in a ranked list of stimulus sounds where the participants judged the SDPE threshold to be.

Chapter 4 described the percussivity-profile algorithm (PPA). After a generic rise-time algorithm and a human hearing model were shown, the PPA was described in full detail. The tuning of the PPA according to the results of the collection of human percussion judgments was a significant part of its development. Practical considerations were shown which involved issues with memory usage and the final visual presentation of the percussivity profile. Finally, one constructed and two electroacoustic music examples of using the PPA to display percussive instants in music were shown.

Chapter 5 described the percussive similarity matrix (PSM) based on Foote's [57] similarity matrix for music. For its similarity metric, the PSM uses the Euclidean distance between time-window vectors of group percussivity profile (GPP). After some practical considerations were presented, the percussive similarity matrix was shown for one constructed example and two electroacoustic music examples.

6.1 Contributions

Several unique contributions to the body of knowledge were made during the course of this research. The collection of human percussion judgments was motivated by a need for a clear understanding of which sound dimensions can be correlated with human judgment of a sound event's percussivity. One sound dimension that is clearly correlated with percussivity and is easily measurable is rise time.

The percussivity profile is the result of the PPA. The percussivity profile is a new two-fold measure of the percussivity at any instant during a piece of music. It is intended to indicate those instants in music that humans also would identify as percussive. The two parts of the percussivity profile are the single-value percussivity profile (SPP) and the GPP matrix.

The PPA was tuned to make the most common participant choices for two of the stimulus sets in the collection of human percussion judgments. *This tuning of the PPA apparently represents the first percussion-detection algorithm that uses experimental results from a human-behavior study in order to determine the algorithm parameters for more human-like behavior.*

The PPA parameters that result in the performance most consistent with the participants in the human judgment collection suggest some ideas about human hearing. These suggestions, outlined in the following paragraphs, obviously are not facts, as a computer model is not the human ear, but these suggestions do warrant further investigation.

The value of 115 Hz for `lowFreq` (the lowest frequency analyzed) suggests that, at least for the stimulus sounds in sets A and C (those stimulus sets with rise time as a varied dimension, see Section 3.1.1), no sound below 115 Hz is necessary in order to discern the percussivity of a sound. The value of 95 channels for `nChannels` (the number of divisions of the basilar membrane model) suggests a frequency selectivity in the auditory periphery above 115 Hz for percussive sounds. The value of 0.0126 seconds for `windowTimeLen` (the time division for decimation and differentiation) suggests something about the time integration of the auditory periphery when presented with percussive sounds.

Perhaps the most remarkable result of the PPA tuning was the value of 6 groups for `nChannelGroups`. It was clearly the best choice under many different values of the other parameters. The actual value for `nChannelGroups` of 6 groups may be significant. This value suggests a concrete number of groups into which channels may be combined in some fashion during the human processing of percussive sounds. The fact that the optimal value for `nChannelGroups` is neither one nor `nChannels` is an indication that a grouping of channels in some fashion may occur in the human processing of percussive sounds. This non-boundary value for `nChannelGroups` validated the use of the GPP.

The utility of the GPP was demonstrated using a new tool that indicates self-similarity in the percussive instants of music, the percussive similarity matrix (PSM). Using the GPP as part of a similarity metric, the PSM indicates self-similarity based not only on the timing of percussive events, but also based on their GPP signature.

6.2 Suggestions for Further Study

The following paragraphs contain many suggestions by which the present research can be expanded. These include changes to the percussion-judgment collection, further measurements of success of the PPA and PSM, changes to the PPA and the PPA tuning, new dimensions of percussive events, a technique for processing stereo recordings, and changes to the display of the information contained in the PPA and PSM.

A personal trait that might affect how a person listens to percussive sounds is musical experience. In any further percussion judgment collection, more information about the musical experience of the participants would be useful, and processing the results of musical experts separately from others may prove informative.

The results of the percussion-judgment collection of Chapter 3 are significantly influenced by the choice of the base stimulus sound, and those results subsequently affect the performance of the tuned PPA and PSM. The synthesized snare drum is intended to be a quintessential example of a single, damped, percussive event (SDPE). Through this research, the concept of an SDPE has been shown to be possibly insufficient to describe all of the sound events which humans might label as percussive. Examples of these types of non-SDPE sound events were discussed in Section 4.5.3, and research should be done to first determine if humans would label them as percussive events. If humans would label them as percussive, then research should be done to characterize and detect these types of sound events.

Even if one accepts that SDPEs are the target of the PPA and that a synthesized snare drum represents an SDPE, as is done in this research, the choice of a synthesized snare drum carries with it inherent specifications of what is percussive by means of its sound spectrum over time. Indirectly, these specifications become expressed in the tuned algorithm parameters of the PPA. This effect was in fact desired on at least one level, but may be too specific in the exact details. Any particular choice of percussive base sound will create similar general specifications, but also will differ in the details. It is probable that other base stimulus sounds would generate at least slightly differing results for further percussion-judgment collections. These new collection results in turn would probably generate differing tuned algorithm parameters and performances for the PPA. By exploring the tuned PPA parameters for several different base sounds (for example, a click, a timpani drum, and a door slam), sets of effective PPA parameters might be found. As an example, if a click were used as the base stimulus sound, the tuned algorithm parameters might lead to PPA performance which would have identified the truncated marble strikes described in Section 4.5.2. By combining these different sets of algorithm parameters in some way, a more broadly scoped PPA might be designed.

Another method which might lead to a more robust PPA would be to first perform a percussion-judgment collection using a stimulus set comprised of completely different percussive and almost percussive sounds. This method may not easily demonstrate the effect of a single sound dimension as a percussive cue, but it might lead to a PPA which could identify more types of percussive sound events. In this case a neural-net design for the PPA seems appropriate. A neural net would unfortunately obfuscate the algorithm aspects that lead to better understanding of human sound processing.

Even when working specifically with a single base stimulus sound, several different new dimensions could be explored as well. The discussions with peers, references, and pilot work mentioned in Section 3.1.1 did not initially generate the following sound dimensions as candidates that might affect the percussivity of a sound event: loudness, fall time, and total sound length. Spectral centroid is another sound dimension that could be explored, though it is related to the gross spectral filtering which was researched. A sound dimension that could possibly have a strong effect on percussivity, but which might be harder to implement in a percussion-judgment collection is dynamic spectral filtering (or the related dimension of dynamic spectral centroid). Even within the sound dimensions already explored, shorter rise times should be added to the current set of stimulus dimension values, as pointed out by the extremely short rise times of the possibly percussive events in Section 4.5.2.

In this research an assumption is made that may reduce the effectiveness of the PPA. The assumption is that the results of the percussion-judgment collection based on the isolated stimulus sounds are also valid for similar sounds found in a musical environment. This assumption is almost certainly inaccurate and a mechanism for researching it would be useful. That mechanism might include a new way of collecting human percussion judgments.

A few additions could be made to the PPA that might improve its performance at identifying percussive instants. A weighting factor could be added to the channels or channel groups so that some channels or channel groups are more important when determining percussivity. Another possible addition is multiscaled rise-time detection across the frequency spectrum. Low-frequency sounds generally have a longer rise time than high-frequency sounds due to the time required to pass through a complete cycle of the waveform. Searching for different rise times across the channels of the cochlear model or across the channel groups might improve the performance of the PPA. Both of these examples would need to be tuned in order to achieve consistently good performance with the PPA.

One of the most important follow-up steps to this research is to answer the question as to exactly how well the PPA and PSM perform the tasks of finding percussive instants and percussive self-similarity in pieces of electroacoustic music. The metric against which they should be judged is human performance at the same task, and expert percussionists should be asked to annotate pieces of electroacoustic music for percussive instants. The collection of this type of information for evaluation of the PPA could be taken in a fashion similar to the collection of percussion annotation by Tanghe et al. [6]. These drum annotations of electroacoustic music could also be used as a new human-performance goal for the tuning of the PPA. New dimensions of sound events may also need to be explored in order to identify percussive events in pieces of music based on what is labelled as percussive by the experts.

The algorithm parameters of the PPA affects the form of the resulting percussivity profile. As an example, a shorter values of `windowTimeLen` tends to make the SPP appear more noisy and long values tend to make it appear smoother. Another area which would be worth researching is changing these parameters by hand in order to evaluate the effect each parameter has on the percussivity profile. The results of this new understanding of the algorithm parameters could be useful to try to tune the PPA performance by hand.

It would be useful to account for the repetition rate at which the integration of quickly repeated percussive strikes becomes a single sound rather than an experience of multiple strikes, such as a drum roll or static-like sounds. No attempt was made to deal with this phenomenon, although it affects the PPA performance in the difficult example in Section 4.5.3. Non-uniform spacing between the strikes may affect this experience as well.

A mechanism to more fully integrate the stereo (or multitrack) experience into the PPA would be useful. Problems arose with trying to process stereo sounds with the PPA. Monaural combination of the two channels worked well under many circumstances, but a few examples existed for which the experience of the monaural combined audio was significantly different from the separated stereo audio (for example, the piece "Superstrings" by Adrian Moore [93] [not shown]). This difference may have been due to out-of-phase versions of the same audio in the separated tracks, or simply different timing on repeated percussive sounds.

A final aspect of this research, which should be further explored, is the visual representation mentioned in Chapter 1. The percussivity profile and PSM are not the most intuitive representations of music. A graphic designer may be able to create a representation of percussivity and percussive self-similarity which would be more intuitive to general listeners.

Appendix A

Source Code

Partially due to Dixon's [37] comment that some of the onset-detection algorithms are sensitive to implementation details or parameter settings, the source code for the most important components of this dissertation is included here. A more complete set of source code used for this dissertation is available on the Internet at *http://academic.konfuzo.net*.

A.1 Csound Source Code

The following Csound orchestra and score file will generate the WAV file shown in the top plot of Figure 4.9.

A.1.1 snare-setC.orc

```
; csound
; John Anderson Mills III - nodog
; 2008-05-12
;
; snare-setC.orc
;

        sr      =       44100
        kr      =       44100
        ksmps   =       1
        nchnls  =       1

;=========================================================================
; snare attempt one
; taken from ohio players' "jive turkey"
; single fundamental with changing harmonic content
; filtered noise added on top
; another crazy amplitude envelope
; with added high pass filtering

                instr   74
;---initialization---------------------------------------------------------

        ;---from-score---------------------------------------------------
        idur    =       p3      ;duration
        idbamp  =       p4      ;amplitude
        irise   =       p5      ;rise time
        idecay  =       p6      ;decay time
        itnfq   =       p7      ;tone frequency
        ihicut  =       p8      ;hi cutoff freq
        ispktm  =       p9      ;spike time
        inssttm =       p10     ;noise start time
        inscyc  =       p11     ;number of noise cycles
        inscntm =       p12     ;noise continuous start time
        inspcnt =       p13     ;percent noise
        icentfq =       p14     ;center freq for the filter
        ibandw  =       p15     ;bandwidth of the filter
        inumlay =       p16     ;number of layers in the resonx filter

        ;---constants----------------------------------------------------
        iamp    =       ampdb(idbamp)   ;linear amplitude
        istst   =       idur - irise - idecay   ;steady state time
        isine   =       1       ;that's my sine wave
        icosine =       2       ;that's my cosine wave
        isoff   =       0.0     ;off
        ison    =       1.0     ;on
        iseoff  =       0.005   ;effective off for exponential envelopes
        ihalf   =       0.5     ;half
        ilseed  =       0.1     ;left random number seed
        irseed  =       0.9     ;right random number seed
        ipi     =       3.141592653     ;pi
        ihifq   =       20000   ;highest frequency heard
        inharm  =       ihifq/itnfq     ;number of harmonics in the gbuzz unit
        ionrat  =       0.92    ;not quite 1.0 for harmonic amplitude multiplier
        ioffrat =       0.2     ;not quite 0.0 for harmonic amplitude multiplier
        ilharm  =       1.0     ;lowest harmonic present
        iscl    =       1       ;scaling done by the resonx filter

        ;---initializaton------------------------------------------------

;---performance------------------------------------------------------------

        ;linear amplitude gate on entire sound (effects rise and final 4% of decay)
        klingt  linseg  isoff, irise, ison, istst, ison, idecay*0.96, ison, \
                        idecay*0.04, isoff, idecay, isoff
        ;exponential amplitude gate on the entire sound (controls decay only)
        kexpgt  expseg  ison, irise, ison, istst, ison, idecay, iseoff

        ;the real gate is the lesser of the linear and exponential
        agate   =       (klingt < kexpgt ? klingt : kexpgt)

        ;harmonic amplitude multiplier
        krat    linseg  ionrat, ispktm, ionrat, ispktm, ioffrat, idur-2*ispktm,
            ioffrat

        ;the fundamental tone is a gbuzz unit which has varying frequency content
        afund   gbuzz   ison, itnfq, inharm, ilharm, krat, icosine

        ;noise continuous gate
        ansctgt linseg  isoff, inssttm, isoff, inscntm-inssttm, ison, idur, ison

        ;noise cycle gate
        anscygt linseg  isoff, inssttm, ison, inscntm-inssttm, isoff, idur, isoff

        ;noise cycle signal (from 0 to 1 starting at 0)
        anscysg oscili  ison, inscyc/(inscntm-inssttm), icosine
        anscysg =       ihalf - anscysg*ihalf

        ;noise gate is a combination of these two
        ansgt   =       (ansctgt+(anscygt*anscysg))

        ;noise
        alns    randi   ansgt, ihicut, ilseed
        arns    randi   ansgt, ihicut, irseed

        ;putting it all together
        alsig   =       iamp*agate*((1.0-inspcnt)*afund+inspcnt/2.0*alns)
        arsig   =       iamp*agate*((1.0-inspcnt)*afund+inspcnt/2.0*arns)

        ;filtering
        alfilt  resonx  alsig, icentfq, ibandw, inumlay, iscl
        arfilt  resonx  arsig, icentfq, ibandw, inumlay, iscl

;               outs    alfilt, arfilt
                outs    alfilt

;-----------------------------------------------------------------------
                endin

;=======================================================================
```

A.1.2 stimulusCollage-setC.sco

```
; csound
; John Anderson Mills III - nodog
; 2007-02-02
;
; soundEvent-setC.sco

;------------------------------------------------------------------------
; functions

; sine function is  always number one in my book
;f01      start     size      gen       p1
f01       0.0       8192      10        1

; cosine function for gbuzz
;f02      start     size      gen       p1
f02       0.0       8192      9         1         1.0       90.0
;------------------------------------------------------------------------

t 0 60
;snare drum sounds
;i74      start     dur       amp       rise      decay     tnfq      hicut     spktm     nssttm    \
          nscyc     nscntm    nspcnt    icentfq   ibandw    inumlay
i74       0.1       0.360     88        0.01      0.30      158.0     15000     0.005     0.004     \
          5.0       0.030     1.0       1000      2000      1
f0        0.620
s
i74       0.1       0.360     85        0.01      0.30      158.0     15000     0.005     0.004     \
          5.0       0.030     1.0       10500     19000     1
f0        0.620
s
i74       0.1       0.360     91        0.01      0.30      158.0     15000     0.005     0.004     \
          5.0       0.030     1.0       500       1000      5
f0        0.620
s
i74       0.1       0.360     91        0.01      0.30      158.0     15000     0.005     0.004     \
          5.0       0.030     1.0       11500     16000     10
f0        0.620
s
i74       0.1       0.380     85        0.03      0.30      158.0     15000     0.005     0.004     \
          5.0       0.030     1.0       10500     19000     1
f0        0.620
s
i74       0.1       0.380     88        0.03      0.30      158.0     15000     0.005     0.004     \
          5.0       0.030     1.0       1000      2000      1
f0        0.620
s
i74       0.1       0.380     91        0.03      0.30      158.0     15000     0.005     0.004     \
          5.0       0.030     1.0       500       1000      5
```

```
f0      0.620
s
i74     0.1     0.380   91      0.03    0.30    158.0   15000   0.005   0.004   \
        5.0     0.030   1.0     11500   16000   10
f0      0.620
s
i74     0.1     0.400   88      0.05    0.30    158.0   15000   0.005   0.004   \
        5.0     0.030   1.0     1000    2000    1
f0      0.620
s
i74     0.1     0.400   85      0.05    0.30    158.0   15000   0.005   0.004   \
        5.0     0.030   1.0     10500   19000   1
f0      0.620
s
i74     0.1     0.400   91      0.05    0.30    158.0   15000   0.005   0.004   \
        5.0     0.030   1.0     500     1000    5
f0      0.620
s
i74     0.1     0.400   91      0.05    0.30    158.0   15000   0.005   0.004   \
        5.0     0.030   1.0     11500   16000   10
f0      0.620
s
i74     0.1     0.420   85      0.07    0.30    158.0   15000   0.005   0.004   \
        5.0     0.030   1.0     10500   19000   1
f0      0.620
s
i74     0.1     0.420   88      0.07    0.30    158.0   15000   0.005   0.004   \
        5.0     0.030   1.0     1000    2000    1
f0      0.620
s
i74     0.1     0.420   91      0.07    0.30    158.0   15000   0.005   0.004   \
        5.0     0.030   1.0     500     1000    5
f0      0.620
s
i74     0.1     0.420   91      0.07    0.30    158.0   15000   0.005   0.004   \
        5.0     0.030   1.0     11500   16000   10
f0      0.620
e
```

A.2 MATLAB Source Code

The following MATLAB source code is the percussivity-profile algorithm. Several functions from Slaney's Auditory Toolbox [73] are necessary to run the code. The Auditory Toolbox is available at *http://www.slaney.org/malcolm/pubs.html*.

A.2.1 percussivityProfile.m

```
function [ percProfile , groupPercProfile ] = percussivityProfile( ...
  soundFilename , debug , ...
  nChannels , lowFreq , windowTimeLen , hairCellScaling , nChannelGroups , ...
  nStartAvgs );

% PERCUSSIVITYPROFILE
% returns two matrices:
%- a 2 by N matrix where the first row is a measure of percussivity for the
%   current instant in the WAV file and the second row is the time of the
%   beginning of the timeframe.
%- a M by N matrix where each row is a measure of percussivity for the current
%   instant in the WAV file for the current timeframe and group of neural
%   channels.
%
% One can always specify the function parameters in the function call, or
% one can create a defaults.mat file with the last six parameters as variables
% and the function will load those if they aren't specified.
%
% soundFilename - the filename of the sound to process
% debug - a value of 1 turns on debugging output
% nChannels - number of divisions of the basilar membrane model
% lowFreq - the lowest frequency analyzed by the basilar membrane model
% windowTimeLen - the time division for decimation and differentiation
% hairCellScaling - the apparent amplitude of the sound file
% nChannelGroups  - number of groups that the channels are put into
%
% example:
% [ p, g ] = percussivityProfile( 'gobeil-first30-mono.wav' )
% plot( p( 2, : ), p( 1, : ) )
% plot( p( 2, : ), g( 1, : ), p( 2, : ), g( 2, : ) )

% Dependent on custom functions:
% calcRealignShiftSamps

% Dependent on Auditory Toolbox functions:
% MakeERBFilters
% ERBFilterBank
% MeddisHairCell

% dissertation research
% John Anderson Mills III - nodog
% 2007-11-08

% CONSTANTS ==========================================================
% Take care of default parameters (from optimization).
if nargin < 8, load defaults nStartAvgs; end
if nargin < 7, load defaults nChannelGroups; end
if nargin < 6, load defaults hairCellScaling; end
if nargin < 5, load defaults windowTimeLen; end
if nargin < 4, load defaults lowFreq; end
if nargin < 3, load defaults nChannels; end
if nargin < 2, debug = 1; end
if nargin < 1, disp( 'soundfile must be given.' ); return; end

subtractSpontaneous = 1;   % subtract the spont firing rate from the hair cell
maxFiringRate = 1000;          % assumed max firing rate for a neuron

firstOrder = 1;                % difference order
acrossCol = 1;                 % dimension direction
acrossRow = 2;                 % dimension direction
acrossPln = 3;      % dimension direction
switchRowCol = [ 2 1 3 ];   % use to switch the rows and columns of a 3d matrix

% Measure the compute time.
if debug, startTime = cputime; end;

% Acquire input data (the soundfile) -----------------------------------
[ soundEvent, sampFreq, nBits ] = wavread( soundFilename );
nSampsSoundEvent = size( soundEvent, 1 );

% CALCULATIONS (find the sound file's rating) ==========================

% Cochlear Reaction Step -----------------------------------------------
if debug, tic; disp( '  step - cochlear response'); end;

% Determine the filter coefficients.
filtCoefs = MakeERBFilters( sampFreq, nChannels, lowFreq );

% Calculate the cochlear reaction to the input stimulus.
cochlea = ERBFilterBank( soundEvent, filtCoefs );
[ cochleaRows, cochleaCols ] = size( cochlea );
clear( 'soundEvent', 'filtCoefs' );

% Zeropadding Step -----------------------------------------------------
if debug, toc; tic; disp( '  step - zeropadding'); end;

% Zero pad the cochlea response to prepare for time realignment
% Calculate the upcoming time realignment shifts.
realignShiftSamps = calcRealignShiftSamps( lowFreq, sampFreq, nChannels );
maxRealignShiftSamps = max( realignShiftSamps );
% Window samps for lowpass avging
nSampsLPWin = ceil( windowTimeLen*sampFreq );
% Zero pad with ( first multiple of nSampsLPWin > maxRealignShiftSamps ) + 1
nSampsCochZeroPad = nSampsLPWin*( ceil( maxRealignShiftSamps/nSampsLPWin ) + 1 );
cochlea = [ zeros( nChannels, nSampsCochZeroPad ) cochlea ];

% Hair Cell Reaction Step ----------------------------------------------
if debug, toc; tic; disp( '  step - hair cell response'); end;
```

```
% Modify the cochlea response into the reaction of the hair cell.
% The output of MeddisHairCell is time vs firing rate (spikes/sec) so it is
% non-dimensionalized by multiplying by (time/maxFiringRate).
hairCell = ( windowTimeLen/maxFiringRate )* ...
  MeddisHairCell( cochlea*hairCellScaling, sampFreq, subtractSpontaneous );
[ nHairCellRows, nHairCellCols ] = size( hairCell );
clear( 'cochlea' );

% Time Realignment Step ---------------------------------------------------
if debug, toc; tic; disp( '  step - time realignment'); end;

% shift the hairCells a number of samples according to the cf of the channel
% and remove the rate of firing at the first sample to avoid a false indication

for iChannel = 1:nChannels
  % shift
  hairCell( iChannel, : ) = ...
    circshift( hairCell( iChannel, : ), [ 0 -realignShiftSamps( iChannel ) ] );
  % remove the firing rate at the first sample from the entire channel to avoid
  % spurious artifacts of low pass filtering the initial value.
  hairCell( iChannel ) = hairCell( iChannel ) - hairCell( iChannel, 1 );
end

% Lowpass Filter Step -----------------------------------------------------
if debug, toc; tic; disp( '  step - lowpass filter'); end;

% Lowpass filter the hairCell data for decimation (following Slaney's example)
% Note that Slaney's example does *not* preserve amplitude at all, so
% this is reverse engineering his example into a more useful form.
cutoff = 1/nSampsLPWin;
filtB = [ 2*cutoff ];
filtA = [ 1 -1*( 1 - filtB ) ];
LPHairCell = filter( filtB, filtA, hairCell, [], acrossRow );

% Copy with Time-Shift and Decimation Step ---------------------------------
if debug, toc; tic; disp( '  step - copy with time-shift and decimation'); end;

% This creates nStartAvgs planes of lowpass-filtered, time-shifted, decimated
% hairCell response. Each plane is the hairCell data time-shifted by
% startShiftLength = windowTimeLen/nStartAvgs.

% shift length for averaging different start points together
startShiftLength = floor( nSampsLPWin/nStartAvgs );

% Now decimate and shift the lowpassed hair cell data
LPDecHairCell = zeros( ...
  nHairCellRows, ceil( nHairCellCols/nSampsLPWin ), nStartAvgs );

for iShift = 1:nStartAvgs
```

```
  if debug, toc; tic; disp( sprintf( '      shift %d of %d', iShift, nStartAvgs ) );
      end;
  tempLPHairCell = circshift( LPHairCell, [ 0 -iShift*startShiftLength ] );
  LPDecHairCell( :, :, iShift ) = ...
    tempLPHairCell(:, 1:nSampsLPWin:nHairCellCols );

end  % iShift

clear( 'hairCell', 'LPHairCell', 'tempLPHairCell' );
[ LPDecHairCellRowSize, LPDecHairCellColSize, LPDecHairCellPlnSize ] = ...
  size( LPDecHairCell );

% Differentiation Step ————————————————————————————————————————
if debug, toc; tic; disp( '   step - differentiation'); end;

% now we need the slope of the haircell activity
% diff is a "difference" so to make it a slope one must divide by windowTimeLen
diffAvgHairCell= diff( LPDecHairCell, firstOrder, acrossRow )/windowTimeLen;
clear( 'LPDecHairCell' );

% Half-Wave Rectification Step ————————————————————————————————
if debug, toc; tic; disp( '   step - half-wave rectification'); end;

% we're only interested in the onset, so halfwave rectify
halfDiffAvgHairCell = diffAvgHairCell;
halfDiffAvgHairCell( find( halfDiffAvgHairCell < 0 ) ) = 0;
[ nHalfDiffAvgHairCellRows, nHalfDiffAvgHairCellCols, ...
  nHalfDiffAvgHairCellPlns ] = size( halfDiffAvgHairCell );
clear( 'diffAvgHairCell' );

% Channel Group Means Step ————————————————————————————————————
if debug, toc; tic; disp( '   step - channel groups'); end;

% Create groups of channels from the nChannels, and get the average response
% for each group (the average response is used because different groups can
% have different numbers of channels in the group).
groupingFactor = nChannels/nChannelGroups;
groupOnsetIndicator = ...
  zeros( nChannelGroups, nHalfDiffAvgHairCellCols, nStartAvgs );

for iChannelGroup = 1:nChannelGroups

  groupStart = floor( ( iChannelGroup - 1 ) * groupingFactor ) + 1;
  groupEnd = floor( iChannelGroup * groupingFactor );

  % Take the average across each channel group
  groupOnsetIndicator( iChannelGroup, :, : ) = ...
    mean( halfDiffAvgHairCell( groupStart:groupEnd, :, : ), acrossCol );
```

```
end  % iChannelGroup

clear( 'halfDiffAvgHairCell' );

% Upsample and Unshift Step ------------------------------------------------
if debug, toc; tic; disp( '  step - upsample and unshift'); end;

% In order to properly average across the different time offsets, the different
% planes need to be time offset correctly.  This first requires a upsampling by
% repeating elements by a factor of nStartAvgs.
dummy = repmat( groupOnsetIndicator, nStartAvgs, 1 );
upSampGroupOnsetIndicator = reshape( dummy, ...
  nChannelGroups, nStartAvgs*nHalfDiffAvgHairCellCols, nStartAvgs );
clear( 'dummy' );

% Then the planes must be shifted for their different time offsets
shiftedGroupOnsetIndicator = zeros( size( upSampGroupOnsetIndicator ) );

for iShift = 1:nStartAvgs

  % shift the upSampGroupOnsetIndicator exactly iShift samples later in time
  shiftedGroupOnsetIndicator( :, :, iShift ) = ...
    circshift( upSampGroupOnsetIndicator( :, :, iShift ), [ 0 iShift ] );

end  % iShift

clear( 'upSampGroupOnsetIndicator' );

% Maximums of Channel-Group Means Step ------------------------------------
if debug, toc; tic; disp( '  step - maximums of channel-group means'); end;

% The ousetIndicator for each plane is the max of the group values.
onsetIndicator = max( shiftedGroupOnsetIndicator, [], acrossCol );
[ onsetIndicatorCols ] = size( onsetIndicator, acrossRow );

% Means of Time-Shifted Copies Step ---------------------------------------
if debug, toc; tic; disp( '  step - means of time-shifted copies'); end;

% The averagedOnsetIndicator is the average across planes.
averagedOnsetIndicator = mean( onsetIndicator, acrossPln );

% Average across planes for the multigroup percprofile.
averagedGroupOnsetIndicator = mean( shiftedGroupOnsetIndicator,  acrossPln );

% OUTPUT ==================================================================
% put the percProfile in its final form (removing zeropadding).

% Calculate indicator frames to remove for zero padding the percProfile
nFramesZeroPad = nStartAvgs*floor( nSampsCochZeroPad/nSampsLPWin );
```

```
endTime = ( nSampsSoundEvent - 1 )/sampFreq;
nPercProfileCols = ( onsetIndicatorCols - nFramesZeroPad );
percProfile = [ ...
  averagedOnsetIndicator( ( nFramesZeroPad + 1 ):onsetIndicatorCols ); ...
  [ 0:endTime/( nPercProfileCols - 1 ):endTime ] ];
groupPercProfile = ...
  averagedGroupOnsetIndicator( :, ( nFramesZeroPad + 1 ):onsetIndicatorCols );

% Show the compute time.
if debug
  toc;
  disp( sprintf( '  percussivityProfile took %d seconds to complete.', ...
    round( cputime - startTime ) ) );
end;  % if debug
```

A.2.2 calcRealignShiftSamps.m

```
function shiftSamps = calcRealignShiftSamps( lowFreq, sampFreq, nChannels )

% CALCREALIGNSHIFTSAMPS
% calculates the number of samples in each channel necessary to time realign
% due to the shift imposed by the Gammatone filterbank.

% Dependent on custom functions:

% Dependent on Auditory Toolbox functions:
% ERBSpace

% dissertation research
% John Anderson Mills III - nodog
% 2007-11-19

nr_cycles = 2;

% shift the hairCells a number of samples according to the cf of the channel
centerFreqs = ERBSpace( lowFreq, floor( sampFreq/2 ), nChannels );

% This is taken almost directly from the A.I.M. code of Stefan Bleeck
EarQ = 9.26449;         %  Glasberg and Moore Parameters
minBW = 24.7;
order = 4;
ERB = ((centerFreqs/EarQ).^order + minBW^order).^(1/order);
b=1.019.*ERB;
B=1.019*2*pi.*ERB;
envelopecomptime=(order-1)./B;
phasealign=-2*pi.*centerFreqs.*envelopecomptime;
phasealign=mod(phasealign,2*pi);
```

```
phasealign=phasealign./(2*pi.*centerFreqs);

shiftSamps = zeros( nChannels, 1 );

for iChannel = 1:nChannels
  % use this line for gross structure and local phase
  shiftTime = envelopecomptime( iChannel ) + phasealign( iChannel );
  % use this line for gross structure only.
  %shiftTime = envelopecomptime( iChannel );
  % use this line for a number of the wavelength cycles.
  %shiftTime = nr_cycles/centerFreqs( iChannel );
  shiftSamps( iChannel ) = round( shiftTime*sampFreq );
end
```

Bibliography

[1] T. Licata, ed., *Electroacoustic Music: Analytical Perspectives.* Westport, Connecticut: Greenwood Press, 2002.

[2] J.-C. Risset, “Forward,” in *Electroacoustic Music* (T. Licata, ed.), pp. xiii–xviii, Westport, Connecticut: Greenwood Press, 2002.

[3] D. Berezan, “Graphical scores in electroacoustic music: History, developments, and personal experience.” website http://www.davidberezan.com/homepage/Graphical Scores.html. accessed 2008-05-07.

[4] D. Berezan, “Unheard voices, ancient spaces: An acousmatic composition for eight channel digital tape and eight loudspeakers,” Master’s thesis, University of Calgary, March 2000.

[5] J. A. Mills III, “The application of psychoacoustic audio analysis techniques to electroacoustic music for the purpose of visualization,” in *147th meeting of the Acoustical Society of America*, May 25, 2004.

[6] K. Tanghe, M. Lesaffre, S. Degroeve, M. Leman, B. D. Baets, and J.-P. Martens, “Collecting ground truth annotations for drum detection in polyphonic music,” in *ISMIR 2005: Proceedings of the 6th International Conference on Music Information Retrieval, Queen Mary, University of London and Goldsmiths College, September 11-15, 2005* (J. D. Reise and G. A. Wiggins, eds.), (London), University of London, September 2005.

[7] F. C. Mish, ed., *Webster’s Ninth New Collegiate Dictionary.* Springfield, Massachusetts: Merriam-Webster, 1985.

[8] B. Kostek, *Perception-Based Data Processing in Acoustics: Applications to Music*

Information Retrieval and Psychophysiololgy of Hearing, vol. 3 of *Studies in Computational Intelligence.* Berlin: Springer, 2005.

[9] T. D. Rossing, *The Science of Sound.* Reading, Massachusetts: Addison-Wesley Publishing Company, 2nd ed., 1990.

[10] S. Emmerson and D. Smalley, "Electro-acoustic music," in *The New Grove Dictionary of Music and Musicians* (S. Sadie and J. Tyrrell, eds.), London: Macmillan, second ed., 2001.

[11] M. Simoni, "Acknowledgments," in *Analytical Methods of Electroacoustic Music* (M. Simoni, ed.), p. vii, New York: Routledge Taylor and Francis Group, 2006.

[12] L. Camilleri and D. Smalley, "Introduction," *Journal of New Music Research*, vol. 27, no. 1–2, pp. 3–12, 1998.

[13] M. Simoni, ed., *Analytical Methods of Electroacoustic Music.* New York: Routledge Taylor and Francis Group, 2006.

[14] D. Smalley, "Spectromorphology: explaining sound-shapes," *Organized Sound*, vol. 2, pp. 107–126, 1997.

[15] W. Moss, "Oscillococcinum," in *Music from SEAMUS*, vol. 10, Los Angeles: SEAMUS, 2001. CD format.

[16] B. Bossis, "The analysis of electroacoustic music: from sources to invariants," *Organised Sound*, vol. 11, no. 2, pp. 101–112, 2006.

[17] J. O. Pickles, *An Introduction to the Physiology of Hearing.* London: Academic Press, second ed., 1988.

[18] B. C. J. Moore, *An Introduction to the Psychology of Hearing.* San Diego: Academic Press, 4th ed., 1997.

[19] S. Mallat, *Tissues and Organs: A Text Atlas of Scanning Electron Microscopy.* San Francisco: W. H. Freeman and Company, 1979.

[20] A. F. Ryan and P. Dallos, "Physiology of the cochlea," in *Hearing Disorders* (J. L. Northern, ed.), pp. 253–266, Boston: Little Brown, 1984.

[21] D. W. Fawcett, *A Textbook of Histology.* Philadelphia: W. B. Saunders, 1986.

[22] G. von Békésy, *Experiments in Hearing.* New York: McGraw-Hill, 1960.

[23] T. D. Rossing, *Science of Percussion Instruments.* River Edge, New Jersey: World Scientific Publishing Co. Pte. Ltd., 2000.

[24] K. Ohta, S. Kuwano, and S. Namba, "Sound quality of implusive sounds in relation to their physical properties," in *Technology Reports of The Osaka University*, vol. 49, no. 2360, pp. 189–199, October 1999.

[25] S. Lakatos, "A common perceptual space for harmonic and percussive timbres," *Perception and Psychophysics*, vol. 62, no. 7, pp. 1426–1439, 2000.

[26] S. McAdams, A. Chaigne, and V. Roussarie, "The psychomechanics of simulated sound sources: material properties of impacted bars," *Journal of the Acoustical Society of America*, vol. 115, no. 3, pp. 1306–1320, 2004.

[27] B. Giordano, *Sound source perception in impact sounds.* PhD thesis, Università Degli Studi Di Padova, June 2005.

[28] M. Fingerhut, "Music information retrieval, or how to search for (and maybe find) music and do away with incipits," (Oslo, Norway), International Association of Music Libraries-International Association of Sound and Audiovisual Archives 2004 Congress, August 8–13 2004.

[29] A. Klapuri, A. Eronen, and J. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Transactions Speech and Audio Processing*, vol. 14, no. 1, 2006.

[30] J. Laroche, "Efficient tempo and beat tracking in audio recordings," *Journal of the Audio Engineering Society*, vol. 51, pp. 226–233, April 2003.

[31] A. Zils, F. Pachet, O. Delerue, and F. Gouyon, "Automatic extraction of drum tracks from polyphonic music signals," in *Proceedings of the 2nd International Conference on Web Delivering of Music*, (Darmstadt, Germany), December 9–11 2002.

[32] V. Sandvold, F. Gouyon, and P. Herrera, "Percussion classification in polyphonic audio recordings using localized sound models," in *Proceedings of the 5th International Conference on Music Information Retrieval* (C. L. Buyoli and R. Loureiro, eds.), (Barcelona), pp. 537–540, Universitat Pompeu Fabra, October 10–14 2004.

[33] J. P. Bello, G. Monti, and M. Sandler, "Techniques for automatic music transcription," in *ISMIR 2000 (Music IR 2000): Proceedings of the 1st International Conference on Music Information Retrieval, Plymouth (Massachusetts), October 23, 2000 - October 25, 2000* (D. Byrd, ed.), (Plymouth, Massachusetts), October 2000.

[34] G. M. J. dos Reis and F. F. de Vega, "A novel approach to automatic music transcription using electronic synthesis and genetic algorithms," in *Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*, (London), pp. 2915–2922, 2007.

[35] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions on Speech and Audio Processing*, vol. 13, pp. 1035–1047, September 2005.

[36] N. Collins, "Comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions," in *Audio Engineering Society 118th Convention*, (Barcelona, Spain), May 28–31 2005.

[37] S. Dixon, "Onset detection revisited," in *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-06)*, (Montreal), pp. 133–137, September 18–20, 2006.

[38] W. You and R. B. Dannenberg, "Polyphonic music note onset detection using semi-supervised learning," in *ISMIR 2007: Proceedings of the 8th International Conference on Music Information Retrieval, Vienna, Austria, September 23-27* (S. Dixon and D. B. R. Typke, eds.), (Vienna), pp. 279–282, sterreichische Computer Gesellschaft, September 2007.

[39] D. FitzGerald, *Automatic Drum Transcription and Source Separation.* PhD thesis, Dublin Institute of Technology, 2004.

[40] C. Uhle, C. Dittmar, and T. Sporer, "Extraction of drum tracks from polyphonic music using independent subspace analysis," in *4th International Symposium on Independent Component Analysis and Blind Signal Separation*, (Nara, Japan), April 2003.

[41] A. Nesbit, L. Hollenberg, and A. Senyard, "Toward automatic transcription of australian aboriginal music," in *ISMIR 2004: Proceedings of the 5th International Conference on Music Information Retrieval, Universitat Pompeu Fabra, Barcelona,*

Spain, October 10-14, 2004 (C. L. Buyoli and R. Loureiro, eds.), (Barcelona), Universitat Pompeu Fabra, October 2004.

[42] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An experimental comparison of audio tempo induction algorithms," *IEEE Transactions on Speech and Audio Processing*, vol. 14, no. 5, 2006.

[43] E. D. Scheirer, *Music-Listening Systems.* PhD thesis, Massachusetts Institute of Technology, June 2000.

[44] D. van Steelant, K. Tanghe, S. Degroeve, B. D. Baets, M. Leman, J.-P. Martens, and T. D. Mulder, "Classification of percussive sounds using support vector machines," in *Proceedings of the Annual Machine Learning Conference of Belgium and The Netherlands, Brussels, Belgium, January 8-9, 2004*, January 2004.

[45] F. Gouyon, F. Pachet, and O. Delerue, "On the use of zero-crossing rate for an application of classification of percussive sounds," in *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00), Verona, Italy, December 7-9, 2000*, December 2000.

[46] A. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound.* Cambridge, Massachusetts: The MIT Press, first ed., 1990.

[47] B. Arons, "A review of the cocktail party effect," *Journal of the American Voice Input/Output Society*, vol. 12, pp. 35–50, July 1992.

[48] A. D. Pierce, *Acoustics: An Introduction to Its Physical Principles and Applications.* Woodbury, New York: Acoustical Society of America, 1989.

[49] G. J. Brown and M. Cooke, "Computational auditory scene analysis," *Computer Speech and Language*, vol. 8, no. 4, pp. 297–336, 1994.

[50] D. P. W. Ellis, *Prediction Driven Computational Auditory Scene Analysis.* PhD thesis, Massachusetts Institute of Technology, June 1996.

[51] M. Goto, *Analysis of Musical Audio Signals*, ch. 8, pp. 251–296. Hoboken, New Jersey: Wiley-Interscience, 2006.

[52] A. Moreau and A. Flexer, "Drum transcription in polyphonic music using non-negative matrix factorisation," in *ISMIR 2007:Proceedings of the 8th International*

Conference on Music Information Retrieval, Vienna, Austria, September 23-27 (S. Dixon and D. B. R. Typke, eds.), (Vienna), pp. 353–354, sterreichische Computer Gesellschaft, September 23–27 2007.

[53] J. C. Brown and M. S. Puckette, "An efficient algorithm for the calculation of a constant q transform," *Journal of the Acoustical Society of America*, vol. 92, no. 5, pp. 2698–2701, 1992.

[54] R. Kronland-Martinet and A. Grossman, *Application of time-frequency and time-scale methods (wavelet transforms) to the analysis, synthesis, and transformation of natural sounds*, pp. 45–85. Cambridge, Massachusetts: MIT Press, 1991.

[55] R. Meddis and L. O'Mard, "Psychophysically faithful methods for extracting pitch," in *Computational Auditory Scene Analysis* (D. F. Rosenthal and H. G. Okuno, eds.), ch. 4, pp. 43–58, Mahwah, New Jersey: Lawrence Erlbaum Associates, first ed., May 1998.

[56] P. Desain and H. Honing, "Can music cognition benefit from computer music research? from foot-tapper systems to beat induction models," in *Proceedings of the 1994 International Conference of Music Perception and Cognition*, (Liege BE), pp. 397–398, 1994.

[57] J. Foote, "Visualizing music and audio using self-similarity," in *Proceedings of ACM Multimedia '99*, (Orlando, Florida), pp. 77–80, November 1999.

[58] K. Church and J. Helfman, "Dotplot: A program for exploring self-similarity in millions of lines of text and code," *Journal of the American Statistical Association*, vol. 2, no. 2, pp. 153–174, 1993.

[59] M. Bartsch and G. H. Wakefield, "To catch a chorus: Using chroma-based representations for audio thumbnailing," in *Proc. Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 15–18, IEEE, October 21–24, 2001.

[60] Y. Shiu, H. Jeong, and C.-C. J. Kuo, "Similarity matrix processing for music structure analysis," in *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, (Santa Barbara, California), pp. 69–76, Association for Computing Machinery, 2006.

[61] J. Foote and M. Cooper, "Visualizing musical structure and rhythm via self-similarity," in *Proceedings of the International Conference on Computer Music (ICMC)*, (Habana, Cuba), September, 12, 2001.

[62] M. Cooper and J. Foote, "Summarizing popular music via structural similarity analysis," in *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, October, 19, 2002.

[63] R. B. Dannenberg and N. Hu, "Pattern discovery techniques for music audio," in *ISMIR 2002: [Proceedings of the] Third International Conference on Music Information Retrieval, Ircam - Centre Pompidou, Paris, France, October 13-17, 2002* (M. Fingerhut, ed.), (Paris), pp. 63–70, Ircam - Centre Pompidou, October 2002.

[64] W. Chai, *Automated Analysis of Musical Structure*. PhD thesis, Massachusetts Institute of Technology, September 2005.

[65] J. A. Mills III, "The application of psychoacoustic audio analysis techniques to electroacoustic music for the purpose of visualization (human judgment collection)," in *153rd meeting of the Acoustical Society of America*, June 5, 2007.

[66] D. W. Martins, *Doing Psychology Experiments*. Pacific Grove, California: Brooks/-Cole Publishing Company, 1996.

[67] R. Boulanger, ed., *The Csound Book*. Cambridge, Massachusetts: The MIT Press, first ed., 2000.

[68] C. Dodge and T. A. Jerse, eds., *Computer Music: Synthesis, Composition, and Performance*. New York: Schirmer Books, 2nd ed., 1997.

[69] M. R. Spiegel, *Mathematical Handbook of Formulas and Tables*. San Francisco: McGraw-Hill, 27th ed., 1968.

[70] G. G. Judge, R. C. Hill, W. E. Griffiths, H. Lutkepohl, and T.-C. Lee, *Introduction to the Theory and Practice of Econometrics*. New York: Wiley, 2nd ed., 1988.

[71] M. Hollander and D. A. Wolfe, *Nonparametric Statistical Methods*. New York: Wiley, 2nd ed., 1999.

[72] J. A. Mills III, "The application of psychoacoustic audio analysis techniques to electroacoustic music for the purpose of visualization (percussion onset detection)," in *151st meeting of the Acoustical Society of America*, June 6, 2006.

[73] M. Slaney, D. Naar, and R. F. Lyon, "Auditory model inversion for sound separation," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing '94*, (Adelaide, Austrailia), pp. II–77–II–80, 1994.

[74] R. D. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice, "Spiral vos final report, part a: The auditory filterbank," Contract Report APU 2341, Cambridge Electronic Design, 1988.

[75] M. Mathews, "The ear and how it works," in *Music, Cognition, and Computerized Sonud, An Introduction to Psychoacoustics* (P. Cook, ed.), pp. 1–10, Cambridge, Massachusetts: The MIT Press, 1999.

[76] S. Bleeck, T. Ives, and R. D. Patterson, "Aim-mat: the auditory image model in matlab," *Acta Acustica*, vol. 90, pp. 781–788, 2004.

[77] R. Meddis, "Stimulation of mechanical to neural transduction in the auditory receptor," *Journal of the Acoustical Society of America*, vol. 79, pp. 702–711, March 1986.

[78] P. G. Zimbardo and R. J. Gerrig, *Psychology and Life.* New York: Harper Collins, 1996.

[79] S. Uppenkamp, S. Fobel, and R. D. Patterson, "The effects of temporal asymetry on the detection and perception of short chirps," *Hearing Research*, vol. 158, pp. 71–83, 2001.

[80] T. Dau, O. Wegner, V. Mellert, and B. Kollmeier, "Auditory brainstem responses (abr) with optimized chirp signals compensating basilar membrane dispersion," *Journal of the Acoustical Society of America*, vol. 107, pp. 1530–1540, 2000.

[81] O. Wegner and T. Dau, "Frequency specificity of chirp-evoked auditory brainstem responses," *Journal of the Acoustical Society of America*, vol. 111, pp. 1318–1329, 2002.

[82] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge, United Kingdom: Cambridge University Press, second ed., 1992.

[83] G. Forsythe, M. Malcolm, and C. Moler, *Computer Methods for Mathematical Computations.* Englewood Cliffs, New Jersey: Prentice Hall, 1976.

[84] D. T. Blackstock, *Fundamentals of Physical Acoustics.* New York: Wiley-Interscience, 2000.

[85] M. Schroeder and J. L. Hall, “Model for mechanical to neural transduction in the auditory receptor,” *Journal of the Acoustical Society of America*, vol. 55, pp. 1055–1060, 1974.

[86] Åke Parmerud, *Invisible Music.* Stockholm: Phono Suecia, 1995. CD format.

[87] G. Gobeil, *La mecanique des ruptures.* Montreal: empreintes DIGITALes, 1994. CD format.

[88] B. Eno, “Ambient 1: Music for airports,” in *Ambient 1: Music for Airports*, UK: Editions EG Records, 1990. CD format.

[89] M. Cooper and J. Foote, “Automatic music summarization via similarity analysis,” in *ISMIR 2002: Proceedings of the Third International Conference on Music Information Retrieval, Ircam - Centre Pompidou, Paris, France, October 13-17, 2002* (M. Fingerhut, ed.), (Paris), pp. 81–85, Ircam - Centre Pompidou, October 2002.

[90] The Beatles, “Magical mystery tour,” in *Magical Mystery Tour*, Los Angeles: Capitol, 1990. CD format.

[91] B. Schutz, *Geometrical Methods of Mathematical Physics.* Cambridge, Unitedy Kingdom: Cambridge University Press, 1980.

[92] C. M. Grinstead and J. L. Snell, *Introduction to Probability.* Providence, Rhode Island: American Mathematical Society, 2nd ed., 1997.

[93] A. Moore, “Superstrings,” in *Music from SEAMUS*, vol. 10, Los Angeles: SEAMUS, 2001. CD format.

Vita

John Anderson Mills III was born in Sumter, South Carolina, on September 11, 1970, the son of Marie Eaddy Mills and John Anderson Mills Jr. He worked as an electrician's assistant at the family electrical construction company, Mills Electric Company, Inc., until graduating as valedictorian from Wilson Hall Academy in 1988. He then went to Clemson University where he graduated magna cum laude with a degree in Electrical and Computer Engineering in 1992. He spent the summers interning at Sun Data, Inc. in Atlanta, Georgia as a personal computer specialist. He entered the Graduate Program in Acoustics at The Pennsylvania State University in State College, Pennsylvania in 1992. He was awarded a Master of Science degree in 1997. During the following year, he designed and implemented an electrical-construction bidding system for Mills Electric Company, Inc., moved to Austin, Texas, and then enrolled in the graduate program at The University of Austin in 1998. He was an Assistant Instructor alternately teaching a course he designed, Acoustics for Musicians and Recording Engineers, and in the Electrical Engineering Senior Design Lab. He also worked as a Graduate Assistant as a high performance computing system administrator for the Mechanical Engineering Department.

Permanent Address: 5 Cassena Ct.
Sumter, SC 29150

This dissertation was typeset with $\LaTeX\,2_\varepsilon$by the author.