HOW DO WE RECOGNIZE A SONG IN ONE SECOND?

The Importance of Salience and Sound in Music Perception

Alexander Refsum Jensenius

Abstract

This project started with the observation that we manage to recognize a song by listening to only a second of it. What perceptual and musical features make this possible, and can such features be used in music analysis and music information retrieval? These questions can be broken down to two main problems: a) segregation of sensory input and b) recognition of musical features. The segregation of musical information from a complex soundscape is discussed with reference to theories of auditory scene analysis and music perception. A problem is that there is still no good way to make computers separate sound streams in a way similar to human perception. When it comes to the recognition process, the thesis focuses on what musical features make a song more or less recognizable. It is argued that a song is recognized quicker if there is some *salient*, or perceptually significant, feature present. Then it is shown how salience points can be analysed with reference to traditional musical parameters such as melody, harmony, rhythm and dynamics. This discussion leads to an acknowledgment of the significance of *sound* in music perception. Next, different methods of analysing, visualizing and synthesizing sound, or more specifically instrument timbre, is shown. Finally, theories of artificial neural networks are outlined, with an example of training a feedforward network with timbre. The success of this simulation is taken as an indication that connectionist models may resemble human perception. Throughout the thesis, several examples are shown of how the graphical programming environment MAX/MSP can be used experimentally in music analysis. The thesis concludes that investigating short term music excerpts might be interesting in music analysis. Due to the limitations of our short term memory, such short passages may reveal noteworthy aspects of music perception. It is also suggested that music theory could benefit from studying salience points and paying more attention to the sound of music.

Contents

Preface

In 1863 Hermann Helmholtz wrote: "In the present work an attempt will be made to connect the boundaries of two sciences [...] physical and physiological acoustics on the one side, and of musical science and aesthetics on the other." (Krumhansl 1995: 53). Helmholtz envisioned a science of music consisting of musical acoustics, auditory physiology, perception and music theory. Never has his words been more appropriate than today. With the advent of fast computers and new computational tools, it is possible to easily work with music in the auditory domain. That is exactly what I have tried to do in this thesis. By approaching music theory from a perceptual point of view, I have looked at how we can relate physical sound waves to traditional musical parameters.

This project, in the field of music technology, has been an "experimental" work rather than theoretical. I have spent a great amount of time learning various computer programs, tools and techniques, such as musical programming languages (Csound, Common Lisp Music), graphical environments (MAX/MSP, jMAX, PD, Squeak) and various Matlab toolboxes (Signal Processing Toolbox, IPEM-toolbox, SOM-toolbox). Of all these, MAX/MSP emerged as my favourite tool and a number of small programs created in this environment will be presented in the thesis[1]. Since I consider these programs an important part of the project I recommend the reader to test out the programs on the accompanying CD-ROM. Compiled programs can be found in the "Applications" folder, and should run on any Macintosh computer with Mac OS 9. Please refer to Appendix 1 or the 'Readme' file on the CD-ROM for more information about this.

The various plots and graphs presented throughout the thesis are all made in Matlab. The source code I wrote for this is presented in Appendix 2.

Sound examples in the text are referred to as "Example #", and can also be found on the CD-ROM. Since there are so many short sound examples, I decided to link them from a HTML-page that can be opened in any web browser[2]. When clicked on, the sound files

---

[1] MAX/MSP is a visual programming environment especially designed for MIDI and audio applications. Originally developed at IRCAM by Miller Puckette, and now commercially available from Cycling'74, it has become a "standard" tool in computer music.

[2] It has been tested with the newer versions of Internet Explorer, Netscape Navigator, and Opera under Windows and Mac OS 9/X.

should open in the preferred audio player installed on the system. The sounds can also be accessed directly from the "Sounds" folder on the CD-ROM.

1   Introduction

This chapter presents the background of the project followed by a specification of the research questions and hypotheses. Definitions of important terms and concepts are discussed, and the choice of a perceptual approach to music theory is presented. Finally, the relevance of the subject to other fields of study is outlined and the structure of the thesis is presented.

1.1   An Idea Comes to Life

Like many other research projects this thesis has become something quite different than the original idea. The basic fascination, however, is still the same, namely that of trying to figure out why music is so powerful and appealing. Second, I have also been interested in how computers could help in organizing, categorizing and finding music.

Starting on this two year music technology program, I thought about doing a statistical investigation of melody. This was based on my reading of (Lerdahl and Jackendoff 1983), (Narmour 1990), (Krumhansl 1995), (Huron 1996) and (Eitan 1997). With a background from both the natural sciences and music, I thought it would be interesting to do a music project on the boundary between the strictly rule-based and the creative. After reading more of the literature and trying various experiments, I came to realize that music is more than "dots on paper". Musical notation is the composer's intention of a piece, the recipe for the creation of music, but the music we actually hear consists of so much more information than what can be found from the score. This led me to understand that doing a statistical analysis of symbolic notation, would miss many of the perceptually relevant things about music.

After reading about perceptual approaches to music theory, I started thinking about the fact that we can recognize familiar songs very quickly. With a large collection of music on my computer, I often start a randomized search function to scroll through the songs looking for something to listen to. Often, I noticed that only a couple of seconds of listening was sufficient to recognize the song. Some short music excerpts (Examples 1a-e) should help to prove this point. All of these examples are approximately 2 seconds long, but still sufficient to recognize the song. Even if the music is not familiar to the listener, the musical information in the excerpt should be sufficient to recognize for example the instruments playing or musical style.

However, this ability of recognition from only a short sensory input is not particular for music. It seems that it is a quite universal feature of human perception, whether the sensory input is for example visual, tactile or auditory. Short sensory information of images, smells, sounds, actions and movements can all lead to recognition in some way. Actually, our whole existence is based on our ability to quickly recognize and discern information from our different "sensors".

With this new awareness, I started to think about this phenomenon whenever I was scrolling through stations on the radio, skipping tracks on a CD or even when walking past open windows with music playing inside. I also noticed that everybody seems to have this ability of quick recognition, independent on the degree of musical training. An example of this is that of Eric Clapton introducing his song *Layla* in concert (Example 1f). He only plays a couple of tones on the guitar, but still the audience recognizes the song in less than three seconds.

Quite fascinating too, is the fact that not only can we recognize a particular song, but also the person playing. Listen to for example Louis Armstrong playing *Summertime* (Example 1g). Most people that know some of his music would probably recognize that this is Armstrong playing his trumpet.

The immediate question might thus be: how do we do it? What is in the sound that makes us recognize a song so quickly? I have always thought that the melody is probably the most important element for our ability to recognize a song. Often, though, it seems that we recognize the song after only hearing a few tones, so there is not much of a melody to speak of yet. In such cases, the harmonic and rhythmic figures might be argued to be important, but one of the most significant factors is probably the *sound* of the music. In the example of Louis Armstrong, one might argue that recognition occurs because the trumpet "sounds" like him.

All of this made me think about how much relevant musical information there must be in such a short excerpt. After all, if we are able to recognize a song, the composer, the instruments and the performers within only a couple of seconds, such an excerpt could be sufficient to categorize the whole song. If we could figure out how to relate what we actually hear to traditional music theory and the physical signal, it could also be possible to make computers "listen to" music in the same way. Even though much effort has been put into research on *music information retrieval* and *automatic music recognition*, there are still no computer models that can even compare to the human brain when it comes to recognizing a song. I think there are a number of reasons for this, many of which will be discussed in this thesis. The most important, however, is probably the lack of satisfactory

tools for doing *computational auditory scene analysis*, or separation and description of separate sound events from complex audio signals. Human perception, on the other hand, works remarkably well dealing with this. Think about for example a jazz quartet playing, and how easily you can recognize the four different instruments. When you can hear each instrument separately, it is also easy to hear what they are playing. The problem with computers is that since they cannot "hear" each instrument separately, they also have problems recognizing what is being played.

This problem of sound segregation might be the reason that much work in computer music and artificial intelligence uses some sort of symbolic notation as the basis for analysis[3]. Interesting is also the fact that even studies that do take the actual sounding music as point of departure, are mainly concerned with doing automatic transcription of the music. The problem with this, of course, is that the highly complex and multi-dimensional original sound is reduced to a limited symbolic structure. The end result is quite the same as the aforementioned, namely a reduction to a statistical analysis of abstracted notes. In this thesis I will therefore advocate a perceptually based approach to music theory, taking the sounding music as the source of analysis.

A very important part of perception and recognition is the mutual cooperation between the various senses. Listening to music in a concert hall involves much more than only auditory perception. Visual information, for example, probably plays a significant role, since seeing musicians perform actions will set up expectations for the sounds that will follow. Even though I think such multi-sensory information is very important, I decided to focus on only the auditory information in this thesis.

I believe that the features of music perception to be treated in this thesis could be called universal[4]. That said, I am sorry that all the sound examples are taken only from the "western" tradition. This is solely because this is the music I know best, and has nothing to do with the quality or importance of other musics. However, I will use examples taken from the classical, jazz and popular music repertoire, and I think this shows some of the width of the ideas. All the examples are chosen entirely from music that I personally find

---

[3]Consider for instance papers from The International Conference on Music and Artificial Intelligence (ICMAI 2002) in Edinburgh, Scotland, and the International Computer Music Conference (ICMC 2002) in Gothenburg, Sweden.

[4] This generally means that it should be applicable to all traditions and cultures. Claiming something to be universal often evokes discussions about the possibility of universality. There will always be people that are deaf, have special diseases etc. that makes such a notion non-appropriate. I still take the chance of using it, but more as a term including *most* people, traditions and cultures.

interesting, and I do not claim to be scientifically objective, even though I try to generalize from my own perception.

An important part of this project has been the exploration of various computational tools, platforms and programming languages related to audio and music. Therefore a lot of the discussions will be focused on various programs I have made. These programs are intended as examples only, and even though they are compiled and do work as stand alone applications, they should be seen as "sketches" rather than final products.

This thesis does not come up with any clear conclusions or answers, but rather presents some thoughts and ideas that can be used in future studies. It is my belief that the best way of approaching such an interdisciplinary subject is by actually getting involved with the literature, tools and techniques from the related disciplines. The project thus resembles a survey of related literature, theories and techniques more than a traditional musicological thesis. I regard the thesis not as a finished product, but merely a start for further research.

1.2   Interdisciplinarity

Officially this is a thesis in music technology, a sub-discipline of musicology. As such, it is not a purely music analytical project, nor a psychological experiment. I do include an example of a simulation of artificial neural networks, but it is not a thesis in computer science, nor is it a psychoacoustical study. If I was to describe which field I think this thesis belongs to, I would probably suggest music cognition, or music theory based on perceptual models and using computational tools. More important than deciding what particular field such a project could be categorized under, is to acknowledge its interdisciplinary approach. That is why I also think it is important to consider some issues related to interdisciplinarity.

Working in an interdisciplinary area always requires caution, since there is always more to read and understand in every possible direction. Writing this thesis has therefore been like the path of a line dancer, trying to stay on the line without falling to one side or the other. Klein (1990: 12-13) presents some topics that might cause confusion and problems in interdisciplinary studies. First of all, there is often an inconsistent use of important terms and concepts. The result is that terms that might have the same underlying meaning, may be used slightly differently in various fields, and lead to confusion and discussion. As an example of this, and as will be discussed in more detail later, I ran into difficulties with the words *timbre* and *texture*. These concepts might be well defined in each of the fields like psychology, computer science and music theory, but when trying to find a common platform I find that they are used quite differently.

Another problem with interdisciplinarity is that of a lack of a professional identity and a unified body of discourse. When dealing with fields of study as divergent as musicology, physics, psychology and computer science, it is quite challenging to have in-depth knowledge of all the fields. Thus simplifications and abstractions may be the result. Although sometimes fruitful, this might also lead to problems of irrelevance, or even miscommunication. Unfortunately, it seems like for example music theorists suggest theories that cannot easily be applied by computer scientists or psychologists. On the other hand, computer scientists often make algorithms and programs for analysis of music that is not very interesting from a musicologist's point of view. Being able to balance between the two is an ideal worth striving after.

As background material for discussions in this thesis, I refer to literature from a number of different subjects. Since I have only a limited knowledge in many of these fields, it is not possible or even desirable to give a full and elaborated treatment of all the topics. A rough selection had to be done in choosing what to include and what to leave out, and it can be argued that many more things should have been added. It has also led me to treat some topics very briefly, but in these cases I have thought that it would be better to do this than leaving them out entirely.

In such an interdisciplinary thesis there are also the problems related to explanations and references. For the most part I have tried to explain important concepts and terms so that it should be understandable for readers coming from different fields. Explaining every concept in detail, however, would not be possible. I therefore assume the reader to have knowledge of music theory and also a basic understanding of music perception and digital signal processing. I have chosen to include some technical discussions of MAX/MSP, since I regard this as an integral part of the programs made for the project. To follow these discussions, however, I assume the reader to have some knowledge of this programming environment. When it comes to the theories of artificial neural networks presented in Chapter 6, this necessarily involves some equations and knowledge of computer programming, but the main ideas have been outlined also in normal language. As such, I think the thesis might serve as an introduction to a number of the related fields of study.

1.3   Main Principles of the Project

The thesis is inspired by the remarkable ability of the human mind to perceive sound and quickly recognize or categorize meaningful content. The underlying observation is:

- <u>Observation</u>: We manage to recognize a song by listening to only a second of it at any point in time.

I will use the term *recognition* in the meaning of recognizing *something*. The notion of "a second" refers to the concept of short term music perception, rather than "one second" (see Section 1.4 for more detailed definitions).

But how is it actually possible to recognize a song in a second? I think this can be seen as analogous to for example reading words on a paper. Firstly, we separate the black spots on the paper as separate characters. Then we group characters together to words, and words to sentences, and finally we can understand the content. The same applies to music, where separate sound waves are grouped to tones, and tones are grouped to music. So the recognition process can be broken down to two parts:

- *Segregation* of sensory input
- *Recognition* of musical content

Theories related to auditory segregation will be presented in Chapter 3, but the rest of the thesis will concentrate more on musical features that are important for our recognition. When it comes to the recognition process, the first step is to verify that it is actually possible to recognize music from a short excerpt, and to figure out how much time we actually need to recognize a song:

- <u>Question 1</u>: How much time do we need to recognize a song?

Next is to find out how we actually do it, or rather to figure out what musical content we recognize. I pose two hypotheses about this:

- <u>Hypothesis 1</u>: We recognize music faster when there is some *salient* feature present.

- <u>Hypothesis 2</u>: The *sound* plays a significant role in our music perception.

By *salient* feature, I mean a subjectively significant element. The concept of *sound* is used to denote the quality of the overall features we hear. This will be outlined in more detail in Section 1.4.

It is also necessary to discuss whether the abovementioned ideas can be used to suggest general principles of music perception:

▪ Question 2: Is it possible to use knowledge about short term music recognition as a basis for a general model of music perception?

The ideas up until this point could be regarded as the first part of the thesis, rooted in music theory and perception. The second part will follow the suggestion that *sound* is important for music perception. I think it is important to try and relate our perception of music to the physical signal, and also to the concepts of traditional music theory. To make computers able to "listen" like humans, we need to formulate music theoretical ideas in such a way that it is actually possible to make computational models based on it. Or put in another way, we want to understand more about our music perception and the subjective experience of music listening, and be able to explain it within both a music theoretical and a computational framework. Obviously, this thesis will not be able to solve all these problems, but my contribution is an attempt to discuss some related issues by focusing on *sound*:

▪ Question 3: How is it possible to visualize and analyse *sound*?

The focus on *sound* is inspired by the fact that this seems to be a topic that has not received so much attention in music theory. One reason for this might be the lack of good analytical tools. Today, the advent of new digital signal processing techniques and faster computers makes it possible to study music from audio in ways that would not have been possible only some years ago.

To summarize the project, Figure 1 shows a sketch of how I think the main topics are connected. First, music recognition can be subdivided into segregation and recognition processes. I further believe that the concept of salience is significant for the recognition process, and it can be explained by reference to the musical parameters. One of these parameters is timbre, an important constituent to the sound of music. As shown in Figure 1, I think that sound, and more specifically timbre, is significant both in the segregation and recognition processes. The most important, however, is to realize the complexity, multidimensionality and interconnectedness of all the various topics. This calls for alternative ways of approaching the problem, for example by investigating computer models based on biological systems.
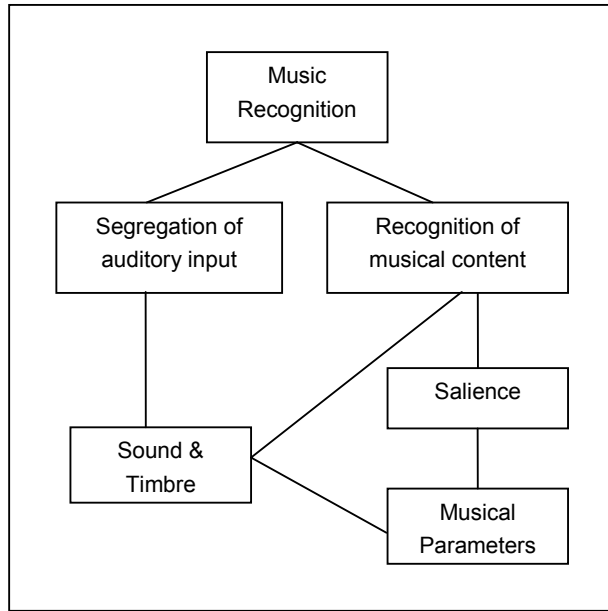
Figure 1. Overview of some elements of music recognition as they will be discussed in this thesis. I believe that all of these interact in some way, and that sound *might be a key topic since it is important in our perception of salience, as a musical parameter and for auditory segregation.*

1.4 Definitions

The term "sound" is used in daily life to denote the waveforms that are sensed through the hearing system. In this thesis, however, I will use *sound* as a description of the quality of the overall features of what we can hear. This is in accordance with how the term is used in jazz and pop studies as denoting overall timbral qualities of a musician or a band. The concept of sound could be argued to be very close to the concepts of timbre and texture, but I believe that there are some important differences and also some inconsistencies in the use of these terms.

The concept of *timbre* is not easy to define. The 1960 definition by the American National Standards Institute (ANSI) states that "Timbre is that attribute of auditory sensation in terms of which a listener can judge that two sounds similarly presented and having the same loudness and pitch are dissimilar." (Risset and Wessel 1999: 113). The 1973 definition reads that timbre is "everything that is not loudness, pitch, or spatial perception…" (Houtsma 1997: 105). Such negative definitions are difficult to use when looking for specific qualities in a signal. If we look to musicology, there seems to be some inconsistency in the use of timbre as opposed to the German *Klangfarbe* and *Klang*[5]. While

---

[5] Rooted in a German tradition, the same concepts are also widely used in the Scandinavian countries.

the former could be translated to "tone colour" in English, it could be argued that the latter is often used in a sense of what I call sound. The concepts of timbre, tone colour and Klang therefore seems to be used somehow differently in various studies. Sometimes they are used for referring to qualities we can hear, other times as tools for describing physical qualities in frequency spectra. They are even used as referential concepts within technical discussions of orchestration.

Many of the problems associated with timbre also apply to *texture*. In musicology this concept is often used to describe harmonic spacing and chord relationships, or "the distribution of sonic events in time and spectrum" (Godøy 1999: 69). In psychology and computer science, on the other hand, it is often used to "describe statistical characteristics of the spectral distribution" (Tzanetakis 2002: 38).

I see that there are both similar and different connotations of the concepts of sound, timbre and texture, and it would be very interesting to see a longer discussion of these concepts with a clarification of the meanings in different fields. That, however, is not the scope of this thesis, and since the focus will be on the actual sound that we hear, I will use the term *sound* when talking about the sound quality of a section as a whole. *Timbre* will be used to denote the multi-dimensional sound quality of a single instrument, or an instrument group that is perceived as a coherent entity. Thus the *sound* in an excerpt of Wagner's *Tristan* (Example 1h) includes all the timbral qualities of each instrument as well as any sound effects, reverb etc.

I prefer to make a distinction between the concepts of note and tone. While *note* refers to our understanding of a certain pitch, the *tone* is a note with a related timbre. For example, it should be quite clear that there are sonic qualities in the *tones* we hear in Example 1i that are not apparent when reading the same *notes* (D-E-F#) in a score. This fact is important for my choice of a perceptual approach to music theory.

The notion of *short term music perception* will be used, and by this I mean events of duration up to about 3-5 seconds. This is linked to the general acceptance that our short term memory lasts for such an interval of time (Snyder 2000: 47).

The concept of *perception* has traditionally been used to denote the processes involved in our hearing system dependent on immediate sensory input. *Cognition*, on the other hand, has been used to describe what is going on in the higher levels of the brain. Nowadays, it is agreed that all perception involves cognition, and that there is no such thing as "neutral" perception (Godøy 1999). Furthermore, perception is often used in music, and also everyday life, in a more philosophical sense; as the contact between musical sound and our

mind (Aksnes 2002: 285). Since this thesis will not deal with higher level mental processes, I will leave cognition out, and use perception both in relation to our hearing system and more "high level" extraction of musical content from what we hear.

I use the word music *recognition* in the meaning of recognizing *something*. Recognition is based upon mental processes, and is therefore individually and culturally dependent. What we perceive and recognize is based on our former experience, so even though we hear the same song, the experience of it might be very different. Some people might know the song title, composer and performers, while others only recognize that it belongs to a certain musical style. For example, classical fans might "close" their ears when they happen to listen to pop music, and without further thinking conclude that it is definitely not classical music. Another example is that of jazz, a type of music that it is often necessary to "learn to like". The beginner might recognize the tune being played, while the expert listener enjoys differences in solos and knows the whole story about the recording session. For this study, however, I have not been interested in figuring out specifically *what* we recognize from a musical example, but rather to investigate the musical features in the excerpt that makes this recognition possible. Said in another way, we are interested in analysing the input signal and find out what musical parameters are important for our perception of that signal. I will therefore continue to use the word *recognition* in the meaning of recognizing *something*, whether it be timbral qualities, the tune, or the musicians playing.

I suggest the term *musical point* to denote our perception of music at a certain "point" in time. When listening to music we might notice something that happens "now". Since this can be recalled at a later time, we obviously have a quite well-defined notion of this subjective "now". The musical point can be related to the perceptual "now" that Husserl suggested with his time line model presented in Figure 2 (Schneider and Godøy 2001: 13).
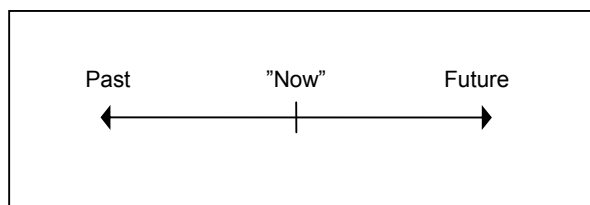
Figure 2. Husserl's Zeitstrecke. In the perceptual "now" there will always be a connection to the past and an expectation of the future (Schneider and Godøy 2001: 13).

The idea is that the perceptual "now" will be intermittently changing and will always be related to what is in our memory (the past) and our expectations for what is coming (the future). The problem is what that "now" actually is, when it is, and how we can relate it to a physical sound signal. Since music unfolds in time, it is of no interest to analyse a point

of an auditory wave as one byte of information. That one byte would not be audible in itself, and it would probably not be very representative for the other "bytes" surrounding it. Therefore a musical point in time must be of a certain length, for example one second. Since the musical point is a subjective entity, it is very much dependent on the musical content and context. Therefore its exact duration will have to be evaluated on the basis of what musical parameters are at work, and the style of the music. For example, 300 milliseconds of a song might be sufficient to give a sense of pitch, while perception of timbral qualities of instruments might necessitate an excerpt lasting 1500 milliseconds. The musical point will therefore be able to help us refer to a specific musical feature occurring in time. This is important since it enables a discussion of music using traditional musical concepts, but with audio as its source. This way the musical point will help in referring to specific musical features in much the same way as when referring to notes in a score.

Another concept that will be used throughout the thesis is *salience.* Salience comes from the Latin word salire that means to leap (*Webster's Revised Unabridged Dictionary* 1913). The word is used with many connotations in different subjects, but generally it is related to something prominent or significant. Mathematical studies use salience as a term to describe a sudden change in the derivative of a function, as for example in (Large and Palmer 2002). When it comes to its use in relation to features or events in music, salience has been used in studies of pulse and rhythm perception. In a study of musical performance, Parncutt (1987) found that the tempo slows down near *salient events*. The performer knows that something "important" is coming, and uses a deceleration in tempo to further enhance the salient point. This is quite often also notated in the score of classical music as ritardando. A somewhat different connotation of salience is suggested in (Rothgeb 1997), where it is argued that we are often mislead to believe that the most apparent features are salient, but they are not really so important for the projection of the musical message. He further argues that "tonal shapes", or large-scale structures that are not immediately apparent, are the "true" salient features. I prefer to stick with a definition of salience as something that is immediately apparent and stands out in our consciousness (Crowther 1995). However, it is important to remember that salience is dependent on context, for example such as Patricia Hanna (2001) points out that something being psychologically salient is not necessarily linguistically salient. When it comes to musical salience, it is probably determined by a number of factors, such as dynamics, pitch, harmony and cadences (Dixon and Cambouropoulos 2000).

By musical parameters I refer to traditional music theory concepts such as melody, harmony, rhythm, dynamics, tempo and timbre (Meyer 1989). Melody denotes placement of notes in succession, evolving over time, while harmony refers to simultaneous placement of notes. The other parameters govern internal and external relationships

between notes. What is important to remember when we are working with a perceptual approach, and in the auditory domain, is that the musical parameters are in no way simple and well-defined features. Their highly complex and multi-dimensional nature makes it difficult to separate them. Refer to Section 4.6 for a more detailed discussion on the musical parameters and their physical representations.

1.5 A Perceptual Approach

This thesis is based on our perception of music, and the sounding music is used as the source of analysis. This stands in contrast to traditional analysis of western tonal music, where a musical score is the reference. Although music notation is well-defined and highly versatile, there are a number of reasons why I believe it is more interesting to use a non-symbolic, continuous stream of sound as the source of analysis[6].

First of all is the problem that musical notation often tells little about the actual sounding music. The musical score, as a symbolic system, could be seen as a recipe for how to perform music. As such it is the composer's description of how actions should be performed on specific instruments. So the musical score in itself is not music, it is first when a skilled musician interprets the notes in the score and plays it, that the music comes to life. Thus the qualities of the performer will always be essential for our perception of the music. For example, most people will agree that listening to a computer playing Beethoven's 5th symphony is not the same as listening to the New York Philharmonic. The interpretation of a piece, how to phrase melodies and to emphasize certain elements, and to shape the *sound* of the music, is decided by the musician. Some people might argue that they can read a score and "hear" the music. This might very well be the case, but then they themselves serve as the interpreter of the notated material. Furthermore, they draw upon their experience of timbral qualities. How can you imagine the sound of an oboe if you have never heard one? This way of reading music is still dependent on the perceptual qualities of instruments. So there is necessarily much musical content that is not, and could never be, notated in a score. This is essential for the choice of using a perceptual approach to music, since analysing the sounding music will reveal what the score cannot.

The second important argument for a perceptual point of view, is that only a small part of the world's music is based on traditional, western notation. Folk, popular, jazz and

---

[6] As suggested by (Leman 1995: 182), music is a non-symbolic, or sub-symbolic, system based on continuity. When referring to a continuous signal, this should not to be confused with the fact that the digital domain on computers is actually discrete. However, because of a high sampling rate the audio from such a discrete source will still be perceived as a continuous stream.

contemporary music do often not use any notation at all, or only use very brief sketches. Folk music, for example, is often passed on from one musician to the next and is usually never written down. Jazz and pop songs, on the other hand, are often only notated with the melody and chord changes, leaving most of the interpretation to the performer. The whole idea of jazz is about making new harmonic changes to standard tunes, improvisation and addition of new ideas and variations. Today's pop music is often entirely based on sound, with not much characteristic melodic or harmonic content to speak of.

Contemporary music has often gone beyond that of traditional notation, because of the limitations it forces upon the creation. There is for example no way of adequately describing timbre changes for instruments. Since quite a lot of music has not been notated before it was played, a great deal of effort has been put into trying to transcribe it afterwards. The result is that the attempt of transcription often cuts away most of what is important in the music. So a good deal of the musical content is lost, just because we try to force it into a symbolic system that is ill-suited to represent it. Clearly, it would be much better to develop techniques to analyse the music without having to go through such a process. Using a perceptual approach to the sounding music might thus be a good solution.

Another problem with traditional analysis is that many perceptually irrelevant conclusions might be drawn. If we do music analysis without listening, we might end up finding structures and impose meanings that are not even perceivable. While this might be interesting in some cases, it can also distract us from finding the perceptually relevant features. A reason for this could be that the traditional concepts of tones, scales and chords rarely occur in the sounding music, as they are by-products of music notational analysis (Serafine 1988). Serafine also claims that the whole hierarchy of a theoretically based, notational system can actually be more of a distraction than help. Elaborating on this, Fiske (1996) argues that the musical parameters, and elements of music theory, are only analytic concepts, made to help us understand vertical and horizontal development in music. When such an apparatus is used as the basis for analysis, it is clear that the findings might be far removed from what we actually hear.

This leads us to another problem, namely that of what musical features we analyse. Melody and harmony are represented in the score in horizontal and vertical dimensions, while the rhythm can be found from the value of the notes. With such a material, it is not surprising that much traditional analysis has focused on the importance of melody, harmony and rhythm. Bent (1980) outlines this in the New Grove article on analysis and points out that of these three, melody must be seen as the single most important element in the historical, analytical literature. This fits well with what Helmholtz writes in the preface to his *On the Sensations of Tone*: "The essential basis of Music is Melody" (Helmholtz 1885/1954).

14

There can be no doubt that melody, harmony and rhythm are important in music, but when they are so easily available in the score, we tend to forget about for example the importance of timbral qualities. Furthermore, it is a fact that people's ability to recognize intervals vary considerably. Shepard (Levitin 2000) found that as much as half of the population might have problems in discerning semitones from each other. If we assume this was the case, how is it then possible that virtually everybody recognizes their favourite song when played on the radio? Clearly, there must be more to a song than only changes in intervals, since even people claiming themselves to be "tone deaf", can still enjoy and recognize music.

With the advent of computers many musicologists started using computational tools in their analysis. Large databases such as the Essen Collection provide easy access to notated music in a standard format[7]. This makes it relatively easy to develop tools for doing for example statistical analysis. Examples of this are Huron's investigation of melodic arches (Huron 1996), Eitan's discussion of melodic peaks (Eitan 1997) and Toivianen's use of connectionist models in folk music analysis (Toiviainen and Eerola 2001). All of these present interesting findings, but the problem is the lack of *listening* to the music. In a response to such an approach, Leman (1995) suggests the use of perceptual models using the auditory signal as the basis. He argues that a change from note semantics, i.e. the relationships between notes, to music semantics, i.e. dealing with music as a constantly changing phenomenon, would be fruitful.

The ideas of Leman are also tightly connected to our mental imagery of music. How we think about music is probably also related to how we talk about music. A music theorist could describe a piece in terms of instruments and functions, such as the wind instruments overlapping from the strings, a modulation and a full cadence. Someone not so familiar with music theory would probably use language metaphors in their explanation of the music. Quite often it is difficult to find words adequately describing the music, and then it is easy to use adjectives like "massive and pompous" or "warm and light". Sometimes it is easier to refer to sounds from real life, such as birds singing or the sound of waves against a shore. It is interesting that when talking about music we tend to use such "images" rather than discrete note events. This should be taken as an indication that our thinking about music follows the same lines. Our reference to pictures, colours or moods, is also a reminder about the complexity and multi-dimensionality of music. Godøy (1999) suggests that we also relate visual cues to our perception of music. That is, we tend to remember sounds through the action that caused them. Sitting at a concert watching the musicians

---

[7] A large database collection consisting of more than 6000 German folk songs notated in the *HumDrum* ***kern* format. Stored and maintained by CCARH, Stanford University.

play, there is a clear correspondence between what we see and what we hear. Seeing a drum stick fall towards the drum causes an immediate expectation that a drum sound will occur. Just think about how annoying it is with asynchrony between image and sound on TV.

Finally, I believe the most important reason for using perceptual models in our understanding of music is the importance of *sound*. As mentioned above, timbral qualities are not well represented in musical notation and are also difficult to define precisely. Although the last decades have seen a promising amount of work on the importance of timbre, such as (Grey 1977), (Wessel 1979), (Krumhansl 1989) and (McAdams et al. 1995), it seems that the approach has been more from a psychological than a musicological point of view. It also seems to have been more effort put into the creation of new sounds than into developing new methods for analysis. A reason for this might be that it has proven to be very difficult to define timbre precisely and to point out its specific role in a musical context.

Even though I favour a perceptual approach, I do believe that traditional methods can also reveal interesting things about a musical piece. For a trained reader a musical score might be a good description of the music. It would also be unwise to totally abandon the many interesting methods and analytical techniques that are used in connection to traditional notation. The ideal strategy is probably to combine the better of the two different worlds. Using our perception of music as the basis for study, we can use the score to find and demonstrate things that would otherwise be difficult to represent.

1.6   Relevance to other Fields of Study

I believe that the topics covered in this thesis are relevant to a number of adjacent subjects:

- *Music information retrieval:* making new and better computational methods for organizing large sound databases and automatically retrieve musical information. This may be interesting for broadcasting companies and music libraries that need quick access to large collections of music. For most people it could be helpful for doing query-by-example or query-by-humming on their computer or on the internet.
- *Sound synthesis:* better tools for music analysis will most certainly improve the possibilities for the creation of sound. This is interesting for composers and sound engineers.
- *Music analysis:* providing a psychophysical understanding of musical parameters.

- *Music education:* automatic music recognition can help provide accompaniment for students while practising.
- *Music performance:* better instruments for live electronic concerts.
- *Note transcription:* let a computer do note transcriptions of music.
- *Sound engineers:* score-following, editing and improvements in sound quality.

Needless to say, these are but a few of a number of highly complex domains that might be related to this project.

1.7   Structure of the Thesis

Given all the research topics that this project is related to, it goes without saying that I can only briefly discuss some of them here. It could now be useful to present an overview of how I have chosen to organize this thesis.

The first chapter has tried to introduce the background of the project, the research questions and hypotheses. Some relevant concepts have been defined and the choice of a perceptual approach has been discussed. In Chapter 2, I will present the concept of musical sound, and discuss what a tone is. After an introduction to the graphical programming environment MAX/MSP, an example of synthesis of a complex tone will be described. Then Chapter 3 will focus on the concept of auditory scene analysis, giving a brief survey of theories of memory, our hearing system and grouping. In Chapter 4 music recognition will be discussed, and an experiment measuring recognition time will be outlined. Analysis of some musical examples leads to a discussion of salience and how each of the musical parameters can be perceived as salient. This is elaborated with more examples taken from both classical and popular music. Finally, salience is discussed in a broader context of music perception, and examples of how salience points can be used in making "musical trailers" are presented.

The abovementioned chapters can be seen as the first part of the thesis, with an overview of topics and discussions of music recognition from a more general point of view. The last chapters (starting with Chapter 5) will focus on sound, or more specifically timbre, and how this can be analysed, visualized and synthesized. This leads to Chapter 6 that starts with a brief discussion on connectionist versus rule based systems, before theories of artificial neural networks are presented. This is exemplified with a simulation of training a neural network with saxophone sounds. Finally, Chapter 7 summarizes the preceding chapters and presents some thoughts about future work.

2 Musical Sound, Synthesis and Visualization

This chapter starts with a discussion of musical sound. After an introduction to MAX/MSP an example of synthesis of a complex tone is described and different types of timbre are discussed. Finally, two methods for visualizing audio are presented.

2.1 The Soundscape and Musical Sound

When studying music from an auditory perspective, it is necessary to clarify what we are actually investigating. Schafer (1977: 274) defines our sonic environment as a s*oundscape*[8]. This can be an actual environment, for example in the city or out in nature, where all the sounds we hear belong to the soundscape, whether it be people talking, birds singing, or cars passing by. But the concept of a soundscape can also mean an abstract construction such as a musical composition. In such a case, the musical piece should be seen as a whole, spanning all instruments and their qualities, loudness and position.

I prefer to use the term *musical sound* to describe the sound in a soundscape that contributes to the music. To clarify this, I suggest looking at a typical concert situation, where it is possible to roughly categorize the soundscape into three groups, dependent on the sound source:

- Sounds from the musicians and their instruments (music they make, noise they make)
- Sounds from the audience (sounds of moving chairs, coughing, whispering, etc.)
- Sounds from the environment (reverb, ventilation, traffic outside, etc.)

There is no easy answer to which of these should be considered musical sound. Let us start by first looking at sounds from musicians and their instruments. They can probably be subdivided into the following groups:

- Musical sounds from the instrument (sounds made with the intention of belonging to the composition)
- Technical sounds from the instrument (e.g. noise from the pedal on a piano, keys on a flute)

---

[8] A project called Soundscape was initiated by Schafer, and included extensive sonic research in several cities throughout the world. The findings are summarized in (Schafer 1977).

- Musical sounds from the musician (singing, humming, whistling, etc.)
- Body sounds from the musician (breathing, fingers sliding, etc.)

Traditional music analysis is usually concerned with the first group of sounds, i.e. the musical sounds from the instruments. For a typical classical piece, these sounds would correspond to the musical idea notated in the score. We may therefore safely assume that this is also the most important contributor to the musical sound. After all, we usually listen to music because we are interested in the composition. However, I think it is also important to recognize the significance of the other sounds.

In the case of technical sounds from an instrument, it is not easy to say whether they should be considered part of the musical sound or not. This is because in some cases they might not be audible at all, while in others they probably are very important for the timbre of the instrument. From the far back of a concert hall one will not be likely to hear mechanical sounds from the piano pedal, but up front it may even be possible to hear the moving keys on a flute. What about the sounds of fingers sliding on a guitar or violin strings? Such sounds are probably not notated in the score, but appear because of choices made by the musician. Should these examples be considered musical sounds? My best answer is that if we hear sounds related to musical creation they do form a part of the performance. This is the case even though the creation of such sounds might not be intentional or controlled by the musician. However, intentional or not intentional, they also contribute to the overall sound we hear.

When it comes to the musical sounds from the musician, these may be liked or disliked. Keith Jarrett, for example, is renowned for humming while playing his piano solos. While some might not like this, Carr (1991) claims that this is an important part of Jarrett's performance. The singing is a means of helping him indulge in the music and is an integral part of the solo. Carr argues that if Jarrett could not sing, he would have to restrain himself, resulting in poorer piano music. This way, the singing could be seen as unintentional body sounds, much the same as the sounds of nails on keys or heavy breathing. Such sounds are not controlled by the musician, but are rather consequences of the musician's effort, and they do form part of the soundscape in a performance.

In a performance situation it is not easy to control what sounds are audible, but on recordings the policy usually seems to have been much stricter. It seems that sound engineers in general have been quite conservative, carefully removing any extra-musical sounds from instruments or musicians. But there are also examples of musicians, especially in contemporary jazz, using amplifying equipment to enhance body and instrument sounds, and thereby making it part of the actual musical content.

When it comes to the sounds from the audience, it is interesting to see an ambiguous treatment. On one side, it seems like all sounds from the audience should be kept as low as possible while the music is playing. On the other side, everybody wants the applause to be as loud as possible. Small (1998) reflects on how the audience, at a typical classical concert, are following unwritten rules about how to behave during a performance. While the musicians are playing, everybody should be absolutely quiet. Between the movements, when there is not supposed to be applause, many people starts coughing and turning pages in their programs. Finally, when the piece is finished, everybody is supposed to start clapping. This is quite opposite to the behaviour that is expected in a typical jazz concert, where the audience is supposed to applaud after every solo, or in a rock concert where the audience is clapping, shouting and screaming throughout the whole concert. In short, the different genres have their own rules for how you are supposed to behave as an audience. This way, it can be argued that the sounds of the audience are indeed an important part of music perception.

Finally, there are the sounds made by other sources than musicians or audience. Some of these, for example reverb in the room might be intentional and very important for our perception of the timbre of an instrument. In many cases, reverb is used as an important parameter of music. Other sounds from the environment, for example sounds of ventilation systems and traffic outside, could be considered noise, because they are not intentional. But if they are audible, they also form an integral part of the music experience. As such, it could also be argued that their presence changes the overall content of the sound. Such environment sounds thus both contribute in themselves, but they may also have a "sound colouration" effect on the rest of the sounds.

As this discussion has shown, defining what should be considered musical sounds in a soundscape is not easy. I believe that a definition of musical sound that only takes the actual music into consideration violates the importance of our perceptual experience. In this thesis, I will therefore consider all components that are audible in a musical context as essential for our perception of the music. If we listen to an old recording of Louis Armstrong, it is impossible not to take into account the bad sound quality. Of course, this has nothing to do with the quality of his playing, but it is probably a significant aspect of our recognition of his music.

After this discussion of musical sound, it is important to have a clear notion of what a sound is in physical terms. The following sections will therefore focus on the creation of a complex tone, and how it can be visualized. This however, requires some basic knowledge of MAX/MSP.

2.2   Introduction to MAX/MSP

This section will give a very short introduction to MAX/MSP, so that readers not familiar with this software can follow the examples presented throughout the thesis.

The intention of MAX was to create a graphical programming environment for musicians and composers[9] (Puckette 1985). Originally developed at IRCAM by David Zicarelli and Miller Puckette in the late 1980's, it soon became popular for controlling MIDI-instruments (Puckette 1988). Its unique flexibility, and the possibility for users to extend the capabilities of the environment by writing new code, secured its position in computer music (Puckette and Zicarelli 1990). The novel idea was the creation of a graphical environment that could be run in real-time, allowing the user to interact with the program. The MSP-package (released in 1996) added audio capabilities, and Jitter (released in 2002) allows the manipulation of video. Today MAX/MSP/Jitter are commercially available products, continuing to attract a large community of users[10].

Since MAX/MSP was created with its main focus on music creation and manipulation, it is also suitable for music analysis. This is due to the large palette of tools that are not easily available in other programming environments. Since I also find it so much more intuitive to use than traditional programming, I decided to use it for my experiments[11]. I also think it should be possible for readers unfamiliar with MAX/MSP to understand some of the structure in the examples that will be provided in this thesis.

A program in MAX/MSP is usually called a *patch*, and looks like a window on the screen (Figure 3). A patch can be seen in two modes, either *edit* or *run*. In edit mode, the user can build up the patch by adding *objects*, the building blocks of the environment. *Externals* can be either C-programs that are compiled to MAX-objects, or other patches. Thus it is easy to add extra functionality by writing new externals. Objects are connected to each other with *patch cords* (Figure 4-1), through the *inlets* and *outlets* of the objects. By following the patch cords it is therefore possible to see how the patch will behave, i.e. the flow of data.

---

[9] MAX is named after the "father" of computer music, Max Mathews.

[10] The software is available from Cycling '74 (www.cycling74.com), which also runs a community forum and active mailing-lists.

[11] After struggling with traditional programming languages like C, java and Perl, I found MAX/MSP refreshingly intuitive. Of course, it can be argued that it does not perform as well as the more "hard-core" languages, but for my experiments it proved to be well-suited.
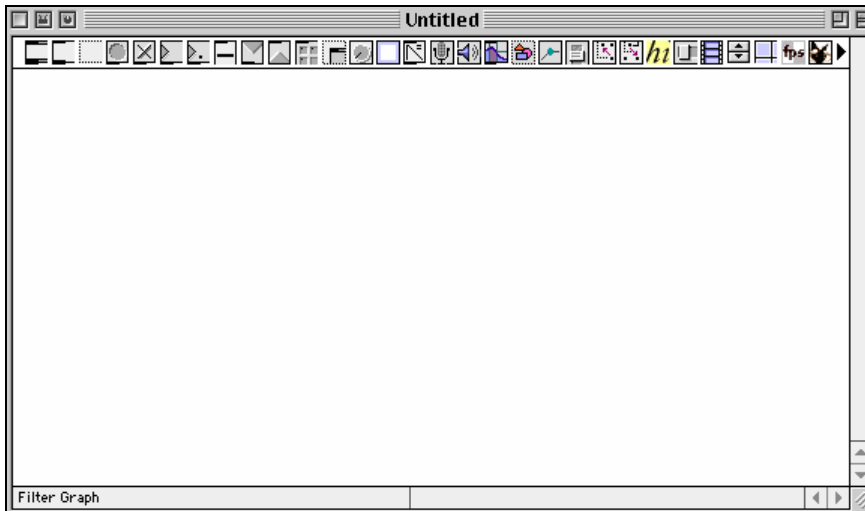
Figure 3. An empty MAX/MSP patch window. A menu on the top provides links to the most common objects.

Switching to the *run* mode makes it possible to actually use the patch (Figure 4-2). Changing values in number boxes, for example, will cause operations to be done (Figure 4-3). Usually it is the left inlet that triggers an operation, while the right inlet store values. Thus, in this example the number 2 received in the right inlet will be stored in the object. It is first when a number is set in the left inlet that the "+" object triggers an output.
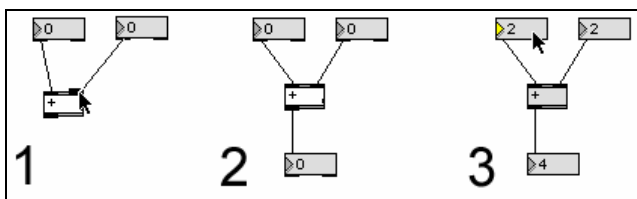
Figure 4. Objects are connected with patch cords in MAX/MSP (1). When all connections are made, the patch can be switched to run mode (2), and the user can change values and see the results directly (3), in this case performing an addition of the two upper numbers.

The example in Figure 4 is very simple, but it shows the basics of MAX. There are many other objects, and they can all be connected in different ways. Most of the objects also have quite logical names, so it is easy to get started making patches. Users with traditional programming experience will appreciate that it is actually possible to use objects such as *if-else*, *for* and *while* also in MAX, even though similar operations can also be accomplished in many other ways. There are always many ways of doing the same data processing in MAX!

Worth mentioning is also the difference between MAX and MSP objects. While MAX is only concerned with mathematical functions and MIDI messages, MSP is the collection of tools that control the audio part of the environment. It is easy to spot MSP objects in a patch since they always have the ~ (tilde) extension. Another difference is that the patch cords connecting MSP objects are thicker and have a dotted line, making it easy to see where audio signals are going. This separation of signal and control routes is logical, but often confusing when starting to learn MAX/MSP. There is for example an important difference between the objects '*+*' and '*+~*'. The former is used in addition of numbers, while the latter is used when adding sound signals. Luckily, MAX will not allow you to connect for example a signal object to a number object.

There are lots of other things to be said about MAX/MSP, but I think the most important is to understand the concept that a patch is built by objects connected together with patch cords. So even in quite complicated patches, it is possible to follow patch cords and understand the programming structure. In the patches presented in this thesis I will not explain how each object works, but I still hope that non-experienced readers can understand the main idea behind the patches.

2.3   Synthesis of a Tone

The sound we hear is built up of sound waves that can be described physically in terms of frequencies. A single frequency tone is called a *sine* tone, because it has the shape of a sinusoidal function. Such tones are not produced by traditional instruments and can only be made artificially. All natural sounds have a much more complex physical waveform, built up by a set of sinusoidal frequencies. Each sinusoidal component that is part of such a tone is called a *partial* frequency. A tone where all the partials are multiples of the fundamental frequency is said to have a *harmonic spectrum*. A partial of a harmonic spectrum is often called a *harmonic*. The fundamental frequency (often referred to as F0) is what we usually perceive as the pitch of the tone, while the other frequencies making up the waveform contribute to the timbre.

The program *Harmonics* is a MAX/MSP patch that allows the user to "construct" a tone [12]. As can be seen in the screenshot of the patch (Figure 5) there are various buttons, sliders and number dials. Together they allow for adjusting the pitch of the tone and the separate amplitudes of up to 60 of its harmonics. The tone is played with a simple ADSR-

---

[12] This and all the other patches and programs referred to in the thesis can be found on the accompanying CD-ROM.

envelope[13] by clicking on the button. The program can be found on the CD-ROM, and I encourage the reader to test its features! In the rest of this section I will briefly go through some technical aspects of the patch.
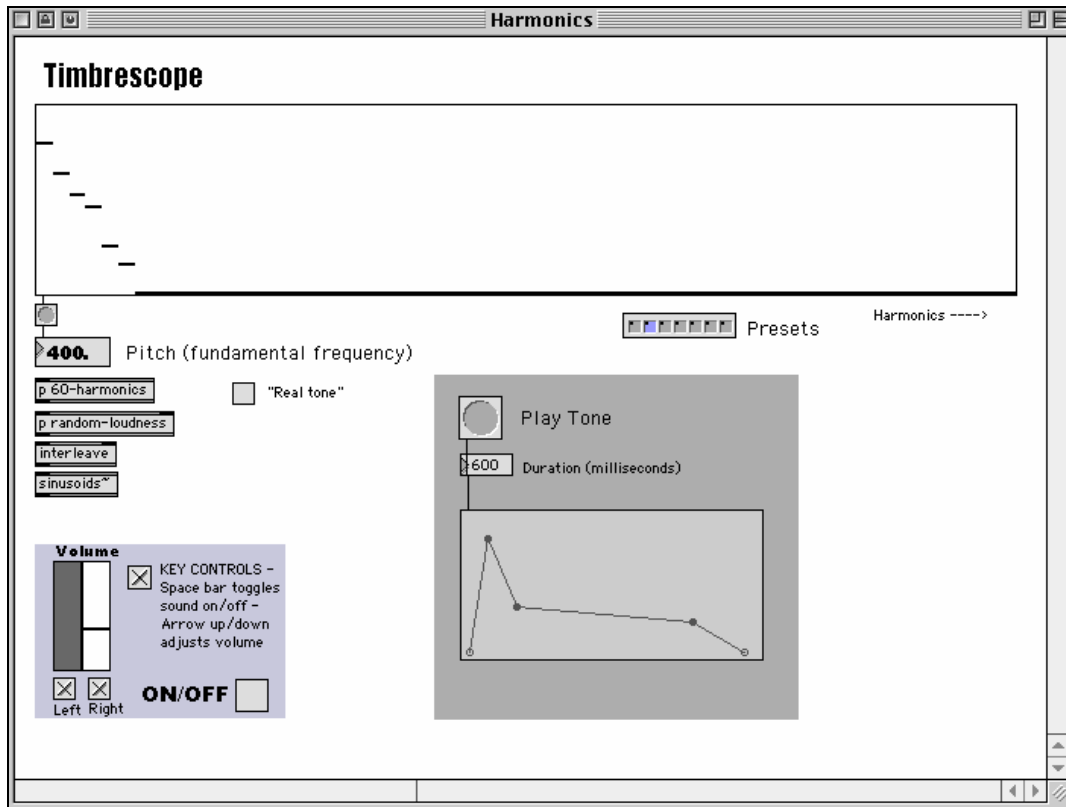
Figure 5. Screenshot from Harmonics*, a MAX/MSP patch where the user can play tones with different number of harmonics.*

In the top part of the patch window (Figure 5) is a *multislider* object with 60 independent sliders. These sliders will change the amplitude of each of the 60 harmonics of the sound[14]. The user is free to adjust all the sliders manually. Changing only the first slider, leaving the others at 0, will result in a sine tone. The pitch of the tone can be adjusted by changing the value of the fundamental frequency.

Since this patch creates harmonic tones, each of the harmonic frequencies can be found by simple multiplying the value of the fundamental frequency. This is done in a subpatch (Figure 6) taking the fundamental frequency as input. The output of the subpatch is a list of the 60 harmonic frequencies of the tone.

---

[13] Attack-Decay-Sustain-Release.

[14] Having 60 harmonics seems to be sufficient for adequately reproducing music. This will be discussed in more detail in Section 5.3.

24

Figure 6. Screenshot from the packing of 60 harmonics.

The *additive synthesis*, the creation of a complex sound wave from its sinusoidal components, is easily done with the *sinusoids~* object[15]. This object takes a list of up to 256 consecutive pairs of frequency and amplitude as input. Doing additive synthesis the "old" way, would require building up a set of for example *cycle~* objects. Not only would this require a lot of work, but would also be computationally demanding for 60 harmonics. So the *sinusoids~* object saves a lot of hassle.

The problem is how to get the two separate lists of harmonic frequencies and the loudness values merged together, before sending it to the *sinusoids~* object. This was solved with the *interleave* object[16], that simply merges two input lists to a single output list. The result is sent to *sinusoids~* and the sound is "created".

Playing this would result in a very "static" and unnatural sound since the volume is constant and the harmonics are not changing. As can be seen in Figure 7, this was solved by adding a simple ADSR-envelope, controlling the volume of the output sound through a *line~* object. The duration of the tone can be adjusted accordingly. Simply by adding such an envelope to the sound, makes it much more realistic.

---

[15] By Adrian Freed at CNMAT, UC Berkeley.
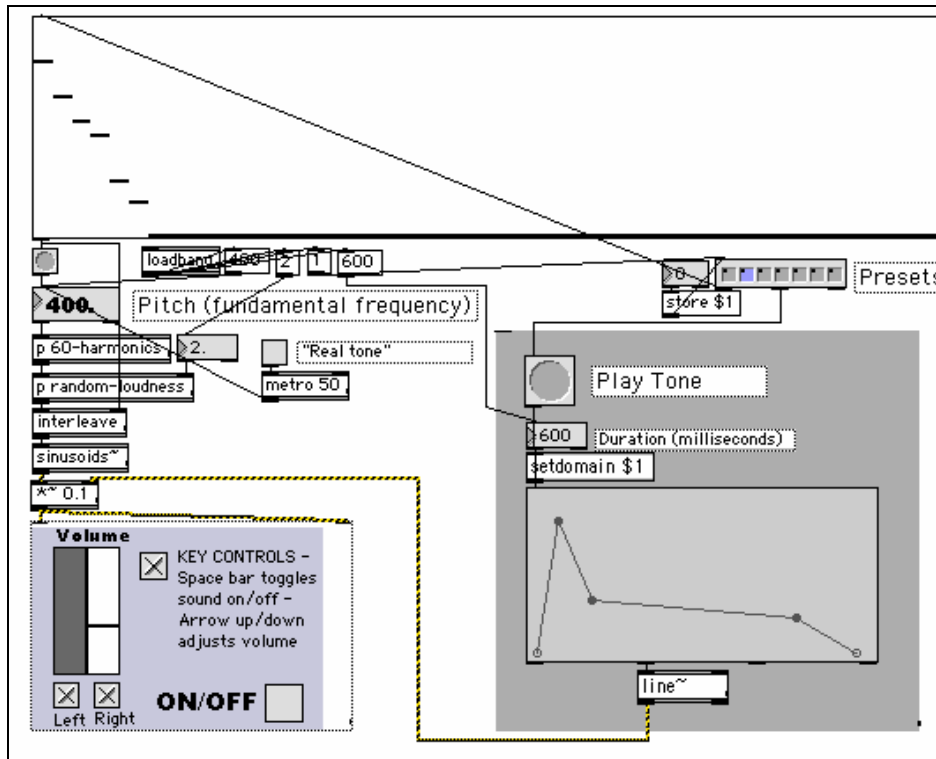
[16] By Tim Madden.

Figure 7.: A detail of the patch Harmonics *in edit mode. Notice the use of the* sinusoids~ *object for additive synthesis.*

To make the sound even more interesting, I also added the option to use a "real" tone. This is just an attempt to model slight movements in the harmonic frequencies. When this mode is enabled, each of the harmonic frequencies will change randomly within a small interval of the set frequency. This is achieved for all 60 harmonics with the subpatch shown in Figure 8. Each of the harmonics is run through an external object that does the actual calculation of the "random" frequency.

As this little demonstration patch shows, a complex tone can be created simply by adding sine tones. From such a tone we usually perceive the fundamental frequency as the pitch, and the other frequencies as the timbre of the tone. The timbral quality of the sound can, as has been shown, be controlled by for example loudness envelopes and movement in the harmonics.
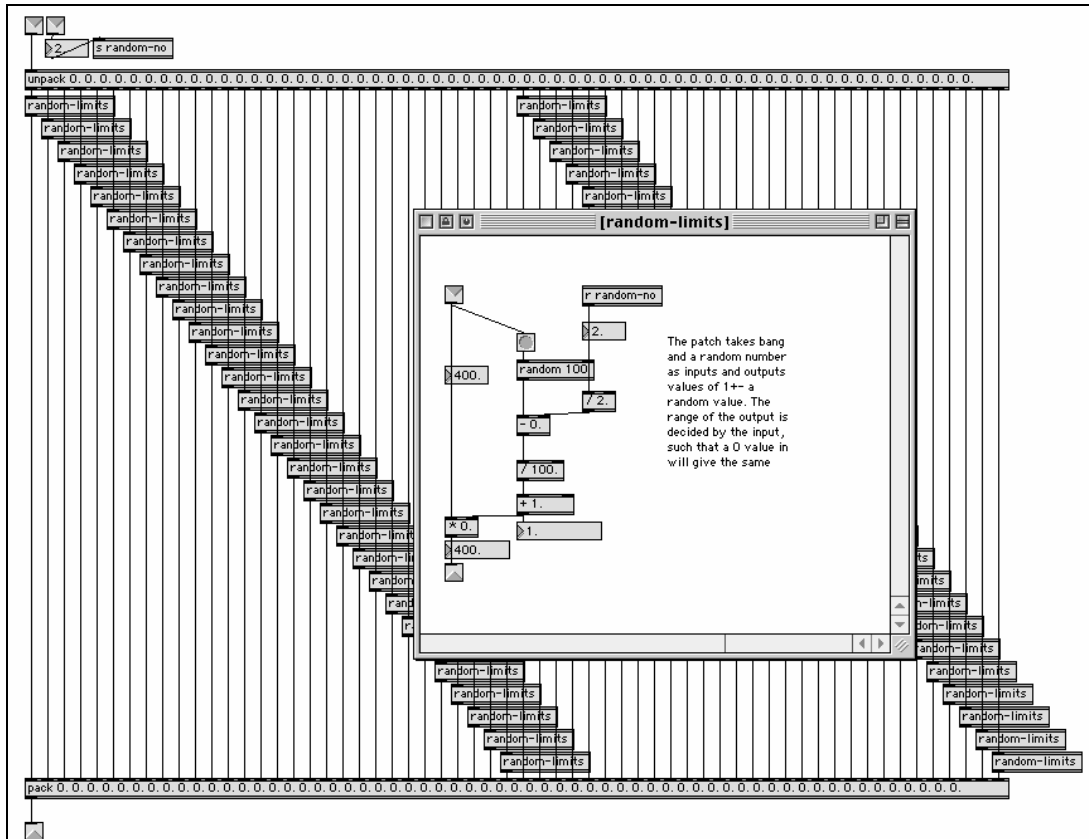
Figure 8. Random generators for each of the 60 harmonics. One of these subpatches is shown in the front.

2.4 Timbral Qualities of Instruments

Timbre is the quality that makes us able to identify different instruments. As presented in the previous section, timbre is dependent on the various harmonic frequencies, and in some cases also inharmonic frequencies, evolving over time. One possible way to differentiate timbre is by referring to the actions that create the sound. Godøy (1999) suggests the division into either *ballistic* or *sustained* actions.

The characteristic of a ballistic sound is a short, discontinuous excitation. Examples of instruments making ballistic sounds are drums, vibraphone and the piano. The amplitude-envelope of such sounds usually has sudden, high peaks in the beginning of the sound, and longer decay segments. The musician plays the instrument by mallets, keys or hands, and gives an initial excitation that leads to resonance. Therefore the musician typically does not have much control over the tone after the attack.

Sustained sounds, on the other hand, are made by continuous excitation, often by instruments that are bowed or blown. Typically all brass, wood and string instruments fall

into this category. The most important is that this method allows the performer to continuously control the instrument. A violinist can for example change loudness, pitch, and vibrato without having to set a new attack.

Different models of excitation are important for how instrument sounds, and for our ability to distinguish between them, but there are also large timbral differences between registers and dynamic levels of a single instrument. This is quite noticeable when listening to a saxophone tone played by Sony Rollins (Example 2a), in comparison to a version that is transposed one octave up (Example 2b) and one octave down (Example 2c). This transposition, done by simply doubling or halving the playback speed, results in sounds that do have some of the saxophone sound but clearly do not sound natural. Also the dynamics of the instrument is important for the timbre. The differences between playing soft and loud are considerable for most instruments. Think of a phrase played by a trumpet, first in *pp* then in *mf* and then in *FF*. For the soft parts, the trumpet would probably sound "round" and mellow, while in louder parts the trumpet would sound much brighter and harder. It is quite interesting that timbral differences between various dynamic levels or registers on a single instrument can be considerable, but we still manage to recognize the instrument.

This thesis will deal with the sound of music, and also timbral qualities of instruments, from an analytical point of view. Let us therefore turn to how it is possible to analyse and visualize sound.

2.5 Visualizing Audio

Since I advocate a perceptual approach, with audio as the source of analysis rather than notation, it is important to have some tools for visualizing the music. Often, a visual display can help us see structures and information that cannot easily be heard.

There are many different ways to represent sound, and some will be presented in more detail in Chapter 5 and 6. In this section, however, I will describe two of the most common ways of displaying auditory information, the *time-domain plot* and the *spectrogram*. There are a number of programs that can display auditory waveforms and spectrograms, but since I wanted to learn Matlab as part of this project, all the examples in this thesis are made in this programming environment[17].

---

[17] Matlab is the "standard" programming environment in the natural sciences, consisting of many ready-made *toolboxes* for doing various types of mathematical operations. Although version 6 includes some graphical user interfaces for doing common operations, it is usually controlled by *scripting* in the Matlab language.

A *time-domain* plot of a sound shows the amplitude of the waveform against time. Figure 9 contains both an overview and a close-up of the waveform of the Rollins tone from Example 2a. The overview image shows that the physical amplitude of the tone decreases, and it is a good visualisation of when a tone starts and stops. This might be practical when looking for overall structure and dynamic changes, but it does not give much information about how the music actually sounds. The close-up shows a detail of the waveform but it tells even less about musical content. Usually, close-ups of waveforms are only used in sound editing and mixing, where it is important to for example cut a sound before an attack. This can easily be accomplished by zooming in on the waveform.
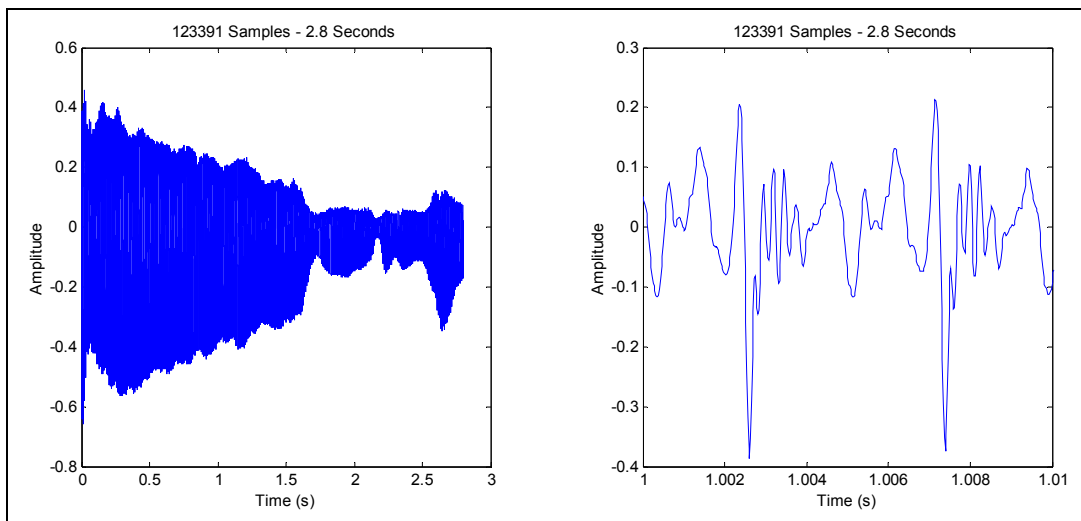

Figure 9. Overview and close up of Sony Rollins tone.

Visualizing the sound of the music, it is better done in a *spectrogram*. This is a display of the different frequencies making up the signal, or more specifically the energy in the various frequency layers of the *spectrum*. Figure 10 shows a spectrogram of the Rollins tone where we can clearly see a decrease in the upper harmonics of the spectrum towards the end of the excerpt. I will not go into detail about this here, since this specific example will be discussed in more detail in Chapter 5.

The spectrogram is found by doing a Fourier Transform on the signal[18]. This is a set of mathematical operations done on successive time *windows*, or delimited segments, of a signal. Such a window can be seen as the counterpart to what I defined as a perceptual

---

[18] Mathematical method for finding the sinusoidal components of a complex wave. There are several different algorithms for doing this, one of the most popular being the Fast Fourier Transform (FFT), which is quite efficient in terms of calculation time on a computer.

musical point in Section 1.4. Since music exists in the time domain, any audio analysis on a computer needs to use windows of a certain duration to capture changes in the signal. Setting an appropriate window size is important for the result of the analysis. To be able to capture quick changes in high frequencies, it is necessary to use a small window size, such that these changes would not "escape" the analysis. The problem, however, is that a small window size would give a poor frequency resolution, since it would not be able to capture the lower frequencies of the signal. There is thus no correct answer to what window size should be used, since it is often a trade-off between a better time or frequency resolution.
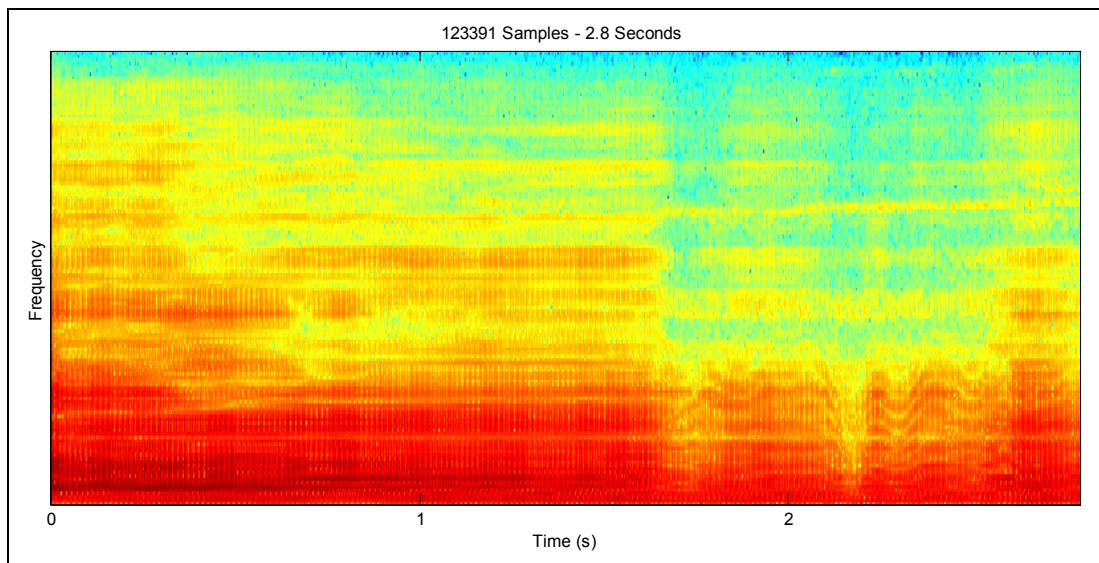
Figure 10. A spectrogram of the Rollins tone. The darker regions represent the higher levels of energy in these regions. Such an image shows how the frequencies change over time.

With some training in reading these images and listening to the corresponding music, the time-domain plot and the spectrogram can tell us much about musical structure and the sound of the music. The challenge is to be able to correlate features in the images with audibly pertinent features. Throughout this thesis I will often use such displays, and I will try to explain how and why we can get interesting information from them.

2.6   Summary

This chapter started with a discussion of what a soundscape is and what in the soundscape can be called musical sound. I argued that all sounds that we hear when listening to music do have an impact on our perception, and so they should also be taken into consideration when doing music analysis.

Then, after a brief introduction to the graphical programming environment MAX/MSP, I presented a small program that allows the user to adjust up to 60 harmonic frequencies of a tone. Adding a simple envelope controlling the dynamics of the tone, and also some "movement" in the upper harmonics, made this tone much more natural. When it comes to describing the timbre of an instrument, I referred to a division based on the sound producing actions, either ballistic or sustained. Also important is the internal timbral differences of an instrument, based on changing registers and dynamic levels.

Finally, since I prefer to use the sounding music as source of analysis rather than notation, two of the most popular ways of displaying audio information was presented. The *time-domain plot* shows the waveform over time, while the *spectrogram* shows the energy-levels across the spectrum over time. However, such physical representations of musical sound have some important shortcomings. First of all, what we perceive as significant in listening may not be easily represented in the visual images. Second, there are a number of physiological, psychoacoustic, and cognitive elements in audition which make the relationship between representations of the physical signal and our perception of the music quite complex and partly non-linear. This necessitates the presentation of some elements of auditory scene analysis in the next chapter.

3   Auditory Scene Analysis

This chapter will briefly go through some theories related to auditory scene analysis, and the processes involved in segregating sounds from a complex soundscape. The hearing system, memory, and theories of primitive grouping related to this project will be presented.

3.1   Auditory Scene Analysis

Since we do not have the option to select what specific sounds in a soundscape we want to hear, our ears receive sound coming from all directions, all the time. To illustrate the complex task our brain is performing constantly, Bregman (1990: 5) makes an analogy to our perception of waves on a lake. Imagine two narrow channels besides each other at the side of the lake, with a handkerchief half way up in each channel. Would it be possible to determine for example how many, and what sort of boats were out in the water by just looking at the movement of the handkerchiefs? Certainly, this would be impossible with our vision, but our auditory apparatus is doing such an operation constantly.

Bregman (1990) uses the term *auditory scene analysis* to describe our ability to separate sounds in time and space. An *auditory stream* denotes our perception of a separate sonic event, evolving through time[19]. For example, in an auditory scene the voice of a person singing will be perceived as a separate stream from the accompanying piano. So in this example there is coherence between the sound sources and the streams. However, in an orchestra we might say that the whole group of violins make up one auditory stream, even though there are many instruments playing. This is because the sound of all the violins is perceived as one group rather than as individual instruments. Of course, this is because they all play the same notes and the timbre of each instrument is relatively similar. If one violin is playing solo, we do indeed hear it separately, and thus it makes up its own auditory stream.

Auditory stream segregation might therefore be seen as the process of sorting out what we hear. When it comes to how this works, Bregman (1993) concludes that there are mainly three processes involved:

---

[19] It seems that this terminology has been widely accepted. Another term suggested by Kashino (Goto and Hayamizu 1999) is *perceptual sound,* a cluster of acoustic energy which humans hear as one sound, a symbol that corresponds to an acoustic entity.

- Primitive processes
- Activation of learned schemata in an automatic way
- Activation of learned schemata in a voluntary way

The primitive processes are the first to occur and govern our ability to group incoming frequencies into separate streams. These processes involve the general acoustic properties at a basic cognitive level. Such processes also form the basis for the creation of what can be called *schemata*, or a collection of learned "auditory mixtures". An example of activation of schemata in an automatic way, is for example the recognition of your own name. This hypersensitivity is probably due to the fact that most persons hear their own name spoken so often that its schema is in a "highly potentiated state" (Bregman 1993: 13). The result of such automatic activation is that we tend to react if we hear our name even though it was meant for another person. But the schema for our name can also be activated in a voluntary way, for example if a list of names is read aloud. Then an expectation for your own name might be set up in a voluntary way.

In the following I will mainly focus on the primitive processes, since I think these are most relevant to this project. But first it is important to look at our hearing system and how our memory works.

3.2   Our Hearing System

How do we actually hear sounds? This is a complicated area, and this section will only give a brief overview. Roughly, the auditory system can be divided into three parts:

- Outer and middle ear
- Cochlea (inner ear)
- Auditory regions of the brain

At moderate sound levels both the outer and middle parts of the ear are close to linear systems (Turicchia, De Poli, and Mian 2000). This means that even though they filter the signal coming in, they do this equally for the whole spectrum.

The cochlea, on the other side, works as a non-linear frequency filter. A typical example of this is the sensitivity of the ear to frequencies in the area between 2000 and 4000 Hz (Plack and Carlyon 1995). The exact region varies somewhat from person to person, and the sensitivity usually decreases gradually on each side of the region. This means that sounds with similar physical amplitudes, but dissimilar frequencies, will be perceived with

different intensity. Thus a sine wave of 3000 Hz will be perceived much louder than one at either 200 or 8000 Hz (Roads 1996: 1056).

The reason that our hearing system has developed such a frequency filter is probably a result of evolution. We are surrounded by sounds all the time, and as opposed to our seeing and the ability to close our eyes, there is no similar way to turn off our hearing. We are therefore dependent on our perceptual ability to "shut out" uninteresting information and let through important sounds. The sounds of screaming children, for example, have much spectral energy in the most sensitive region of the cochlea. Thus the child's scream will be more audible to the parents. In today's "technical" soundscape we often find that alarms and other warning devices, use tones with frequencies in the sensitivity region to get our attention. This is also cost-effective, since small speakers will be able to produce perceptually louder sounds if most of the spectral energy lies in this region.

The same feature is also important for our music perception, since instruments that have a lot of their spectral energy in the mid-range of the frequency region, will be perceived louder. Singers, for example, are therefore trained to use a *singer's formant* where they "add" more energy to the partials in the sensitivity region. This is why it is possible to hear a singer clearly, even though she or he is singing with a large orchestra (Sundberg 1999). The energy distribution across the various harmonic frequencies is also very important for the timbre of the voice or instrument.

Another significant function of the cochlea is that the basilar membrane performs a filtering of the signal such that it is possible for the nervous system to perform a frequency analysis (Mathews 1999). When there is not too much noise, the auditory system actually manages to separate up to the first eight harmonics clearly (Darwin and Carlyon 1995). It is the grouping of these frequencies that forms the basis for pitch perception of complex tones, and together with the rest of the frequency spectrum also forms the basis for timbre perception. As will be discussed in more detail in Chapter 5, a similar technique of frequency analysis is used in signal processing with computers.

3.3   Different types of Memory

It was mentioned in the last section that grouping of frequencies forms the basis for pitch and timbre perception. But for such processes to occur there must be a "buffer" where consecutive items can be compared and connected. That is because at all levels in our perception the contexts of elements are of crucial importance for their meaning or grouping. The human "buffer" is our memory, and it is believed to work in at least three

different levels: echoic memory and early processing, short term memory, and long term memory, as outlined in figure 11 (Snyder 2000).
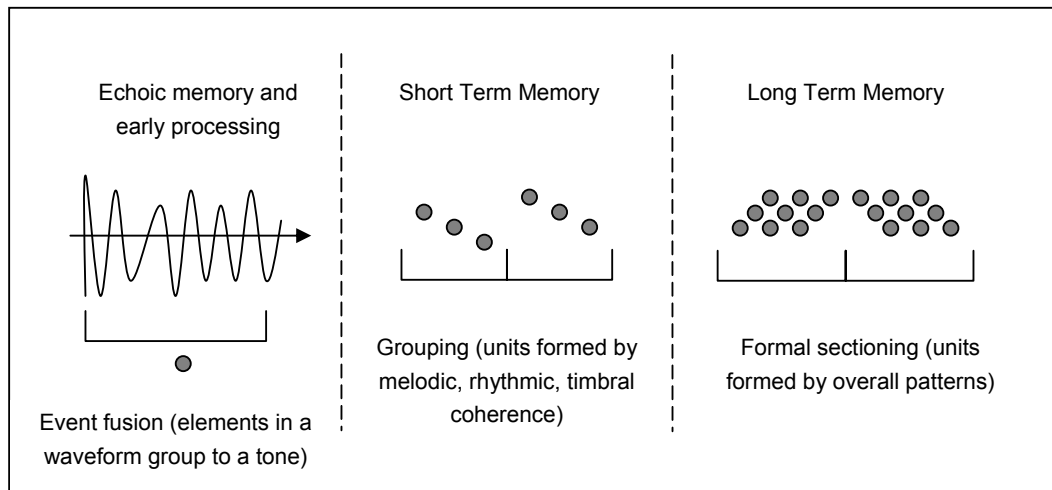

Figure 11. Different levels of memory processes, based on (Snyder 2000: 35). Note that pattern formation requires comparison of events, so each level of fusion covers a certain time span, dependent on the contents. Event fusion usually happens within 250 milliseconds, while grouping in the short term memory requires comparison of anything from 250 milliseconds up to 8 seconds. Formal sectioning may require comparisons of everything from 8 seconds to several hours.

It is in the echoic memory and early processing that events fuse together to form a single unit, for example a tone. The pitch fusion threshold is believed to be about 50 milliseconds, so if a sequence of clicks starts slowly and gradually increases, they will be perceived as a single pitch at a rate of about 20 clicks per second[20]. Snyder (2000) stresses the fact that when such pitch fusion happens, this is because of a change in our perception of the signal. The signal itself does not change except in the frequency of repetition. A parallel phenomenon is that of event fusion in vision, where single frames will be perceived as one "moving" picture when they are displayed faster than a certain threshold rate of 24 frames per second (Bordwell and Thompson 1997: 5).

It is believed that our short term memory lasts for about 3-5 seconds, and that it can contain around 8 elements (Snyder 2000: 140). In music, such elements can be, for example, separate tones or a rhythmic figure. The reason why it is not possible to say exactly how long the short term memory lasts, or how many elements it can contain, is because it is dependent on the complexity and novelty of the information. So if there are

---

[20] That is, if there are 20 clicks per second, each lasting 50 milliseconds.

many different events following each other rapidly, they will "fill up" the short term memory faster than a series of slow events.

As opposed to the long term memory, it is believed that the short term memory does not make any permanent chemical changes between neurons. Lashley used a metaphor of short term memory as a system of reverberation: "… a pattern of recirculating electrical energy that reverberates through reentrant loops of neural circuitry, sustaining the current pattern of activity…" (Snyder 2000: 47). Thus, new energy has to be introduced to the process to keep the signal in the short term memory, otherwise it will fade out. The time limit of 3-5 seconds, mentioned above, therefore refers to the amount of time that the circulating pattern of energy will be sustained without rehearsal. So the short term memory can be seen as a working memory, a buffer that sorts and organizes input before passing it on to the long term memory. If repetition occurs, it is much more likely that formal sections will be recognized by the long term memory and be remembered. So the saying that we learn by repetition is really the case, and well founded in cognitive science.

Relating these theories to the ideas of Husserl's "Zeitstrecke" (Figure 2) mentioned in Section 1.4, it is possible to imagine the memory processes involved in musical listening as in Figure 12. The dots in the figure represent musical elements unfolding in time. At the point of the perceptual "now" there will be going on an early processing, while grouping and formal sectioning will occur in the short term and long term memory. As such, all the different memory levels will always be at work and they are also dependent on each other. They also form the basis for setting up an expectation for what is going to happen next.
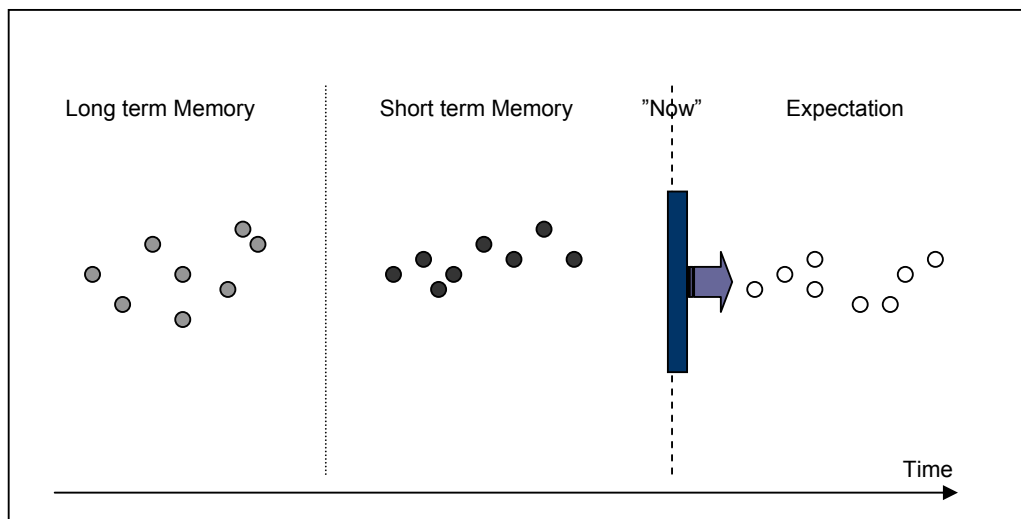

Figure 12. Perception over time, inspired by (Lartillot 2002). The different memory levels always work together, and help in creating expectation for what is going to happen next.

For this project, where the perception of short musical excerpts is the topic, the workings of the short term memory are particularly interesting. Given the fact that the musical inputs are so short, they will probably be processed and grouped entirely in the short term memory. Since the short term memory is where initial grouping is taking place, and forms the basis for formal sectioning in the long term memory, such a study might reveal interesting things about the processes involved in how we recognize music in general. Actually, studying music from such a "micro-perspective" might be a powerful method of music analysis. This will be discussed more in Chapter 4, but first it is necessary to look at how grouping actually occurs.

3.4   Primitive Grouping

Our perception seems to be based on different types of grouping at various levels. The previous section showed that event fusion occurs in the echoic memory, grouping in the short term memory, and formal sectioning in the long term memory. One type of grouping is called the primitive processes, and is often referred to as the *Gestalt principles* of grouping. These principles were developed by German psychologists early in the 20th century, and were originally explanations for understanding perception of visual material, but they also apply to sound (Bregman 1990). As shown in Figure 13, it is suggested that we tend to group events either by proximity, similarity, continuity, symmetry or common fate (Shepard 1999: 32).

The principle of *proximity* states that events close together will tend to group. This can be seen in Figure 13a, where the six dots are grouped in two clusters because three and three of the dots are closer together. In music this principle is at work when we hear for example two separate melody lines, even though there is only one monophonic instrument playing. If only the spacing between the notes is sufficiently large, two separate streams will be formed. That is why it is possible to create "polyphony" with only one monophonic instrument, something referred to as "virtual polyphony".

The principle of *similarity* states that if elements are similarly spaced, the elements that are similar to each other will tend to group. This can be seen in Figure 13b, where the grey dots will probably be grouped because they are similar to each other and different from the white dots. In music such grouping occurs for example when elements are similar to each other in pitch or timbre.

The principle of *continuity*, or good continuation, suggests that objects will be grouped if it is likely that they display a repeating pattern. This is shown in Figure 13c where the dots are grouped into two lines because there seems to be good continuation. This principle

probably governs how a series of successive notes is perceived as a scale. Even though there might be "holes" in the sequence, there is an overall coherent movement in a certain direction. The very same thing is most likely also the reason why melodies that are based entirely on adjacent steps in a scale, might be perceived as "boring" in the long run (Snyder 2000). That is because the melody will not deviate very much from the expectation set up from the previous motion. As shall be discussed later, perceptual salience is often dependent on changes that deviate from our expectations.
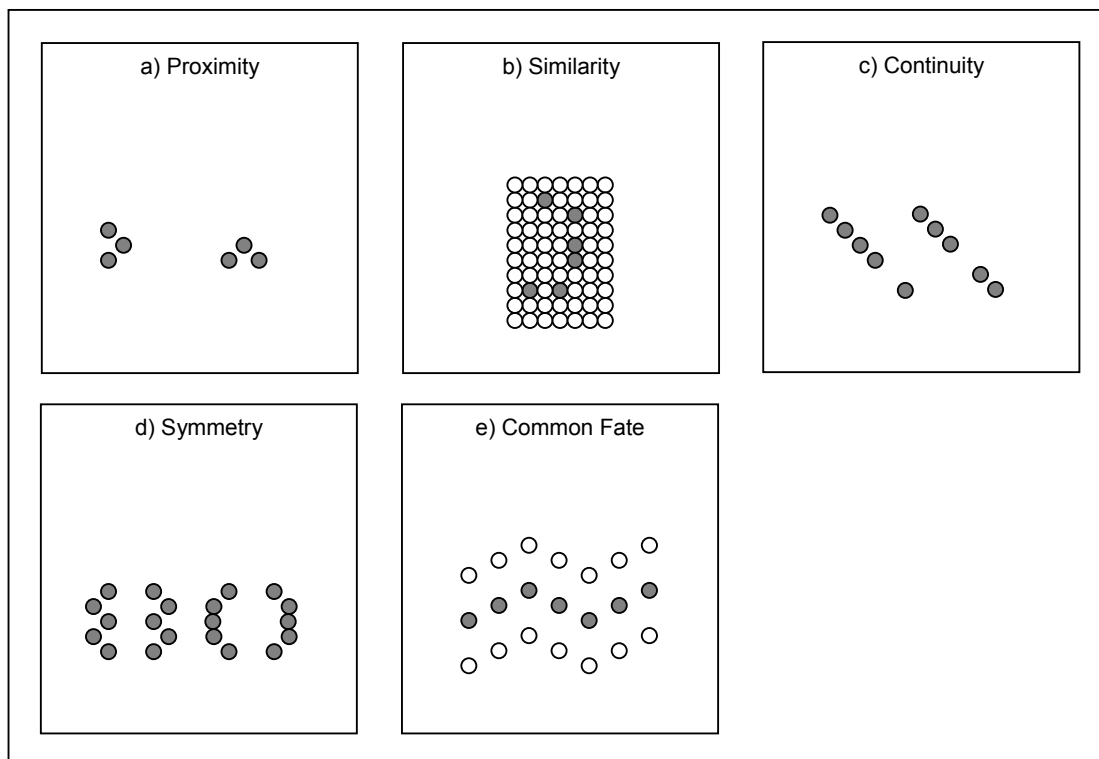
Figure 13. The Gestalt grouping principles. Based on (Bregman 1990: 20) and (Shepard 1999: 32).

Human perception is also very much aware of *symmetry* between elements, and this is also a grouping principle. Since symmetry is not very likely to occur among random objects in the real world, we would tend to group the dots in Figure 13d because of their symmetry. In music there are for example melodies that are built up by symmetric motifs or phrases.

The principles of proximity, similarity, continuity and symmetry are considered to be weak principles of grouping (Shepard 1999: 33). That is because they are often used when the signal is ambiguous, incomplete or noisy. A much stronger principle is that of *common fate*, meaning that elements moving together are more likely to be grouped (Figure 13e). This is rooted in our knowledge of nature, where it is highly unlikely that objects starting at the same time, and moving together, are not connected. The principle of common fate is

very important for our perception of pitch and timbre. That is because frequencies that are grouped to a separate tone usually have similar onset time and they move in parallel, thus they are easily grouped even in a complex soundscape (Shepard 1999).

Another important feature of our perception is how the sense of location is used to group sonic events. The ability to do spatial location is mostly based on the time difference it takes for a sound to reach our two ears. Knowing that sound travels at about 330 meters per second (Illingworth 1991: 451), such a time difference between the ears is very small, but still large enough for quite precisely finding the direction of the sound source. This is particularly useful in everyday life, where our hearing often helps our "limited" visual abilities when it comes to positioning ourselves and moving in for example a large crowd of people. Even though we can only see a limited area in front of us without moving the head, we manage to orient ourselves because of the ability to locate other people's positions through our ears. The very same feature is also at work in music perception, since frequencies coming from a specific location will be grouped together. Thus the principle of common fate applies also to spatial location, helping us to group sounds coming from the same point in space and segregate those coming from different places.

The primitive grouping principles might work separately or together. This sometimes lead to ambiguous cases where our grouping of events changes. One such example is the perception of a series of alternating tones changing from high to low pitch, as shown in Figure 14. Played at a slow speed, each tone will be heard separately in a kind of up-down-up… movement. When the speed increases there will be a certain threshold where we start to hear two separate, and parallel, streams instead of the alternating pitches (Bregman and Rudnicky 1975). Mentioned above as "virtual polyphony", this has been known by composers for centuries, and have often be used to make an illusion of both a bass and melody line played on the same instrument.
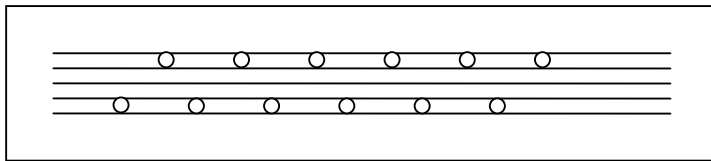
Figure 14. Tones that are far apart will tend to be perceived as belonging to separate streams when the distance between them and/or the speed is increased (Bregman 1990).

Wessel (1979) extended this principle when he showed that also timbre can control stream segregation. This was done by playing sequences of three tones with different pitch and alternating timbre, where every second tone has a higher or lower level of brightness, shown with o and x in Figure 15. When the difference between the two levels of brightness

is small, the sequences are heard as ascending triads. However, when the timbre difference is increased, the tones are grouped by timbre, and two descending lines are heard.
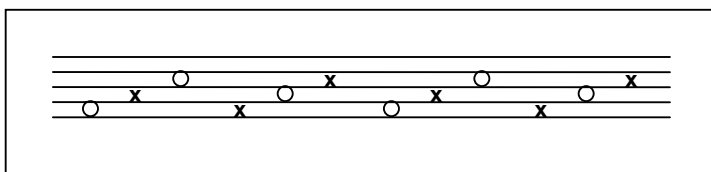
Figure 15. Stream segregation based on different timbres. The o represents a timbre with less brightness, while x has more brightness (Wessel 1979).

The primitive processes presented above are general for our perception, both of visual and sonic events. But there are some important differences between the study of general sound events in a soundscape and the study of music perception. Bregman (1990) argues that even though the same principles are at work, their application is often quite the opposite. In daily life we need to separate sounds and find their location in space to be able to survive. In music, however, we do not necessarily want to separate the sound from all instruments or notes being played. It is actually this mixture of sounds in time and space that is the constituents of music, and the primitive grouping processes can therefore be used to our advantage. An example of this is how the principle of proximity makes us perceive a group of violins as one coherent group. It would be very difficult to listen to a symphony orchestra if we perceived each instrument or even every single frequency as separate events. But to help such instrument grouping to occur, it is important that the instruments are in tune, and that the musicians sit close. If there were only two violins playing, on each side of the stage, it is not very likely that grouping will occur. However, if a group of violinists are sitting close together, and playing the same notes, it is much more likely that they will be perceived as one coherent instrument group.

But grouping can also occur with several different instruments, and an important part of the study of orchestration is to learn how to write scores where the sounds of various instruments can blend together to form new sounds. So the goal of the composer is actually to create fusion of sounds rather than separation. This, of course, requires the composer to have a good knowledge of the principles of auditory fusion and segregation, as pointed out already in the 19th century by Stumpf. In a study of why we hear one single pitch from an instrument rather than a set of its harmonic frequencies, he said that tone fusion was a cognitive principle and central in our perception of sound (Bregman 1990).

Another feature of music perception, is that we tend to separate music from other sounds in the soundscape (Bregman 1990). The music itself can thus be seen as a separate auditory

stream. Sounds that do not belong to the music will be heard independently, and that is why we are aware of any noise, for example coughing or talking, when listening to music.

When it comes to the groupings within music itself, it is important that the separation is strong enough to hear the melody apart from the accompaniment. As mentioned above, several instruments can sound as one group if they sit close together, and play the same pitches with similar timbre. Musicians playing a solo, on the other hand, will need to create sounds that are sufficiently distinct to be heard independently. This can be done by a difference in dynamics and pitch range between the soloist and the other instruments. Quite often the accompanying instruments play more "quiet" trying to blend in with each other, so that the solo instrument can be heard more easily. If the soloist also uses some controlled vibrato, this way of slightly changing both pitch and timbre will be perceived as coming from one source, but it will most often "stick out" from the rest of the soundscape. Another important way for the soloist to be heard is to play or sing rubato. Such a way of stretching the rhythmic grouping, by playing onsets a little before or after the others, also makes the instrument more distinct.

Summing up this section there are many different ways that primitive grouping can occur. Most important for music perception is probably the concept of common fate, and the fact that we tend to group frequencies that appear from the same point in time and space. For musicians and composers such information is significant, since it can help controlling how the sounds of instruments should either blend or be heard separately.

3.5   Schema Theory and Cross-Modality in Music Perception

The primitive grouping processes presented in the previous sections are the first to occur when we perceive a signal. But also important are the segregation processes based on schemata. Bregman defines *schema* as a "control system in the human brain that is sensitive to some frequently occurring pattern, either in the environment, in ourselves, or in how the two interact" (Bregman 1990: 401). Schemata are developed through learning, and they form an integral part of both perceptual and cognitive processes. When we perceive a signal, a number of related schemata will be activated. For example, we may have separate schemata for the letter 'a', the word 'apple' and also for the whole grammatical structure of a sentence. But hearing the word 'apple' we may also start thinking about 'fruit' or 'computer', and choosing between these two connotations will be done by referring to other activated schemata, as well as the context.

Since we often associate many different things to the same input, this may sometimes lead to *ambiguity*. In the case of the 'apple' presented above, the correct meaning can probably

easily be found from the context. But there are also cases where it is not clear which version is the correct. A visual example is the picture displaying either a vase or two faces. Here some people easily spot the faces, and have problems seeing the vase. However, if someone is pointing out the vase, it is much more likely that this will be seen first. This way of preparing the senses for an input is called *priming*, and is a significant part of our perception. In the moment we hear the word 'vase' our brain will relate this to a schema of a vase and give us some idea of its basic features. These features are then used when grouping the elements in the image. A similar phenomenon is evident in auditory stream segregation. When listening to a full orchestra playing, we will probably not focus specifically on any instrument, but if we are trying to follow for example the trumpets, this can fairly easily be done based on our knowledge of a trumpet's sound.

An interesting feature of schema theory is how flexible schemata are, and how they can be used to generalize. Clearly, our idea of 'apple' does not correspond to a specific apple, but rather some set of features that an apple seem to have. So whether it is small or large, green or red, it will trigger the same schema. Also, it is often sufficient to see only a small part of the apple to recognize what it is. For example, if there is a box with oranges, bananas and apples, we can easily separate the different fruits, even though we can only see parts of each item. Another example of such *occlusion* is that of a chair standing in front of a table. We can see the whole chair and only parts of the table, but still it is easy to identify the two objects and to tell what legs belong to each of them. This is done by first perceiving the visible features, grouping these together and finally comparing them to schemata based on our knowledge of shapes, colours and how a table and chair should look like.

Occlusion is also particularly significant in music perception. The sound of an instrument is based on grouping of frequencies, and when several instruments are playing at the same time there will necessarily be many frequencies that "overlap". The result might be that certain parts of the frequency spectrum of an instrument are *masked* by frequencies from other instruments. However, our knowledge of the features of each of the instruments will help in segregating them based on the frequencies we do hear. In cases where full masking occurs, for example if a loud trumpet plays at the same time as a soft recorder, we will only hear the trumpet.

In the previous sections I have discussed how both primitive processes and schemata can help segregating various perceived elements, either visual or auditory. What is important to remember, is that our perception usually works by the joint efforts of our different senses. For example, we usually both see and smell food at the same time, and thus both these features help in triggering schemata that again will result in a feeling of how good the food is. Such *cross-modality* is also significant in music perception. Godøy (2001) discusses

how musical imagery and motor theory are related to our images of sonic objects. This is based on a model where action, vision and sound play together in our perception (Figure 16).
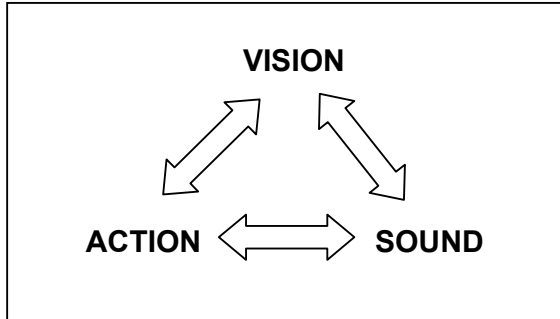
VISION

ACTION ⟷ SOUND

Figure 16. Godøy's figure of the importance of motor-mimesis in relation to sound (Godøy 1997b).

An example of this is how we perceive the motion of a drumstick approaching a drum. By seeing how the stick moves, and associating this with our experience of the action of something falling, we will expect a sound to appear in the moment that the stick hits the drum. In cases where these three features do not correspond well, for example when watching a video tape where the images and sounds are not synchronized, we will be confused and start to think about what is wrong.

Another example of how the visual and auditory senses work together, is the fact that it is easier to follow the speech of another person when we see the person's mouth moving. A famous experiment showing how we unconsciously combine visual and auditory information was done by McGurk and MacDonald (Bregman 1990: 183). They played videotapes displaying a person saying the phrase "ga-ga" but with the overlaid sound of "da-da". The result was that the test persons thought they heard "ba-ba". This result is logical, considering that "da-da" has quite similar acoustic features to that of "ba-ba", while "ba-ba" is pronounced with the lips open just like in "ga-ga". Each of these sounds was correctly identified if only the audio part was played, but when the video was running it resulted in a combination of the visual and auditory information.

3.6   Summary

In this chapter I have presented various topics of perception related to the project. For the discussions in the next chapters, I think it is particularly important to point out the non-linearity of the auditory system. This means that we have to be cautious when analysing the physical signal, and take our perception into account. Also important is the fact that our

memory can roughly be divided into three elements: early processing, short term memory and long term memory. These govern the grouping of perceived elements into units, events and formal sections, respectively. Auditory scene analysis, or how we manage to separate auditory streams from a complex soundscape, is done by primitive processes and/or mental schemata. The primitive processes are based on the Gestalt principles, i.e. on proximity, similarity, continuity, symmetry and common fate. Schemata are mental structures for interpreting what we perceive, and our various schemata are based upon previous experience and learning. Such schemata may include elements from other modalities such as vision and movement, as well as of sound. I believe all these mentioned phenomena are at work in music perception, and they form the basis for our music recognition. In the following discussion of recognition and salience points, however, I will concentrate on the acoustic signal.

4   Recognition and Salience

This chapter looks at our ability to recognize music from short musical excerpts and the importance of salience.

4.1   A Model of Music Recognition

As suggested in Chapter 1, recognition of music can be divided into segregation of auditory input and recognition of musical features. A model of this is shown in Figure 17, and was proposed as early as in the 19th century by Lissauer (Peretz 1993). The first stage involves analyses of the auditory input and primitive grouping of events. The second stage involves the contact between these events and mental schemata of previous experience. If there is a correspondence between these two levels, recognition occurs.
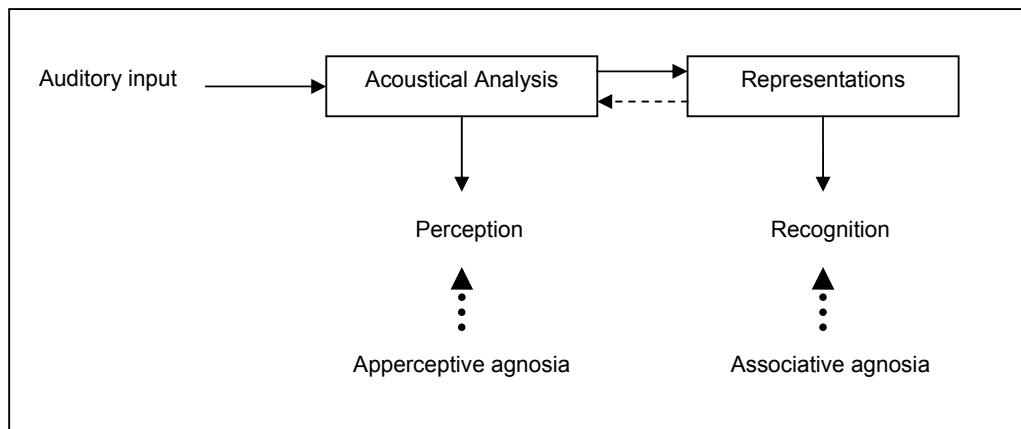
Figure 17. Representation of the two-stage recognition process, based on studies with brain-damaged patients (Peretz 1993: 207).

Peretz (1993) interest in this model was based on her study of brain-damaged patients, suffering from *auditory agnosia*. This is a condition where the subjects can perceive changes in frequency, intensity and duration, but are unable to relate these to any schemata. As such, auditory agnosia involves a problem of recognition and identification that cannot be explained by deafness, nor by a difficulty in verbal expression.

If a model such as shown in Figure 17 is correct, Peretz argues that agnosia can occur at two different levels. *Apperceptive agnosia* would be when people cannot recognize sound events due to deficiency in the perceptual analysis, while *associative agnosia* refers to people that cannot relate the perceptual attributes to mental schemata. I will not go into

details about her argument, but rather skip to the conclusion saying that such a two-stage model might look intuitively correct, but in reality seems too broad. She suggests that it would be better to have models allowing for parallel entries, which conceive of perception as being an intrinsic part of the process of recognition. She further points out that connectionist networks, or parallel distributed processing, might be a better way of investigating the problem.

The nature of connectionist models will be discussed more in Chapter 6, but now I will show one model that Peretz suggests when it comes to recognition of music (Figure 18). This model is based on work with both brain-damaged patients and normal subjects, and she has looked for what musical structures are being used in recognition of a song. The model is based on an idea that every person has some kind of *lexicon* with reference to all the tunes that the person knows. The final recognition is based on a best match between the input signals and what is available in the lexicon.
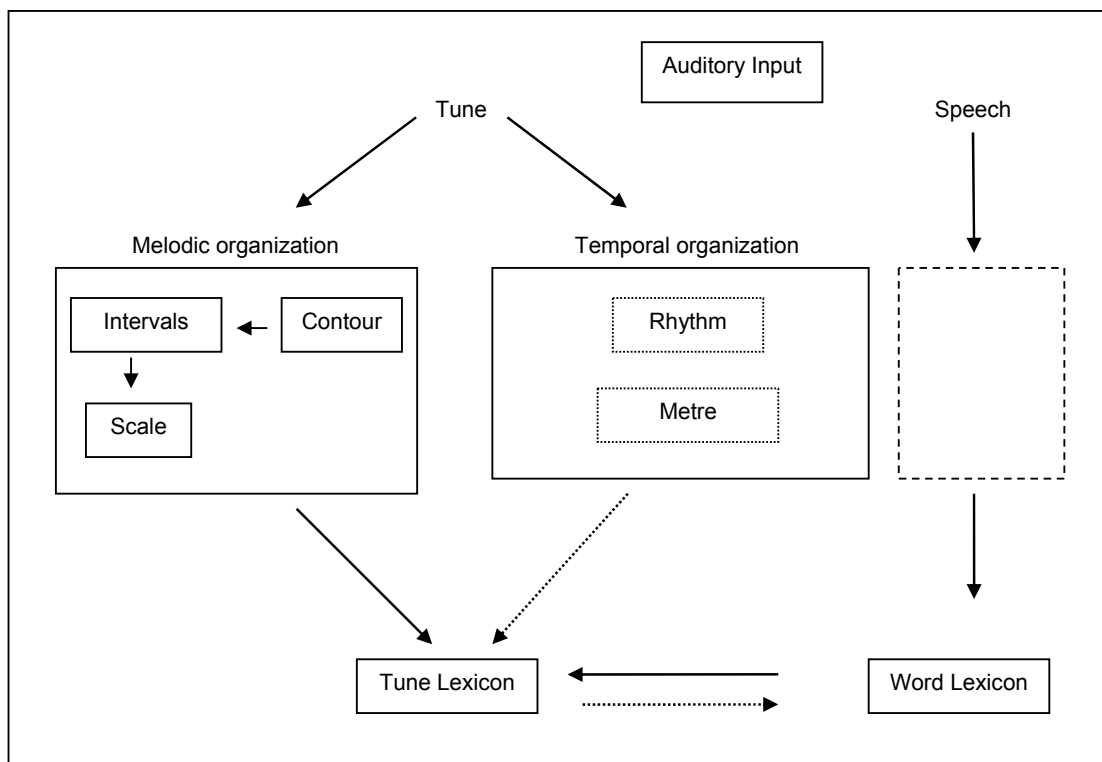
Figure 18. Functional diagram showing processes involved in recognition of a tune by brain-damaged patients (Peretz 1993). The hard lines denote the primary access, while dotted lines show secondary or optional access.

Peretz suggests that recognition of a traditional tune can be done either by melodic or temporal organization, or speech recognition. Of these three, speech recognition is considered a separate process where the lyrics are checked with a separate word lexicon,

while melodic and temporal organization is the basis for checking a tune lexicon. From her discussion, it is quite clear that Peretz believes melodic organization, subdivided to contours, intervals and scales, is the most significant for music recognition. She also argues that "Variations in intensity (dynamics), in spectral composition (timbre), and in speed (tempo) probably facilitate recognition in so far as they respect the structure of the original tune […] but apparently are not determining factors, at least not when highly familiar tunes from the Western musical system are involved" (Peretz 1993: 215). I think the last part of this quote sums up the whole problem with the model. The fact that she only focuses on traditional western music makes such a model, focused around melody, rather limited in its usage.

The problem, of course, is that her model will most likely not be able to explain recognition of any type of music that is not melody-based. I also have problems seeing how this model could adequately describe recognition of short music excerpts. In a short excerpt we might only hear one tone of the melody but still be able to recognize the song. Thus, for making a general model of music recognition, I think we can learn much from investigating short term music recognition. This is what I will do in the following sections and also try to see what musical features contribute to such a phenomenon.

4.2   Measuring Recognition Time

I suggested in the introduction that it is possible to recognize music from a short music excerpt. But how much time is actually needed for such recognition to occur? Is it really the case that we manage to recognize a song in only one second? Does this apply for all songs and all musics?

As an informal test I made the patch *ST-Perception* (Figure 19) which measures recognition time of a song. The program allows the user to start and stop the playback of a song by clicking with the mouse, or using a MIDI-controller such as a keyboard or a pedal. The recognition time is calculated and a new song is automatically loaded when the current song is recognized. I encourage the reader to test the features of the program by running the *ST-Perception* application on the CD-ROM!

As can be seen from the interface (Figure 19), the patch can run in two different "modes". Either the songs start at the beginning (intro), or at a random position (random) in the sound file. This feature was added since I wanted to check if there is a difference in recognition time dependent on where in the song the listening starts. From the interface, the user can also control the volume of playback, and save the recognition results to a text file.
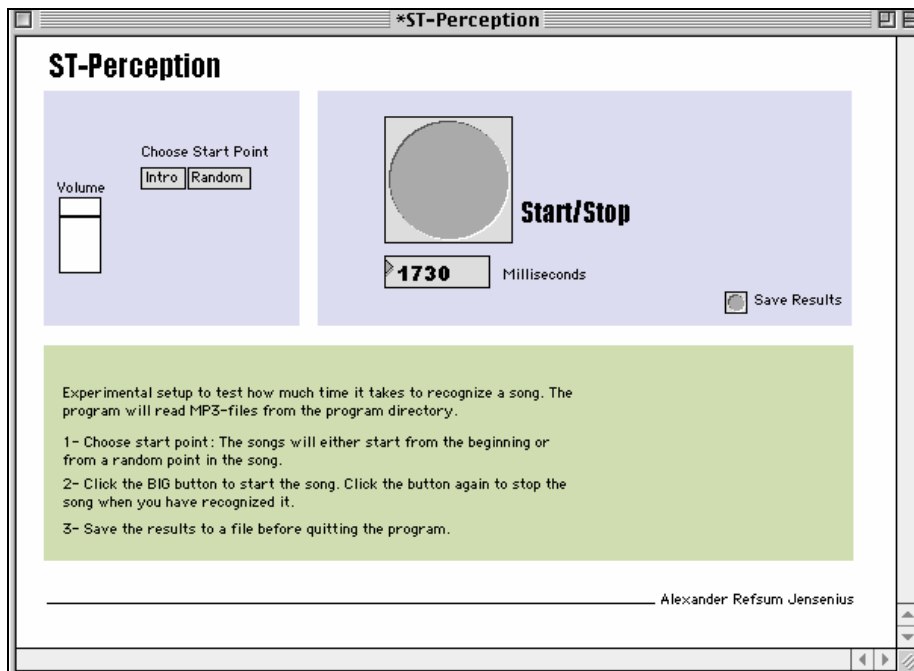
Figure 19. User interface for patch that measures short term perception.

The interface sends information to a subpatch (Figure 20) that contains most of the operations of the program. As can be seen from this subpatch, there are five inlets on the top that receive information from the interface. At the core of the patch is the *movie* object, which can play any file type supported by QuickTime. For this setup I have used five MP3-files as the input, but this could easily be changed to anything else, even video files. To avoid too many patch cords, I use the *send* and *receive* objects (abbreviated to "s" and "r" in the patch) to transmit information "wirelessly". A hidden *loadbang* object initializes the patch and loads the sound files when the patch is opened[21].

When the user presses the "start/stop" button, a series of events happens. First, the start position of the song is found from the sound file. After some testing I found that using a *timer* measuring the time between the user's activation was not precise enough. This is because sometimes the computer waits a little before it actually starts playing the file. A reason for such inconsistent response from the computer is the fact that it is a multi-purpose system not dedicated to only one single operation. Therefore background processes in the operating system might cause inconsistent latency. Such latencies of perhaps up to 500 milliseconds are usually not a big problem, but for this patch where the whole point is to measure recognition times down to some hundred milliseconds, such

---

[21] MAX/MSP allows the user to hide objects and patch cords when in run mode. This helps to make the patch easier to read.

deviations could not be tolerated. Instead I decided to calculate the time difference directly from the sample output of the file. Therefore the first operation done when the user hits the "start/stop" button is to find the specific time code of the sound file. This is done in the subpatch displayed in Figure 21. This subpatch also contains the *random* object that chooses the random start times, if this mode is selected. As can be seen, I have used many *gates* and *triggers* (abbreviated to 't') throughout the patches. This is to secure that things happen in the right order and to secure a stable program.
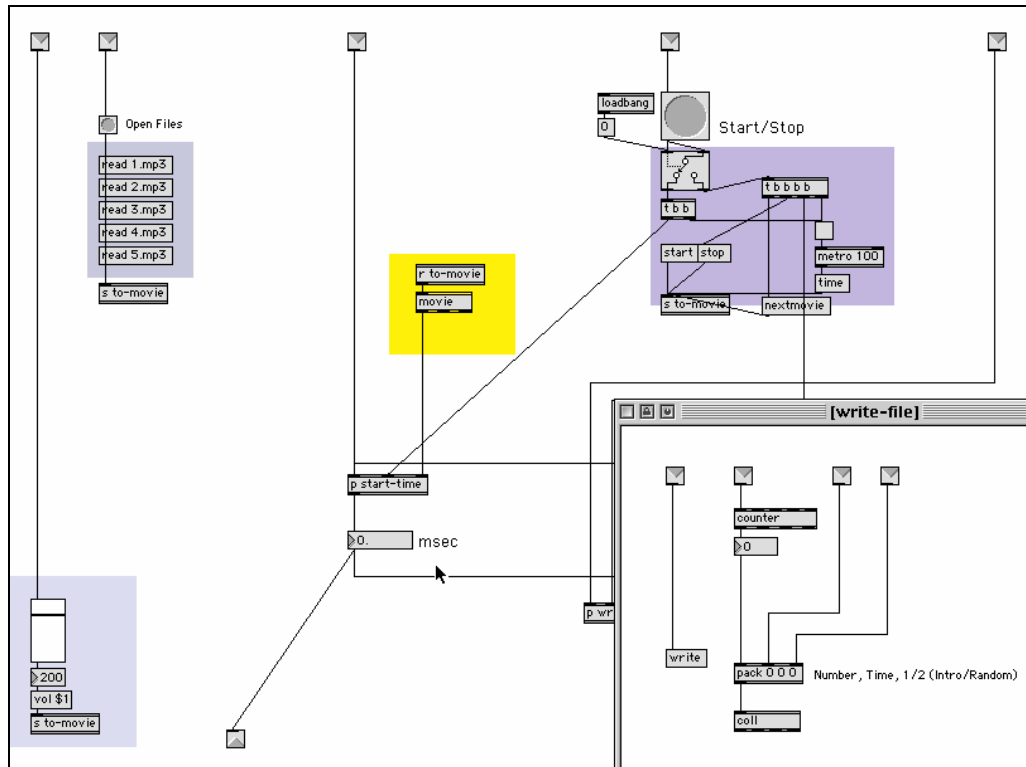
Figure 20. The "interior" of the ST-Perception patch, with the display of the subpatch that saves the results in the bottom right corner.

The program seems to work according to my intentions. It was never planned to be used for a formal, full-fledged experiment, and I will therefore only present some suggestive results. The recognition times varied considerably, everything from 300 milliseconds to 10 seconds. For songs that started at the beginning of the sound file, recognition times were generally less than 3 seconds. For songs with a random start position, the results were much more diverse. In some cases the song was recognized very quickly, while at other start points it took a much longer time. I therefore conclude that it is indeed possible to recognize a song in one second, but the exact recognition time seems to be subjective and very much dependent on the contents of the music. As will be discussed in more detail in

the following sections, I believe that the appearance of salient points, e.g. refrain, voice etc. is significant for recognizing the song.
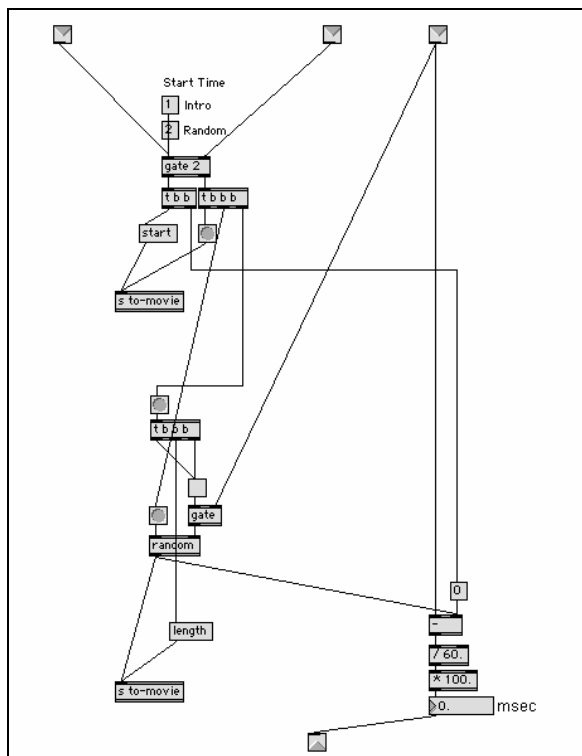
Figure 21. Subpatch controlling the start and stop of time.

The main interest of this test patch was to make an experimental setup that can be used for finding recognition time of music. Although I have tested it and found that it works well, I leave to someone else to do more systematic and large-scale experiments. However, I do think that a program like this can work well in such a context. When it comes to improvements of the program, it would be nice to include calculations of some relevant statistical information. Another interesting feature would be the option to randomly choose between thousands of sound files. That could remove the "priming" effect that a limited number of songs might create.

4.3   Analysis of Two Examples

The recognition time of a song is interesting mostly if it can tell us something about the musical content of a song. In this section I will therefore analyse two different examples, and discuss what may be the reasons they can be recognized.

The song *Tears in Heaven* by Eric Clapton (Example 3), was a major hit from the *Unplugged* album released in 1992. It has been played regularly on radio and TV, and

many people therefore know it well. Example 3a is the shortest excerpt and lasts only 134 milliseconds. At this length it is barely possible to hear a pitch of the tone, although there is a quite distinct attack. Even though it might not be possible to recognize a guitar, many people comment that it sounds more like plucking on an electric bass, so there is certainly a string-like timbral quality to the sound.

The next excerpt, Example 3b, lasts 380 milliseconds. Now it is possible to recognize the interval of a major second quite clearly. When it comes to the timbre of the instrument, it should now be possible to hear that this is a guitar, and the "space" in the sound might lead us to think of an acoustical guitar. People knowing the song well would probably be able to spot the entire song already after such a short excerpt.

In the next excerpt, Example 3c, lasting 960 milliseconds, we can hear the four tones E-F#-A-A. Notice how the E and F# might be perceived as leading up to the accentuated A, giving a sense of pulse and also tonality. The last A may be perceived both as a passing tone, establishing the pulse, as well as a bass tone, establishing the tonic.

Even though the excerpt in Example 3d, lasting about 3,5 seconds, reveals some more tones and further establishes the rhythm, it does not really give us any more relevant musical information that can guide our recognition. Actually, I will argue that the next important point is when Clapton starts singing after about 13 seconds (Example 3e). If you know the song you could probably have recognized it only from the first three tones.

From this example I think we can conclude that an excerpt of only one second is sufficient for establishing a sense of timbre, tempo and tonality. I think the most salient features here are the tones E-F#-A, and the sound of the guitar. In terms of musical parameters this means that melody and timbre are the most prominent.

A quite different example is Sarah McLachlan's song *When She Loved Me* from the film *Toy Story* (Example 4). Starting with what I would call a "low-salient" introduction, there is after one second (Example 4a) only a single chord played by a group of violins. Probably, you should know this song very well in order to be able to recognize it from this input only. Adding some more seconds (Example 4b and 4c) presents some more tones but neither melody nor harmony contains anything that is particularly distinguishable. Still after 9 seconds (Example 4c) it is difficult to hear whether this is a piece from the 19th or 20th century, and whether it is classical, film or pop music. With the addition of a piano (Example 4d), we understand that it is more towards the latter categories, but still there is nothing particularly salient. I think it is first when the voice is added (Example 4e) that we might be able to recognize the song. Although it is possible to hear the violins after one

second, it is probably only after about half a minute that most people would be able to tell this song from any other "violin/piano-song". It is quite interesting that it actually takes more than 9 seconds before we can even decide which style or century this music was written in.

The Clapton and McLachlan songs show some of the width we may encounter in musical material. I will argue that the former has some very salient features right in the beginning, while the latter introduces some salience after half a minute. As such, they are good examples when it comes to demonstrating that recognition is very much dependent on musical content. From these examples we may therefore conclude the following:

- It is possible to recognize a song from a short musical excerpt.
- It is easier to recognize a song if there is some salient feature present.
- It is not clear which musical parameters are more important when it comes to recognition.
- The sound of the music is important for recognition.

In the following discussion I will therefore look more into the concept of salience, the roles of the musical parameters, and the importance of sound.

4.4 Salience

As defined in Section 1.4, salience denotes the most noticeable or pregnant feature of a perceptual input, be that in language, pictures or music. As such, salience may be a subjective feature, often depending on cultural background. However, it often seems that many people agree on what is salient, and composers through all times have known how to use this to their advantage.

As discussed thoroughly by Meyer (1956), expectation is important for our emotional response to music. Our reaction to something will be totally different if we know what is going to happen, rather than if something happens unexpectedly. This does not occur only in music but is rather a central feature of our whole existence. Picking up a ringing phone with the expectation that a certain person is calling, will certainly arise confusion if it turns out to be another person. Therefore, the context that events happen in is crucial for our emotional reaction to it. A musical example of this is the IIm7-V7-I full cadence, which we have heard innumerable times. So hearing a IIm7-V7 we immediately expect a return to the tonic I, so when sometimes there is a IIm7-V7-VIm progression instead it does not result in the same sense of target. That is also why such a progression is called an interrupted cadence, and will probably be perceived as perceptually salient. As a general

rule, we might therefore say that the most salient events are the ones that deviate from the conventional patterns or rules.

One of the most prominent cases of musical salience is that of modulation. A modulation towards the end of *We are the World* (Example 5) serves as an example. The shift in tonality gives a subjective "kick" while listening, and has therefore been used extensively in several styles. However, because of changing contexts such a musical figure might appear salient in one sequence, and unnoticeable in another. While a modulation at the end of a pop song might result in a perceptual "kick", we would probably not react to the same harmonic progression in a jazz song. That is because jazz songs often modulate frequently.

It seems that musical salience is often related to large change between consecutive events for example in dynamics. Take the case of a symphony concert, where after listening to violins playing pianissimo for a whole section we might perceive the change to a brass section in fortissimo as very salient. Dramatic changes in instrumentation and dynamics will most certainly result in the expected effects.

So what happens if many such salient "tricks" are used at the same time? This is often what is done in radio and TV commercials and film trailers. Since they only have a limited time to get through with their message, they pack a lot of information into as little time as possible. Such short sequences are therefore filled with all types of effects that might increase our perceived rate of salience points. Sometimes, however, the opposite effect might be observed, since at a certain level the constantly changing stream might be too fast to cause affection. When we are tired, for example, a radio station with extremely short and feature-packed commercials, fast talking, and techno music, all with a highly compressed soundspace, may lead us to start loosing track of the actual content and perceive it as one long and blurred sequence.

The opposite is the case for the genre popularly called "muzak", music that by definition tries to be "low-salient". This is often played in public places where the music is supposed to create an atmosphere and never catch attention. The intended effect is achieved by allowing only simple and seamless harmonic changes, dissonant-free melodies and non-accentuated rhythms.

I should also mention another domain where auditory salience is of vital importance, namely *system sounds*. We are surrounded by artificial sounds in everything from alarm clocks and mobile phones to ATMs, door bells, and computers. As I have discussed elsewhere (Jensenius 2000), such system sounds are designed with the intention of being as salient as possible since they are short and need to carry much information. This is

achieved by for example making warning sounds very dissonant, based on a minor second, tritone, or major seventh interval, and containing much spectral energy in the sensitivity region of our auditory system.

As a conclusion, I think it is possible to say that perceptual salience often occurs based on a sudden and unexpected change in some parameter.

4.5   Measuring Salience

How do we perceive salience in music and is it possible to measure it? To test this I made the patch *Measure Salience* that can be used for "measuring" a person's response to music. As can be seen from the user interface (Figure 22), the user controls a slider either by moving the mouse, or a slider on a MIDI-device. The idea is that the user adjusts the slider while the music is playing, such that a higher value on the slider would correspond to a higher perceived salience. A simple "salience curve" is then plotted in the interface.
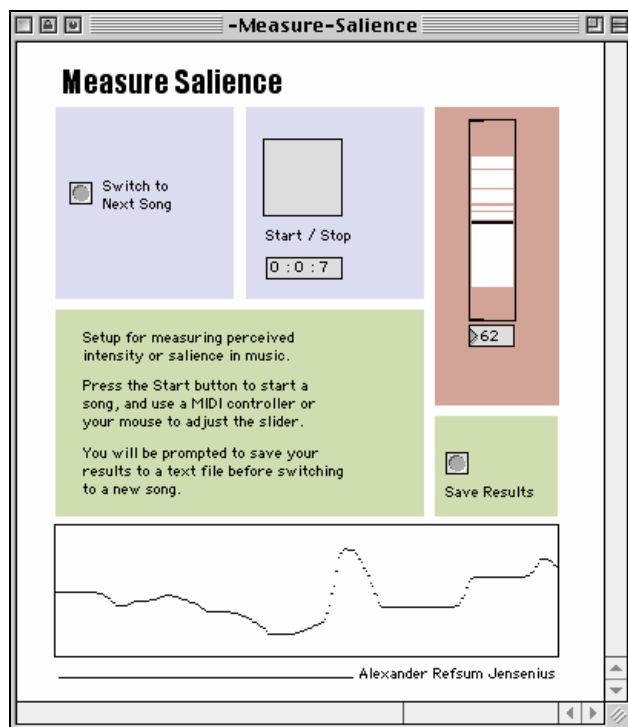
Figure 22. The user interface for the Measure Salience *patch that "measures" perceptual salience. The user controls the slider by mouse or a MIDI-device.*

The "interior" of the patch (Figure 23) shows the different parts of this little program. A *loadbang* object initially loads a set of MP3 files to the *movie* object. The start/stop button starts the music, and also starts a *metro* object that retrieves the time code of the sound file every 200 milliseconds. The time code is sent to the *coll* object together with the measured

salience from the slider in the user interface. The files saved from the *coll* object thus consist of one column with the time code and another column with the associated perceived salience at that time.
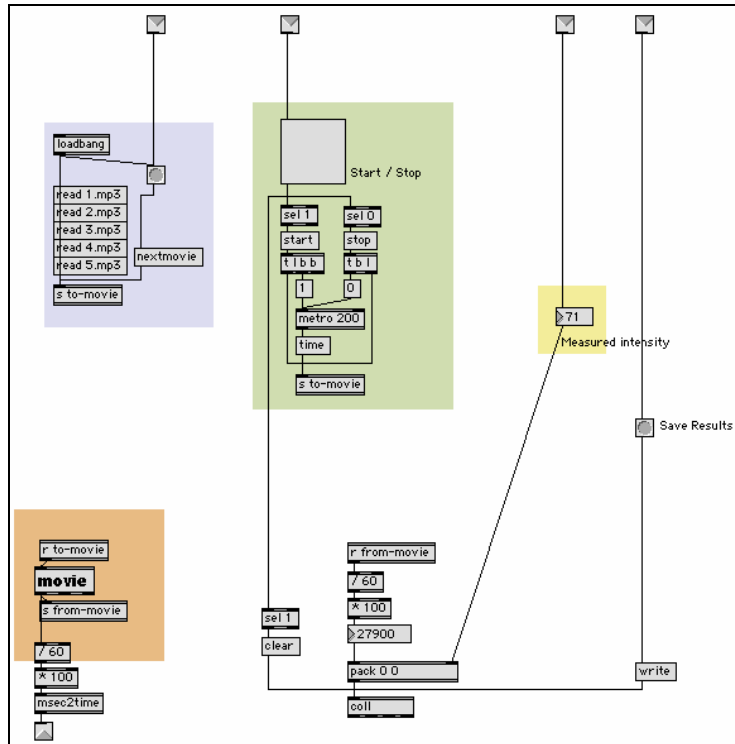
Figure 23. The "interior" of the Measure Salience *patch. The patch is built around the* movie *object, and uses a* coll *object to store the measurements.*

Testing myself, I was not really sure what sort of measurements to expect. Initially I forced myself to adjust the slider continuously. This resulted in measurements like the graph shown in Figure 24.
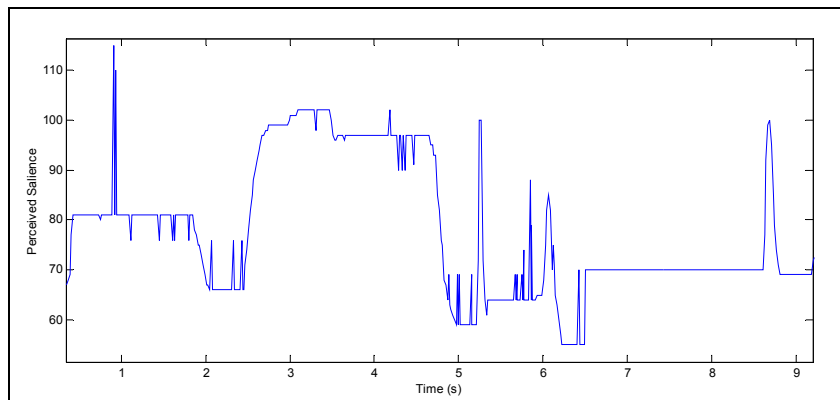
Figure24. Plot of perceived salience versus time.

However, this felt a bit awkward after a while, and after some testing I ended up marking only certain spots in a song. I also found it difficult to determine whether one musical point was more salient than another, and how this could be shown with a higher or lower peak at the salience point.

After some testing, I think there seems to be two possible models for how we perceive salience. The first is based on my results from the patch, where measurements are shown as a continuously updated signal (Figure 25a). Another possibility would be to look at only the peaks of such a signal (for example by using a high-pass-filter). This would result in salience points such as shown in Figure 25b. I think the latter model would be more appropriate, since we seem to be more focused around certain points in time, rather than a continuous stream. From this informal test I have no good answer, but it would be very interesting to see psychological experiments on this phenomenon.
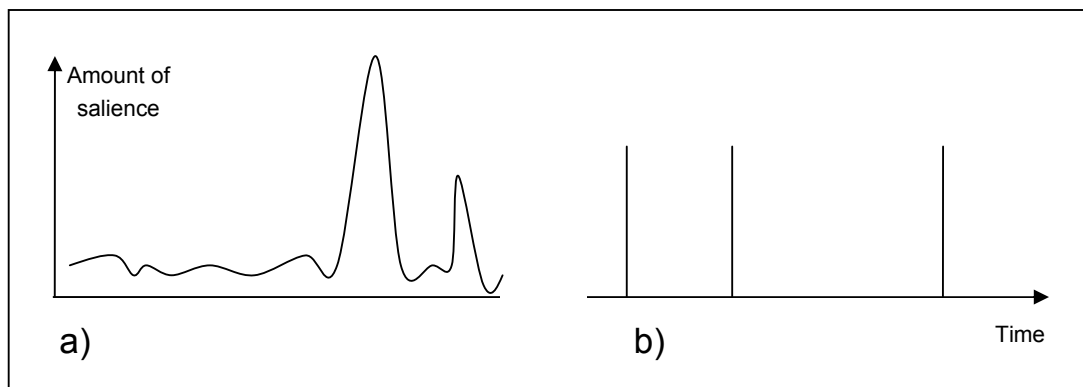

Figure 25. Do we perceive salience as specific points in time (a) or as continuously changing (b)?

Although it is interesting to find out exactly how salience is perceived, it is equally interesting to understand what musical features make a musical point salient. This will be discussed in Section 4.7, but first it is necessary to look more into perceptual and physical attributes of the musical parameters.

4.6   Musical Parameters

Approaching music theory from a perceptual point of view, we should also try to incorporate parts of traditional theories. That way it is possible to draw on the vast experiences and analytical works of other scholars, and extend this knowledge with new models. But this requires a clear notion of how traditional concepts can be expressed in terms of perceptual and acoustic qualities. This section therefore looks at musical parameters, how they are grouped, and how they can be included in perceptual models.

It is important to remember that what we often understand as musical parameters, are mental categories created to explain musical phenomena. As such, they help in describing music and they serve as good tools in separating the different "dimensions" of music. We might say that musical parameters are factors that constrain the creation of music, or put in another way, constitute the variable dimensions in music. What should be regarded as musical parameters might lead to much discussion, but I have chosen to include melody, harmony, rhythm, tempo, loudness, dynamics and timbre here. Many other concepts could have been mentioned, but I believe that these concepts embrace the tools necessary for talking about music. This is in accordance with how Meyer (1989: 209) talks about musical parameters in a discussion on musical style[22]. He further subdivides the concepts into two groups:

- Primary and syntactical: melody, harmony, rhythm
- Secondary and statistical: dynamics, tempo, timbre

Meyer explains that calling the parameters primary and secondary do not say anything about the importance of the parameters in the aesthetic experience of music. This division is rather meant as tools for grouping. The reason he calls the primary parameters for syntactical, is because they can readily be segmented in constant and proportional ways. By this I assume he means that notes are discrete, and well-defined, events in traditional notation. Meyer argues that these syntactical parameters are based on syntax, by which he refers to a series of stimuli related in such a way that a feeling of mobility and closure is established (Meyer 1989: 14).

The secondary and statistical parameters are those that can be described in amounts. Meyer argues that they do not have the same syntactic capacity of creating closure, but are rather uniform and constant throughout a piece. If I understand Meyer correctly, he means that these parameters do not change the *form* of a piece, only the quality of the primary parameters. That is, our perception of a piece with a certain melody, harmony and rhythm would not change considerably if any of the secondary parameters were changed.

I think parts of Meyer's argument for dividing the parameters so rigidly into two groups, is dismantled when he states that a parameter can, indeed, have a syntactic function in one style and a statistical in another (Meyer 1989: 14). For example, harmony was regarded a

---

[22] Meyer actually divides the rules governing style into a hierarchy based on three classes: laws, rules and strategies. In this system, the musical parameters fall into the category of laws, and he suggests that they are "transcultural constraints" and thus universal (Meyer 1989: 14).

primary parameter up until the 19th century, but was often used as a secondary parameter in some 20th century styles. This shows some of the problems associated with such a way of looking at the musical parameters.

While Meyer's view seems suitable to western, classical music from 1650-1900, I think it would be better to look at the parameter's application in music. I therefore suggest the following groupings:

- Melody and Harmony
- Rhythm and Tempo
- Dynamics and Loudness

I choose to relate melody to harmony since both these concepts are focused around the placement of notes in time and space. Notice how the concept of *note* is used as the constituent of melody and harmony, since in Section 1.4 note was defined as a theoretical term denoting a specific pitch, while tone was meant to be a note with an associated timbre. I believe this distinction between mental categories and perceptual entities is quite significant, since when talking about melody and harmony in a traditional context they usually refer to relationships between notes in a score. Melody is the horizontal, or sequential, movement of notes in time, while harmony is the vertical, or simultaneous, placement of notes (see Figure 26 for a clarification of dimensions). These concepts are therefore often regarded as two totally separate parameters. However, from a perceptual point of view it is quite clear that they often coincide, for example in passages where overlapping melody tones are heard as a chord. Actually, some musical styles, for example the fugue, are built around exactly this interaction between melodic and harmonic lines.
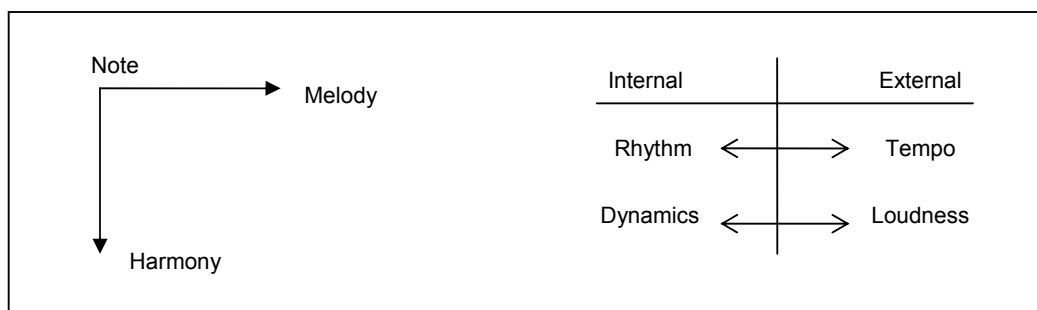

Figure 26. Groupings of musical parameters based on function in musical time and space.

When it comes to the other parameters mentioned above, I choose to regard them as either internal or external (see Figure 26). The internal parameters govern the inner relationships of music, for example the time distance (rhythm) or the volume difference (dynamics)

between notes. The external parameters inflect the overall quality of the music, either the speed (tempo) or volume (loudness).

Rhythm is a phenomenon based on the relationships between events in time. Such events can be either changing or repeating notes or chords, so rhythm concerns both melody and harmony. The rhythm is very important for our ability to organize musical events throughout a song, while the concept of tempo is merely a quantitative parameter telling us the beats per minute. The same can be said about the difference between dynamics and loudness, the first affects the internal relationships, the latter the overall quality. Thus changing either loudness or tempo will not inflect the internal relationships of the music, except of course in extreme cases where otherwise separate sounds are fused and/or masked. Rhythm, on the other hand, will not change if the tempo is increased within reasonable limits, nor will dynamic differences (i.e. pp - mf) change if the overall loudness is changed.

I believe this distinction to be relevant because it seems that the external parameters are easier to find from the sound signal. Loudness, for example, can be determined from the amplitude of the signal, even though there is not a linear relationship between physical amplitude and perceived loudness. Tempo can be found relatively easily by looking at regular attacks in the sound. The other musical parameters (melody, harmony, rhythm) are mental categories and are thus not so easily defined in acoustical terms.

To summarize the ideas presented here, I decided to gather the musical parameters and their perceptual and acoustical qualities in Table 1. Here also timbre is included, a parameter I think is somewhat different from the others. Timbre is an important constituent of the sound of music, and I will discuss it in more detail in Chapter 5.

Making this table I realized just how difficult it is to separate many of the concepts. They all seem to overlap and interfere with each other in some way or another. A reason for this is probably that they are all based on grouping of frequencies in either time or space, and can therefore not be easily defined in acoustical terms. This, of course, is because there is still no satisfactory method for doing auditory stream segregation.

This interference and complexity of the various musical parameters is a reason for not studying them separately. Just think about how much changes in tempo and loudness affect the timbre of instruments. A flutist playing a piece by Mozart very slowly and quietly will produce a totally different sound than when playing quickly and loudly. As we shall see in the next sections, we will benefit from acknowledging the mutual importance and concurrence of all the musical parameters.

Musical parameter	Perceptual quality	Acoustic quality
Melody	Tones arranged in musical time.	Sets of frequencies that follow each other in time.
Harmony	Simultaneous sounding of tones. Chord.	Sets of frequencies that occur at the same time.
Rhythm	The grouping of tones in time.	Time difference between internal events.
Tempo	The overall speed of the piece.	Beats per minute.
Dynamics	Gradations in volume between internal elements.	Local changes in amplitude and harmonic content.
Loudness	The overall level/volume of the sound.	The overall changes in amplitude and harmonic content.
Timbre	The sound of an instrument. The quality that distinguishes one sound from another.	Sets of frequencies changing over time.

Table 1: Musical parameters and their perceptual and acoustic qualities.

4.7   Salient Musical Parameters

The analysis of short excerpts in Section 4.3 showed that there is no simple answer to which musical parameters are most important in our ability to recognize musical content. Actually, I think that any parameter can be salient, either in itself or working together with others. In this section this will be illustrated with some musical examples covering salience points from many different parameters. The focus will be on illustrating the multifaceted existence of salience, rather than detailed musical analysis.

Melody is often considered to be one of the most important aspects of music perception. A reason for this might be that a melody is easy to reproduce by whistling, humming or singing. Let us look at one of the most famous melodies of all time, the fourth movement of Beethoven's 9th symphony (Example 6a). In this excerpt the melodic motion consists mostly of diatonic movement in a A-A'-B-A' form, a typical "question and answer" (A-A') in the beginning, an intermission (B) and then repetition of the first phrase (A'). This way of grouping notes into subphrases of a longer melody, can be seen in relation to how our memory works. As discussed in Section 2.3, our short term memory only lasts for about 3-5 seconds, and for formal sectioning to occur in the long term memory the signal has to be repeated. That might by why so many melodies are built up by repetitions of short motifs. Concerning the melody in Example 6a, it could therefore be argued that the first motif (called A above, and consisting of the notes F-F-G-Ab-Ab) could be sufficient to recognize the whole melody. What I think is quite significant for this motif is the rhythmic grouping of the notes. There is a very distinct emphasis on the first beat of each

bar, enhanced by the repetition of notes (F-F and Ab-Ab) at these places. This extends throughout the whole section, and I would therefore argue that salience in this melody is created by diatonic movement, repetition of motifs and phrases, and finally accentuation of first beats by rhythmic and monotonic grouping.

Some of the same could be said about the melody in the second movement of Dvorak's 9th symphony (Example 6b). It is built up by small intervals (F-Ab), and as the Beethoven melody it also has a structure of repeating phrases. But as opposed to the Beethoven example, the rhythmic grouping of the notes is less significant, and the tones rather seem to "float" slowly around. I once overheard a conversation during a performance of this movement, and it was interesting to hear that "the melody feels so slow". This "slowness", however, was not due to the performance but rather seems to be a feature of the melody. A reason we might perceive this melody as slow or "stretched out" is the fact that the length of the phrases lie close to the limits of our short term perception. This might create a tension, expectation, or the feeling of "waiting" for the next tone. If also the performance is slow, it might actually be problematic to make formal sections, since the phrases vanish from our short term memory before the next repetition. Another interesting aspect of this example is the importance of the underlying chords, especially towards the end. Even though it might not be decisive when it comes to recognizing the song, I believe that the timbre of the instruments and the harmonic content is important for the overall experience of this song.

An example of salient harmony is the intro of Stevie Wonder's *You are the Sunshine of my Life* (Example 7). Probably also rhythm and sound plays a part in recognition of this excerpt, but I believe the most significant feature is that of the chord structure. The sequence of intervals based on a dominant seventh chord with augmented fourth and fifth is quite rare, and immediately interesting.

Probably few motifs in the history of music are more salient than the opening of Beethoven's 5th symphony (Example 8a). It is very short and it immediately grabs our attention. I would argue that this is a prominent example of musical salience mostly because of the rhythm. The effect of this motif might be explained from what could be called an ecological point of view, due to its resemblance to the natural sound of knocking something. That might also be why it is often called the "Hammer motif". Anyway, the motif is very easy to learn and to recognize. The fact that it is also repeated throughout the symphony in various melodic disguises, for example in the 3rd movement (Example 8b), helps in establishing it as a leading motif.

An example of salient dynamics is the opening of Grieg's A minor Concerto (Example 9a). It starts with rumbling timpani increasing to a massive wide-spaced opening chord. Lasting barely more than 3 seconds and with only one attack and not much of melody, harmony or rhythm, it is still very powerful. Looking at a time-domain plot of the sound (Figure 27) reveals the gradually increasing dynamic shape clearly. It resembles an exponential function, and this accelerating loudness could be perceived as a gradually increasing tension before climax is reached. It is also interesting to notice how this excerpt sounds similar to a "reversed" attack, for example an inverted piano tone (Example 9b).
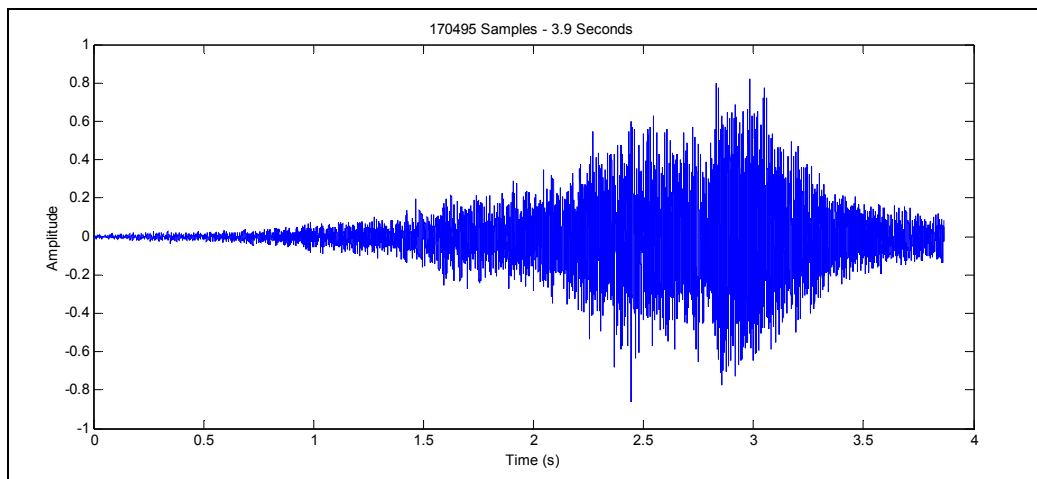
Figure 27. Time-domain representation of the opening chord from Grieg's A minor Concerto.

I also think that the sound in this Grieg excerpt is significant for our perception. We can hear the domination of low frequencies in the beginning, and as the loudness increases the sound gets brighter. In the "attack" at around 3 seconds, there is much energy all across the spectrum. This can clearly be seen in a spectrogram (Figure 28), where most of the energy in the beginning is in the lower frequency range. This corresponds well to our perception of the low-frequent timpani. Then as the volume increases, so does the frequency content. The attack is clearly visible, with much energy all over the spectrum. It is interesting that even though the opening of Grieg's A minor Concerto is played by many different instruments, and the sound changes considerably over the 3 seconds that the first attack lasts, we perceive it as one coherent musical figure with a changing timbre.

170495 Samples - 3.9 Seconds
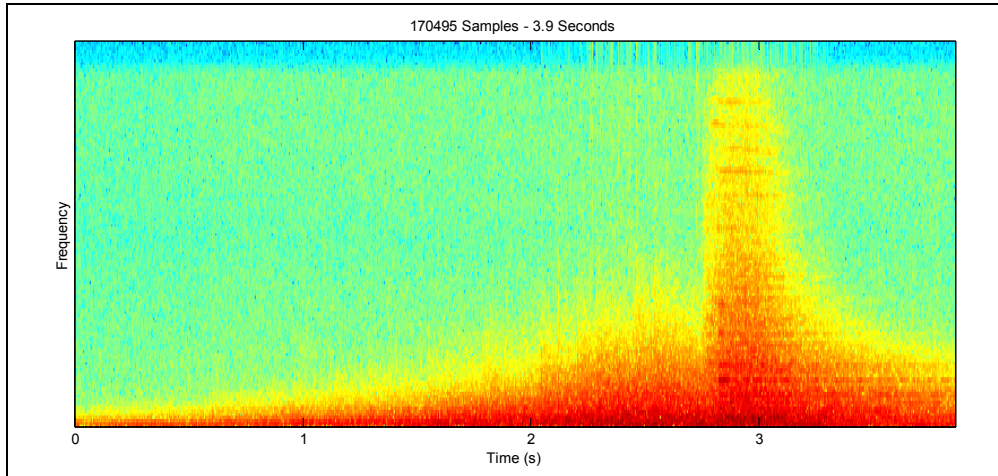
Figure 28. Spectrogram of the opening of Grieg's A minor Concerto. Notice how visible the attack (at approx. 3 seconds) is in the plot.

In the preceding examples, I have tried to show that many of the musical parameters can be salient in themselves. But more important I think it is to evaluate how the parameters act together, and how this allows for a great deal of flexibility. We might for example change the chords of a song and still be able to recognize it from the melody. On the other hand, some tones in the melody could be changed or added, while the rhythm would help in recognizing the song. This flexibility makes it possible to recognize altered versions of a song.

One of the most extreme examples of such musical flexibility is that of jazz. Two versions of a jazz song can be very different, but still we will be able to recognize them. But if everything, including performing the instruments, chords, rhythmic figures, and even melody is so changed; how do we recognize the song? The answer might be that there is "something" in the song that people recognize. I believe that exactly those moments of "something" are the salient points, and that these short excerpts might form the basis for our perception of the entire song. This will be outlined in Section 4.9, but first some examples of salience based on timbre will be presented.

4.8   Timbre Salience

The previous section showed that the various musical parameters can contribute to salience. Quite clearly, though, it seems like the sound of the music always plays a part in music perception. This section will present some examples where timbre, in the sense of being the main ingredient of sound, is salient in itself.

First I will just briefly present some examples of one of the most characteristic instruments in the world, the human voice. Our perception seems to be "tuned" to make us quickly recognize voices, not only voices of people we know well, but also voices of famous people we hear on radio or TV (Sundberg 1999). When it comes to recognizing music, our perception of the timbre of voice is therefore very important. An illustration of how easy it is to recognize different voices is the song *We are the World* (Example 10a). This song was recorded by many of the most popular singers in the late 1980s, and features an amazing mixture of voices. Other examples of voices that are particularly salient are those of John Lennon, Bob Dylan, Louis Armstrong and Ella Fitzgerald (Examples 10b-e). Even though these examples contain salient features also in melody, harmony and rhythm, I think the timbre of the voices might be a decisive factor when it comes to recognizing the music.

Sundberg (1999) argues that the timbre of a voice is governed by, amongst others, the amount of energy in the first harmonic frequencies (about the first 4 harmonics) of the sound. Moving more of the energy from the fundamental frequency to the next harmonics, makes the sound richer. The singer's formant was mentioned in Section 2.4, and it refers to how singers (especially basses, tenors and altos) add more energy to the frequencies in the sensitivity region of the ear, so that the voice is perceived louder. Also important is the use of vibrato, and its regularity or irregularity.

The voice is considered a very "personal" instrument, quite opposite of what could be said about the piano. Since the piano sound is created by hammers hitting the strings, this might lead people to believe that a pianist has little control of the timbre. The reasoning behind this is that the pianist seems to have less control over a tone than musicians playing instruments that create sustained tones, for example a violin. However, the claim does not take into account that pianists have different attacks and the possibility to control the resonance of the sound with pedals. Also, some famous pianists demand to play only on certain brands of instruments, and this might also help in making a specific sound. In the following examples I will show how piano timbre can contribute to a very salient overall sound.

Example 11a is taken from Mozart's Piano Concerto K488. The instrument sounds warm and subtle and the pianist is also playing the attacks very gently and *legato,* assuring an overall "mild" piano sound. This stands in hard contrast to Example 11b, an excerpt from Stockhausen's *Klavierstücke I*. Even though the instruments of these two examples might very well be of the same kind, they certainly sound differently, mostly due to the pianist's attack.

Interesting in the Stockhausen example, is the use of a grace note[23] before the second main tone (Figure 29). This grace note is considerably softer than the following tone, and since there is some duration between the two, we can hear both attacks clearly. The decay of the grace note, however, is masked by the loud attack of tone 2, but even though we cannot hear it separately I believe that the grace note colours the other tone. When listening to such an example, I think the grace note and tone 2 are perceived as a coherent entity quite different from the single tone 3.
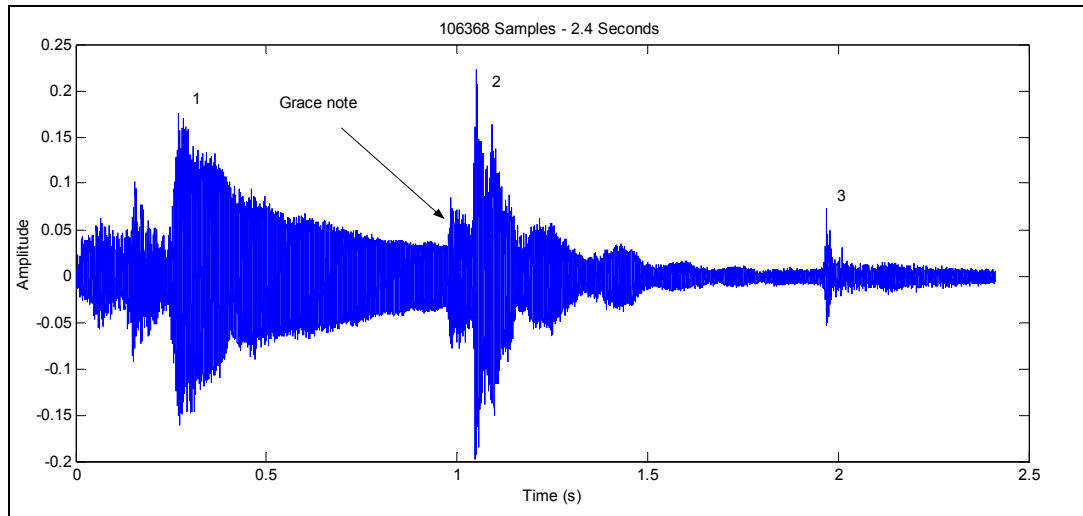
Figure 29. Time-domain representation of excerpt from Stockhausen's Klavierstücke I. Notice the grace note before tone 2.

The Mozart and Stockhausen excerpts are obviously quite different, both in melody, harmony and rhythm. But, as mentioned above, I also think that the piano sounds very dissimilar in the two excerpts. It might be unfair to compare a soft part of a classical piece with an excited part of a 20th century piece. Anyway, we would not expect Mozart to sound that brutal just because there is an "unwritten" notion of how the instrument should be used in different styles. I also believe that the way Stockhausen uses grace notes to colour other tones, would not likely occur in for example a piece by Mozart. So such a detail as a grace note might be crucial for our recognition and perception of the music.

A quite different example is Debussy's *Pour remercier la pluie au matin* from *Epigraphes antiques* (Example 11c). The impressionists often used techniques trying to avoid sharp contours by creating layers and "blurred" patterns. In Example 11c this is achieved on the piano by the generous use of pedals and overlapping tones. The background layer consists

---

[23] As defined in Chapter 1.4 I prefer to use *tone* when referring to the sound of a *note*. However, since "grace note" is a well defined musical concept, I use that also when referring to its sounding quality.

of a tremolo, and due to the sustain pedal this may be perceived as one long, coherent tone with a changing texture, rather than separate tones. The result sounds quite differently from both the Mozart and Stockhausen examples.

The next excerpt is from *Part III* from *The Köln Concert* by Keith Jarrett (Example 11d). Recorded on a mediocre-quality Bösendorfer, this is perhaps one of Jarrett's most popular recordings[24]. In his solo performances, Jarrett often plays in a "fixed" position, or key range, for longer periods of time, with slow development by gradually expanding the rhythmic figures and experimenting with minor/major fill-in tones. Although Example 11d is short, is still shows some of this more "jazzy" approach quite unlike the classical examples above. This is also audible when it comes to the "laidback" use of grace notes. I think the overall sound is "compact" but still with rapid internal motion. So in this case, the combination of a particular piano timbre, and distinctive playing, results in a sound quite different from the much "harder" and more precise Stockhausen excerpt, but not as "blurred" as the Debussy example.

Thelonious Monk is considered one of the most distinguished jazz pianists of all time, much because of his special sound. An important aspect of the sound is the use of pianos with timbral qualities more in the direction of "Honky-Tonk" than the classical "Steinway sound". But he is also famous for consequently playing "misplaced" accentuations and the simultaneous minor second intervals. This helps in creating some of the "blue" feeling on a piano. All of this is clearly audible in Example 11e, an excerpt from *Blue Monk*.

The final piano example is from John Lennon's *Imagine* (Example 11f). In this case, it seems like there has been added some slow reverb on the piano part, thereby creating a characteristic and wobbly timbre.

Analysing piano sound is both challenging and interesting[25]. As I have tried to show in these examples, it involves analysing both the timbre of the instrument, but also the way the instrument is played. Different types of attack, use of pedals and harmonic textures all add up to give a quite distinct sound. Thus it is possible to recognize a particular song, composer or pianist from listening only after the overall piano sound.

---

[24] Jarrett was not happy with the instrument and described it as "a seven-foot piano which hadn't been adjusted for a very long time and sounded like a very poor imitation of a harpsichord or a piano with tacks in it" (Carr 1991: 71).

[25] See (Jensenius 1999) for a more detailed discussion of piano sound.

4.9 Salience as the Basis for our Thinking about Music

The discussion of how we perceive salience through time can also be extended to the problem of how we think about music. This again should be seen in connection to the limits of short term memory, both in terms of time and content. The reason for this is that any stimulus that fits within the boundaries of short term memory will have an "advantage" because it can be grouped in its entirety.

Then a relevant question is how our mental auditory images are updated in relation to the continuous auditory input. Godøy (1999) writes that it is likely that our perception works by "windowing" in a way similar to how digital signal processing works, and as suggested by Husserl's model with the subjective "now" (see Figure 2, Section 1.4). If this is the case, is there a continuous updating like in Figure 30a, a discontinuous and intermittent updating like in Figure 30b, or a combination of the two, as shown in Figure 30c? I think a model like the latter is more likely, with a continuous updating of the auditory image and intermittent "snapshots" forming the basis for our higher level thinking of music. This way our short term memory could be seen as a buffer, constantly receiving information on one side and shuffling out information on the other. Furthermore, the "snapshots" perceived along the way might correspond to the salience points in the music.

There is a basis for such a model in the idea that we think about music not in terms of single events, but rather in *holistic* streams. Opposite to atomism, holism looks at phenomena as coherent entities. Godøy (1997b) argues that it is important to recognize musical objects as holistic entities, instead of separate notes or chords. This follows the ideas of Schaeffer (1966) on the morphology of the s*ound object* (*l'objet sonore*). This is not the notated score, physical signal, sounding body or a state of the soul, but rather a phenomenological sound formation, and primarily independent of its referential qualities as a sound event. Schaeffer's idea was to characterize the "surface" qualities of the musical object, and he made a matrix where various metaphors such as for instance mass, grain and gait denote features of the musical object[26].

---

[26] For an English translation of Schaeffer's matrix, see (Godøy 1997a)
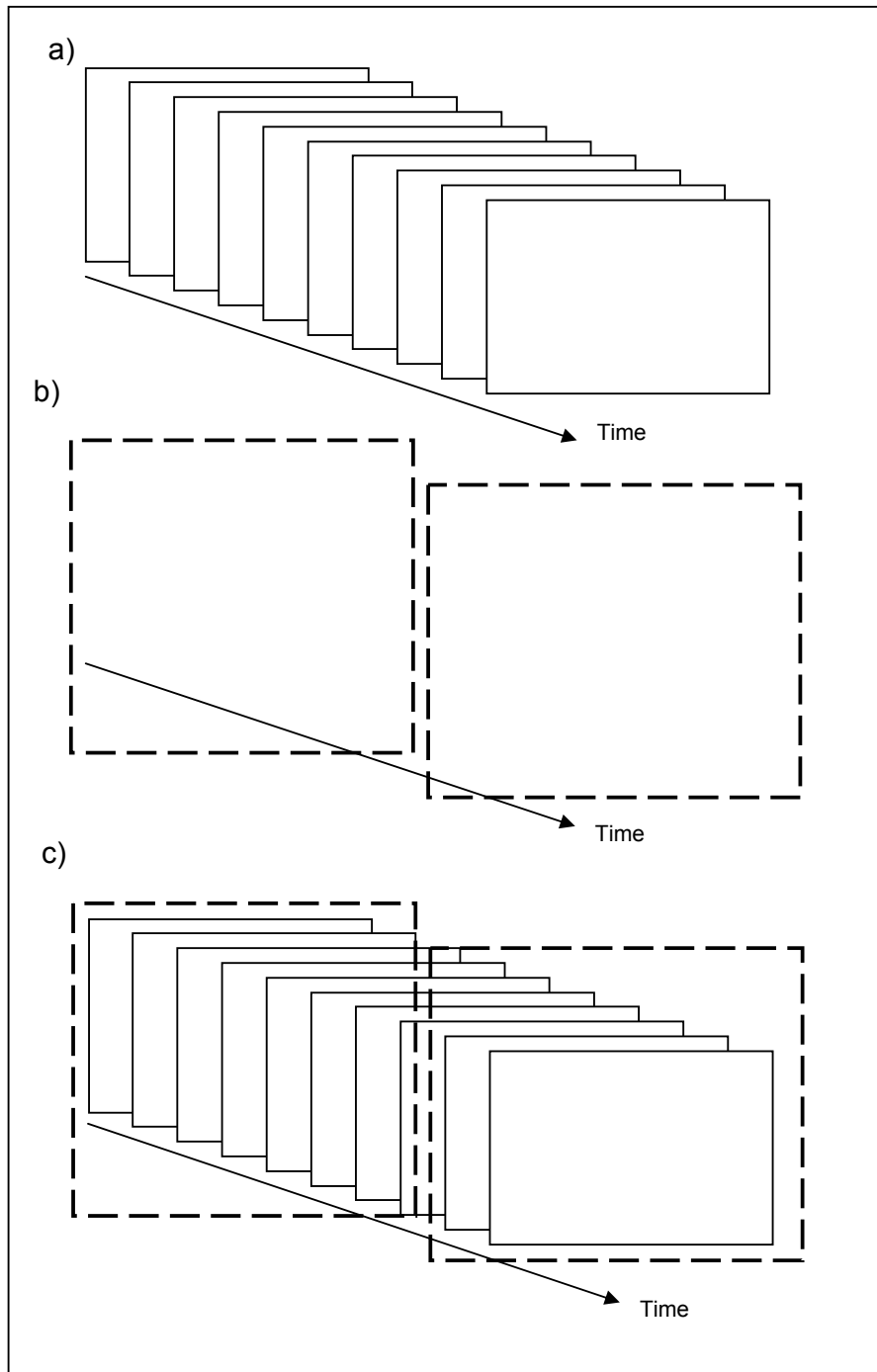
Figure 30. Is the auditory image updated a) continuously b) discontinuously c) a combination? Based on (Godøy 1999).

This leads Godøy (1997b) to suggest the term *shapes* as a representation of musical objects as holistic entities. Such shapes are highly multi-dimensional, with various emergent qualities such as timbre, texture and contour, and they may form the basis for our mental imagery of music. I think that understanding mental imagery as shape cognition corresponds well with the idea of perceptual salience. Actually, a salience point might be

seen as an important factor of the shape, or in some cases even be equal to the shape. Imagine a multi-dimensional space with a timeline and a musical shape, like in Figure 31, where a certain peak dominates the shape.
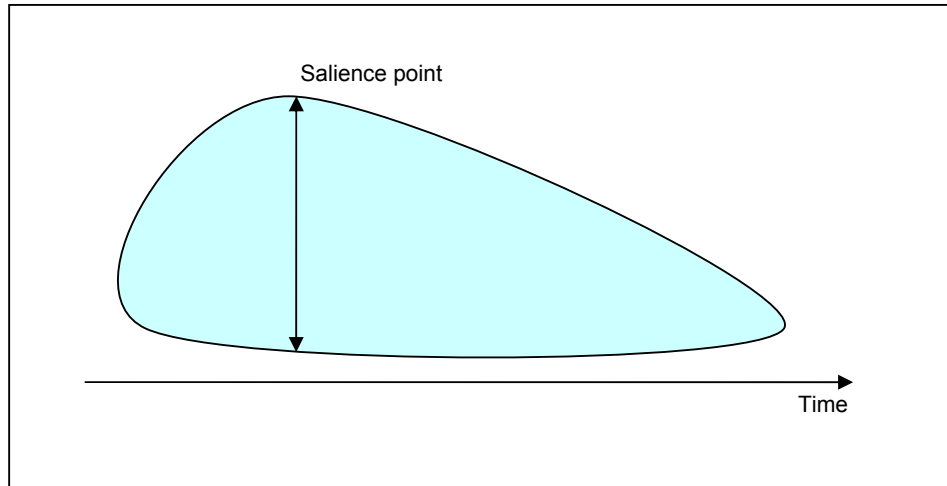
Figure 31. A multi-dimensional shape might be dominated by a salience point.

I believe that a salience point can be such a peak or "centre of gravity" of any shape, or the factor governing the form of any shape. Thus an excerpt of a salience point could help in revealing the whole shape, and it should also be possible to make a "music trailer" based on salience points.

4.10 Music "Trailers"

Watching movie "trailers" I have always been fascinated by how much information seems to be packed into a very short sequence. Somehow it should be possible to make music "trailers" or "thumbnails" in a similar manner. I therefore experimented a little bit with different methods of doing this.

First, I thought of doing a time compression of a piece that would preserve the overall form and also some of the timbral qualities. A test of this was done by compressing Ravel's *Bolero,* with a phase vocoder, from 15 minutes to 15 seconds (Example 12). Even though it might be possible to get some understanding of the musical content and changing timbral qualities from this example, such a method hardly gives a good representation of what I think is perceptually important in this piece: rhythm, melody and timbre.

A better way might be to cut out short pieces of the entire piece. Since it has been argued that short excerpts are perceptually significant, such a collection of short segments could tell a great deal about a song. Trying to do this automatically I made the patch *Music*

Trailer that plays certain segments of a song. The interface (Figure 32) allows the user to choose the number of segments that shall be played in the song, and also the window size, or the duration of each musical segment.
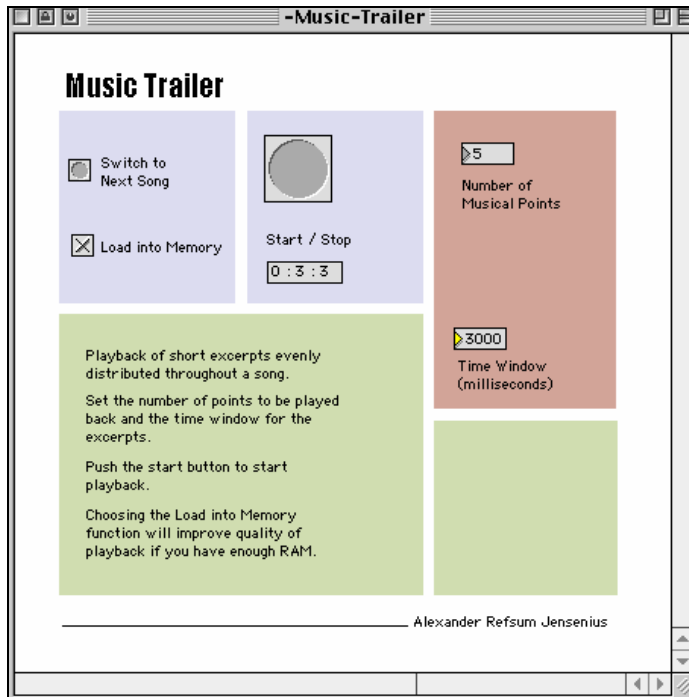
Figure 32. User interface of the patch Music Trailer *that plays short excerpts from a sound file.*

The main technical feature in the patch is to find the length of the song and calculate the evenly distributed segments to be played. This is accomplished with a series of subpatchers as shown in Figure 33. The playback points are found simply by dividing the length of the song by the number of points assigned. A *metro* object keeps track of how much time is played, and skips a corresponding amount of samples after the set time.

Since moving from one point to another in a large sound file requires some computer processing, there might be some hiss or interruptions in playback. I tried to implement some feature of preloading cues, but this did not work well. After some trial and error, the best results were achieved with the *loadintoram* function. This reads the entire sound file into memory reducing the problem somewhat, but requires a larger memory block dedicated to MAX/MSP.
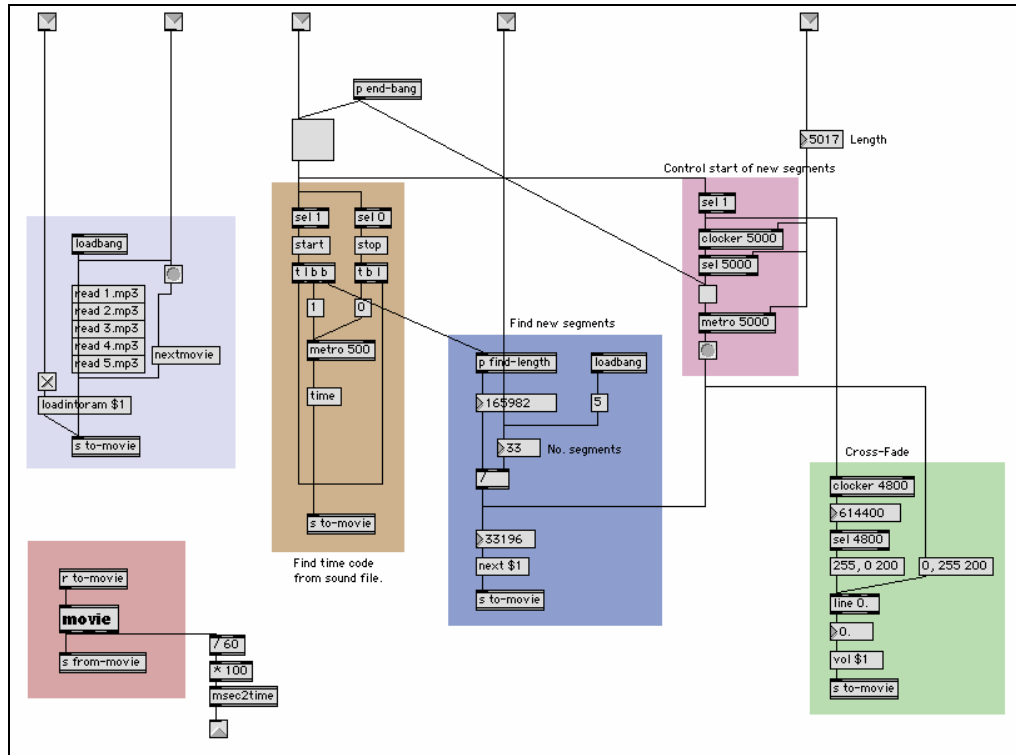
Figure 33. The inside of the Music Trailer *patch.*

The reader can test the application, or refer to Example 13a and 13b that are ready-made "trailers" of *Bolero* with either 5 excerpts each lasting 3 seconds, or 3 excerpts each lasting 5 seconds. These examples show that it is possible to hear some of the rhythmic figures, melody and timbre. Even though the patch selects segments without any knowledge of musical content, I would argue that the result is more relevant than the compression example presented above. Trying to play pop songs with the patch is also quite interesting, since it reveals how remarkably little the overall sound changes from one part to the other. In some cases, it is actually difficult to hear that two excerpts following each other might be several minutes apart. The problem with this approach, however, is that the program does not know anything about the musical content, and this lack of musical "knowledge" might result in missing out on perceptually relevant points.

Quite clearly, the best "music trailer" would consist of salient features. Unfortunately, since there is no way of doing this automatically, I made some test samples of *Bolero* manually (Example 14a and b). The interesting thing to notice, is that these two examples capture some of the structure of the entire song and parts of the melody while still preserving timbral qualities.

4.11 Conclusions

This chapter started with a model of music recognition by Peretz (1993), where she suggests that melody is the most significant element for recognition. The problem is that her focus is only on western melody-based music, and the model is thus quite limited. I therefore turned to investigate short term recognition, first by an example of how MAX/MSP can be used in measuring recognition time of a song. My informal tests show that it is indeed possible to recognize music from excerpts lasting less than 3 seconds. Exactly how long it takes to recognize a song seems to be dependent on the musical content, but also whether the song is played from the beginning or at a random starting point. In general I would assume that we recognize the introduction of songs more easily. This might be because many famous songs start with the most salient features (such as Eric Clapton's *Tears in Heaven*). In the example from *Toy Story* it was difficult to recognize the song because no uniquely significant features occurred in the first 30 seconds. This further supports my hypothesis that salience is crucial for our recognition of music.

Then perceptual salience was discussed and it was suggested that it is often caused by unexpected events or sudden and large changes. When it comes to how we actually perceive salience, a patch that measures salience from user input was presented. I leave it as an open question whether salience is perceived as a continuously updated stream or as discrete points in time. Quite clearly, though, most musical parameters (melody, harmony, rhythm, tempo, dynamics and timbre) can all contribute to salience, either alone or together. I believe that a traditionally strict separation of these parameters might not be fully in accordance with our perception of the acoustical attributes they refer to. It is important to acknowledge their mutual relationships and effect on each other, and this was elaborated with examples of how each parameter might contribute to salience in music. From this discussion it was argued that the overall sound of music seems to be an important factor when it comes to music recognition, and examples of salient voices and piano sound were shown.

Following these ideas, it was suggested that salience can be used as the basis for our thinking about music. This is rooted in a holistic way of looking at music as multi-dimensional shapes. Salience points may be a "centre of gravity" in such shapes. This led to a final example of how a music "trailer" based on salience points can effectively give an overview of a song.

5   Sound and Timbre

This chapter will focus on how we can analyse, visualize and synthesize sound, or more specifically the timbre of instruments.

5.1   Pitch and Timbre Perception

As presented in Chapter 3, our perception of music is based on the grouping of frequencies in time and space. That is why a set of frequencies can be heard as a specific tone with an associated pitch, loudness and timbre. Such grouping is done by relating frequencies that have their origin close in spatial location, have similar onset time, and move in the same direction. The problem, however, is that there are no computational tools that can do this in an immediate and straight forward way like the human brain.

There is not even an easy way for computers to find the perceived pitch from a set of frequencies. Usually the pitch is associated with the lowest frequency, but this is not always the case. Sometimes we can hear a pitch that is not physically present in the sound. Such cases of a "missing fundamental" may be caused by filtering, for example if the sound had to travel through a wall or a device that does not carry the lowest frequencies, for example a telephone or radio. In these cases, the fundamental is usually reconstructed in our hearing from the spectrum, i.e. we will tend to "find" the pitch from the harmonics. This way we can still enjoy music and have a perception of the correct pitch.

Similarly, it is also difficult to handle a concept such as timbre by computational tools. Despite our excellent ability for discerning and recognizing timbre, it still remains to find an easy way of describing it, both in normal language and in physical terms. Usually we refer to the sound source to describe the timbre, e.g. "piano-sound" because it was created by a piano, but there is for instance no simple way of finding "piano-sound" from a sound file.

A reason why it is difficult to analyse timbre using physical and mathematical terms, is probably because it depends on so many parameters. In a classical experiment, Grey (1977) showed how 16 different instrument timbres can be categorized in a simplified and three-dimensional timbre space. The three dimensions of this space were:

- Axis I: *Spectral energy distribution*. This gives sounds ranging from dull to sharp. For example, the French horn is an instrument with a dull sound, while the oboe is much sharper.
- Axis II: *Synchronicity in harmonic transients*, and decay of upper harmonics. This is related to the spectral fluctuation through time. Woodwinds have upper harmonics that enter, reach maxima, and exit in close alignment. The strings are on the other extreme and have harmonics which do not have such synchronicity.
- Axis III: Amount of *high-frequency inharmonic noise in the attack*. Strings, flutes and clarinets have high-frequency, low-amplitude, and inharmonic energy in the attack, while brass and bassoons have low-frequency inharmonicity and no high-frequency energy in the attack.

Grey's results were used by Wessel (1979) when he showed that Grey's timbre space could be used as a control structure. Wessel operated with two axes, one with the spectral energy distribution controlling the brightness of the tone, and another with the frequency transients in attacks controlling the "bite" of the tone. The problem, however, with both these approaches is that they are based on psychological experiments on a limited number of instruments. As such, they do not represent methods that can easily be used for classifying timbre directly from an acoustical signal (Cosi, De Poli, and Lauzzana 1994). To exemplify some of the problems related to timbre analysis and recognition I will look at various ways of analysing a saxophone tone.

5.2   Analysis of a Saxophone Tone

Example 15a is a 2,7 second saxophone tone played by Sony Rollins. A subjective description of the sound might be that it is a single-pitched tone with slight changes in timbre and loudness towards the end of the sound. How can this be related to the physical signal?

Figure 34 shows a time-domain plot of the sound and we can see that there is clearly a decrease in the amplitude. In this case the amplitude curve fits well with our perception of the tone, also the slight decrease down to a sudden "fall-off" at around 1.7 seconds. Even though there is not a one-to-one correspondence between the physical amplitude and our perceived loudness of a sound, the plot matches our perception quite well.
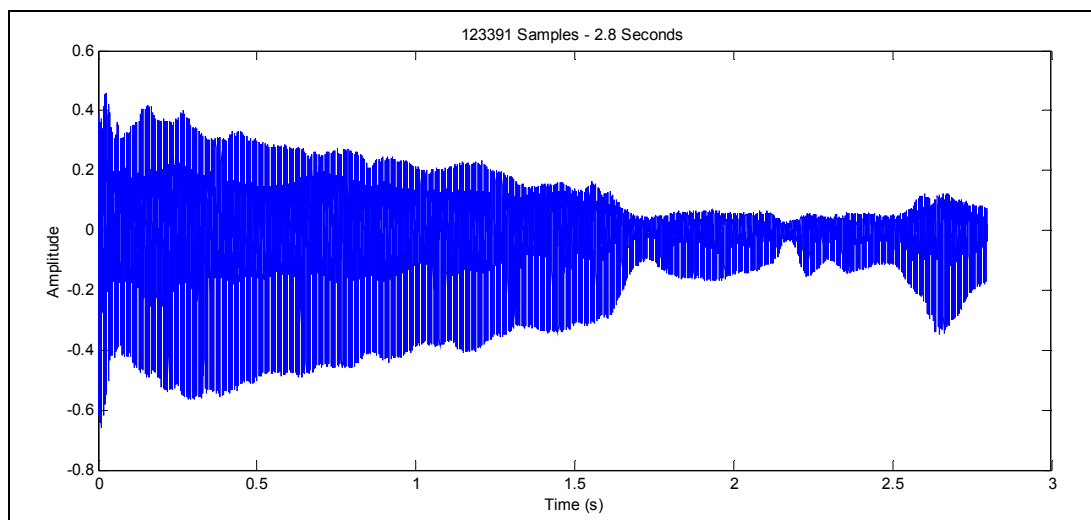
Figure 34. Amplitude-domain representation of the 3 second Sony Rollins saxophone tone.

An automatic pitch extraction from this sound, using Addan 3.0[27], is shown in the plot in Figure 35. It is interesting to notice that when listening to the example and comparing our sensation of pitch with the measurements here, we notice that the software has found a pitch one octave lower than the perceived pitch of Ab4 (approx. 420Hz). That is a quite common "mistake" done by pitch trackers, since most algorithms only look for the lowest frequency in the sound.
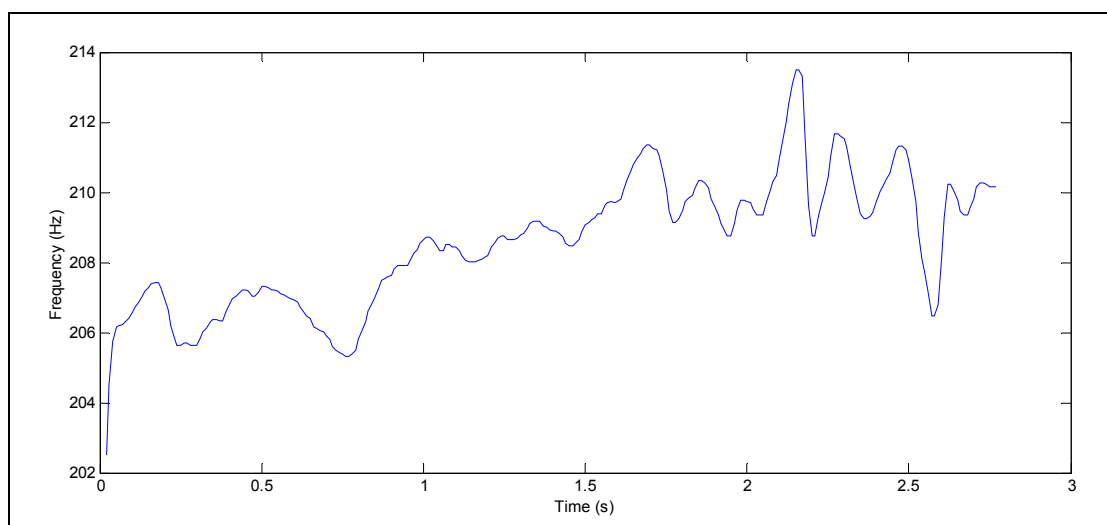
Figure 35. Estimate of the fundamental frequency of the Rollins tone. The y-axis is cropped for the sake of clarity. Notice how much the frequency changes.

---

[27] Addan is part of the Diphone Studio, distributed through the IRCAM Forum.

The second interesting thing to notice is how much this estimated fundamental frequency (F0) changes throughout the plot. The difference between the lowest and highest value is almost 12 Hz, and that corresponds to a difference larger than a semitone. There seems to be a local peak at around 1.7 seconds and a global peak at around 2.2 seconds. When listening to the tone, I do think that the pitch changes slightly, but not as much as is suggested in Figure 35. The reason for this might be that the pitch extraction and/or our perception is "imprecise".

A reason for the "problems" with such pitch extraction might be found by investigating the rest of the frequencies we hear. Let us first look at a plot of the amplitudes of each of the harmonic frequencies (Figure 36). As can be seen, the harmonic close to 800 Hz is very prominent. There are also quite high values for the frequencies in the sensitivity region, so they will probably be perceptually louder than the harmonics around 200, 400 and 600 Hz. So for this tone the perceptually loudest harmonic frequencies are much higher than the fundamental.
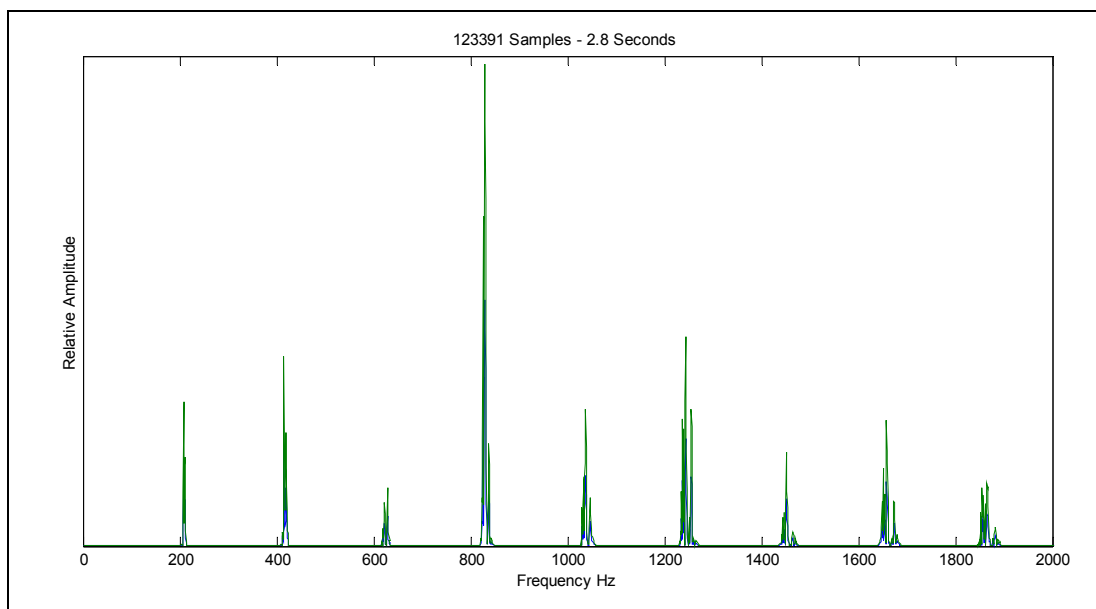
Figure 36. Plot of the spectrum of the Rollins tone. The graph shows the average amplitude for each frequency in the spectrum. Notice how each of the harmonics are clearly visible, and that the harmonic close to 800 Hz is the one with highest amplitude.

The same thing can be seen from the spectrogram (Figure 37), where there seems to be a lot of spectral energy in the higher harmonics (the darker regions in the plot). Notice also that there is more energy in the higher frequencies in the beginning of the sound. Then, at around 1.7 seconds, there is a sudden decrease of energy across the whole spectrum. This corresponds well to how we perceive the tone to become "duller" at this point.
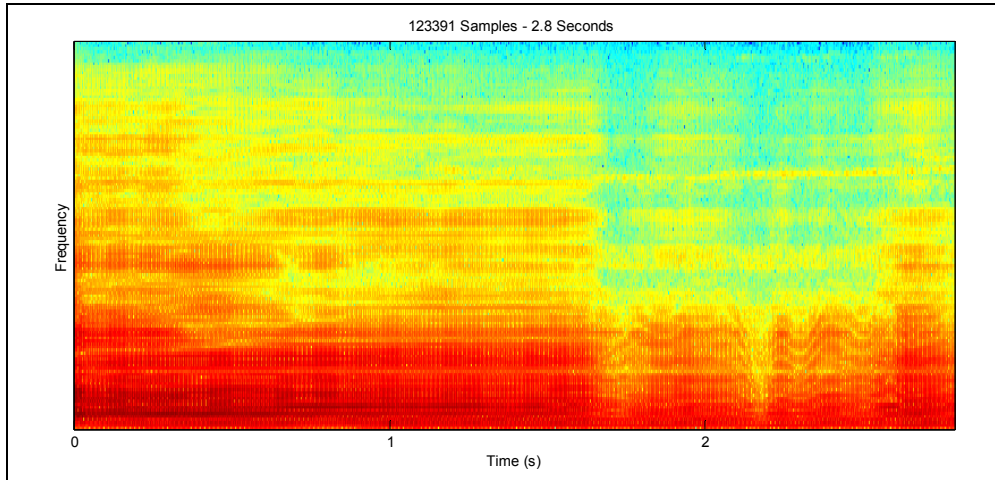
Figure 37. Spectrogram (frequency vs. time) of the saxophone sound.

As suggested by Wessel (1979), the perceived brightness of a tone is related to the spectral energy distribution. Beauchamp (1982) further suggested that the relationship between intensity and the *spectral centroid* may be an important perceptual correlate of timbre. Thus a tone which seems "brighter" has a higher spectral centroid. The spectral centroid can be found from the physical signal as the mean of the spectral energy distribution, or the "balance point" of the spectrum. This is accomplished by summing over pairs of amplitude and frequency for a given time window, where *a* is amplitude and *f* is frequency:

$$\text{spectral centroid} = \frac{\sum_{i=1}^{N} a_i f_i}{\sum_{i=1}^{N} a_i} \qquad \text{(Equation 1.)}$$
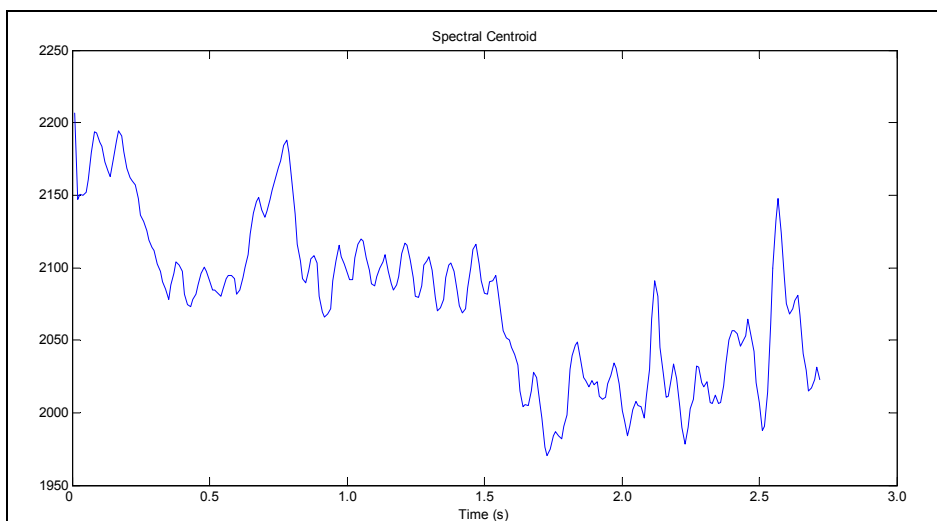
Figure 38. Spectral centroid of the Rollins tone.

Figure 38 is a plot of the spectral centroid of the Rollins tone, where more energy in the higher frequencies indicates a brighter tone. So from this graph we can conclude that the brightness should be at a minimum at about 1.7 seconds, and this fits well with what we hear.

5.3   Relevance of the Harmonics

Looking at all these graphs and plots, a relevant question might be whether we actually perceive all the information that is displayed. Do we actually need all the highest frequencies? Figure 39 shows a spectrum plot of the Rollins tone (Example 15a), and from this plot it seems that only the 15 first harmonics have relatively large values. One reason for this is that the values are plotted on a linear scale, and a logarithmic scale would have shown a less drastic curve. But anyway we could ask ourselves whether the 15 first harmonics would be sufficient to adequately describe the tone?
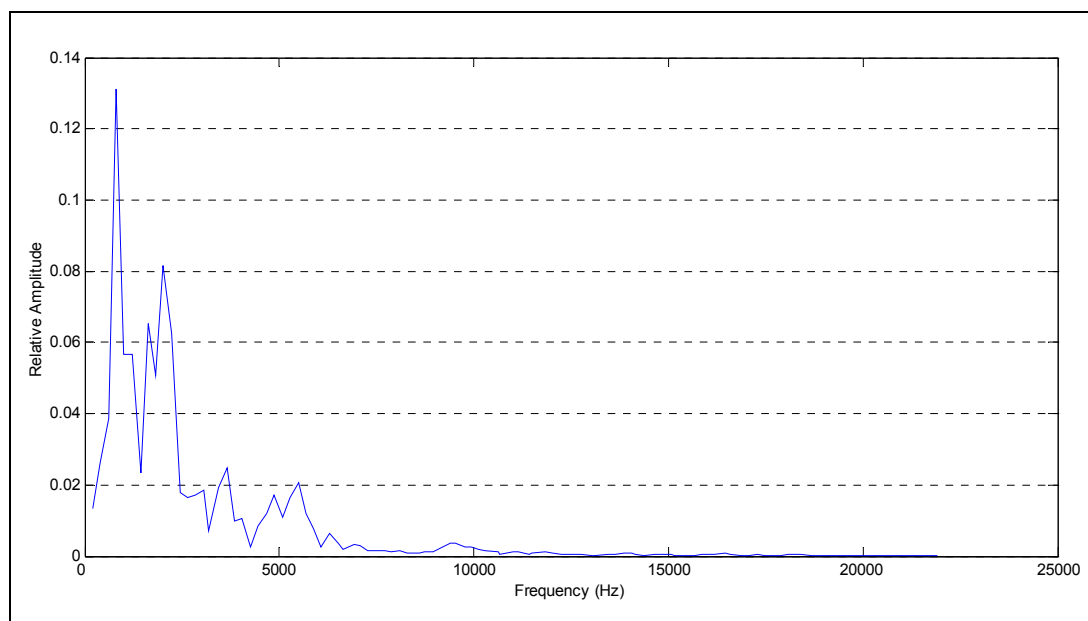
Figure 39. Linear spectrum plot of the Rollins tone, showing the relative amplitude over frequency.

The best way of testing this is by trying to synthesize the tone and see how it is perceived. So I did frequential analysis of the tone in Addan, extracting all the harmonic frequencies with corresponding amplitudes. To check the precision of the analysis, the tone was synthesized directly from the analysis values. If the analysis is good, the resynthesized tone should have few deviations from the original. Indeed, the synthesized tone (Example 15b) sounds quite like the original (Example 15a). This can further be checked by subtracting the synthesized tone from the original, and the result of this could be considered noise and

deviations. Example 15c shows that the noise is almost inaudible, and this is another proof that the analysis is good.

To test how many harmonics are necessary to adequately reproduce the sound, I made the patch *Play-Add-Files* (Figure 40). This program plays back analysis files saved in the SDIF file format[28]. In the patch the user can adjust the number of partials to be played back and the corresponding original sound file is also loaded so it is possible to check the result.

Figure 40. Screenshot from the patch Play-Add-Files *that plays SDIF-files using additive synthesis.*

The patch uses the CNMAT *SDIF-menu* to load analysis files into the *SDIF-buffer*. The *SDIF-tuples* object has a nice feature for retrieving a certain number of rows or columns of data from an SDIF-file, and in this patch the number of harmonics to be played back can therefore easily be constrained. The additive synthesis is done with the *sinusoids~* object, such as presented in previous chapters. Unfortunately, it was not possible to make a stand-

---

[28] The Sound Description Interchange Format was proposed by CNMAT in (Wright, Chaudhary et al. 1999), and tools for using SDIF-files in MAX/MSP were presented in (Wright, Dudas et al. 1999) and (Schwarz and Wright 2000). The file format allows for saving both sound and analysis information in one file. SDIF seems to be a standard with its inclusion in MAX/MSP and programs distributed by the IRCAM Forum.

alone application of this patch, due to the many specialized external objects. However, the patch may be inspected and tested if opened in MAX/MSP 4.

Selections of five different saxophone phrases by Sony Rollins are included in the patch. Some of these phrases are more difficult than others, in the sense that they contain large interval changes and noise that causes glitches and problems in the analysis. This is audible in some cases, but does not really inflict so much on the overall point I am trying to make, namely that of the number of harmonics necessary to adequately synthesize the sounds. Testing with different settings, it seems quite clear that 15 harmonics are not sufficient to give a good approximation of the timbre of the instrument. On the other hand, I would argue that choosing more than 60 partials does not give so much extra information. It is also interesting to try and play only one or two partials, and hear how the analysis sometimes has left out the correct fundamental. This can be heard in Example 16 where nine versions of the Rollins tone are presented, each with a different number of harmonics. A spectrogram showing these tones is displayed in Figure 41, and it is easy to see the reduced amount of partials.
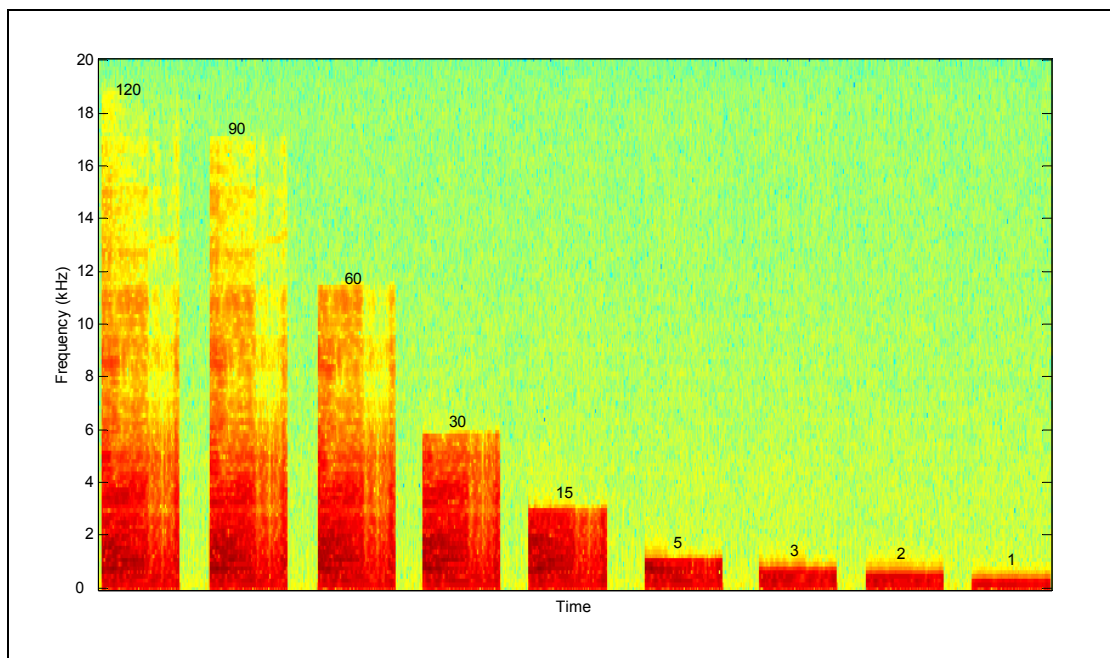
Figure 41. Spectrogram of the saxophone tone played with fewer and fewer partials. Notice how visible the reduction of harmonics in the spectrum is.

From the spectrogram we can also find the reason for our observation of the leap in perceived loudness between the tones with 5 and 15 partials. This is due to the fact that partials number 11-15 lie in the sensitivity region of the ear.

5.4   Perceptual Models

Up until this point I have presented various ways of visualizing music directly from the physical signal. However, there is also the possibility to incorporate perceptual models before displaying the signals. The IPEM-toolbox[29] for Matlab is a collection of tools for doing music analysis based on a perceptual framework (Leman, Lesaffre, and Tanghe 2001a). This toolbox uses an Auditory Peripheral Module adapted from the Van Immerseel and Martens model, involving several stages of filtering similar to how our ear works:

- Simulation of filtering in the outer and middle ear.
- Simulation of the filtering in the inner ear, using an array of band-pass filters.
- Simulation of a hair cell model where the band-pass filtered signals are converted to neural rate-code patterns.

The output of the Auditory Peripheral Module is an *auditory nerve image* of a sound, or "a kind of physiological justified representation of the auditory information stream along the VIIIth cranial nerve" (Leman, Lesaffre, and Tanghe 2001b: 18). Such a *primary image* thus represents excitation in various channels in the auditory system, and an example of how this looks for the Rollins tone (Example 15a) is shown in Figure 42. Notice how it is possible to see a decline in the energy levels through time, similar to the decrease in the spectrogram showed in Figure 37.
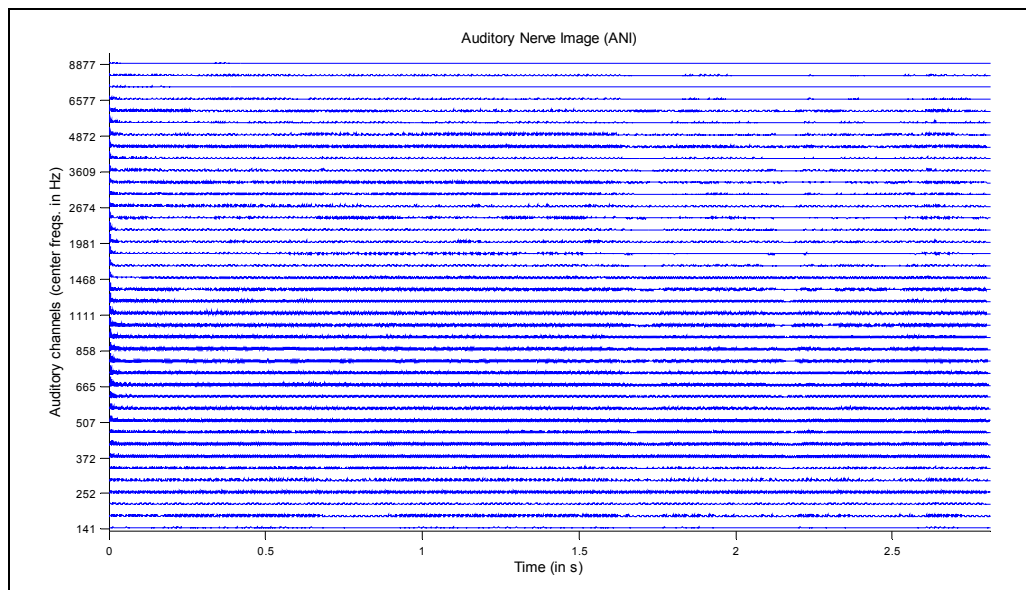
Figure 42. Auditory Nerve Image (ANI) of the Rollins tone.

---

[29] Available from http://www.ipem.rug.ac.be/Toolbox/.

In the IPEM-toolbox there are a number of modules based on the Auditory Peripheral Module, but I will only mention the Roughness Module here. *Roughness* or *sensory dissonance* was introduced by Helmholtz as a description of the texture of a sound dependent on impure or unpleasant qualities, and it can be defined as the energy of the relevant beating frequencies in the auditory channels (Leman 2000). As such, roughness is considered to be highly related to micro-level texture perception. The upper section of Figure 43 shows the energy distributed over the auditory channels of the Rollins tone (Example 15a). The middle section shows how the energy of the beating frequencies contributes to the roughness curve shown at the bottom. In this example, though, there is not much change in pitch, and thus the roughness varies very little.
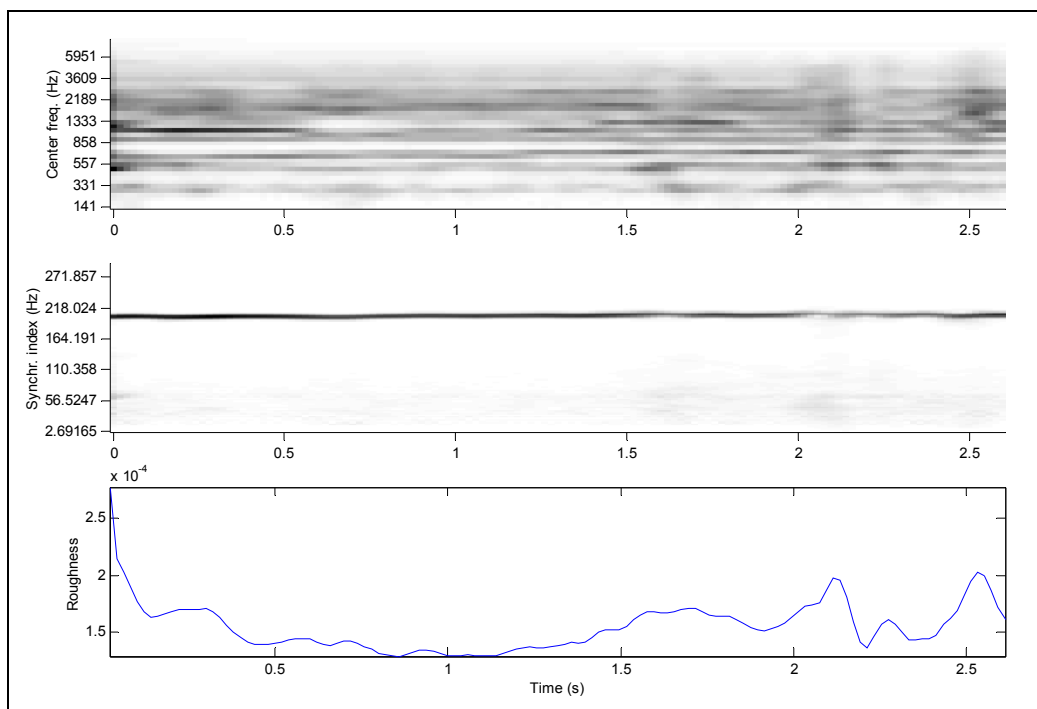
Figure 43. Output of the Roughness Module of the IPEM-toolbox for the Rollins tone. The upper section shows the energy distributed over the auditory channels, the middle section shows the contribution of the beating frequencies to the roughness curve at the bottom.

The idea of showing this example is that the model actually manages to recognize a "constant" pitch with slightly changing texture. So this example shows that new computer models seem promising when it comes to using a perceptual model for extraction of perceptually relevant information. Such an approach will hopefully result in major advances in the study of music in the coming years.

5.5   Conclusions

This chapter has presented various ways of analysing and displaying musical sound. This was exemplified with a seemingly simple saxophone tone lasting less than 3 seconds. However, as it turned out, this short example was in no way simple and well defined. The pitch tracker hit one octave off, and the spectrogram revealed quite large shifts in spectral energy throughout the sound. I also showed that about 60 harmonic frequencies are necessary to adequately synthesize the tone.

So the main conclusion from this discussion is that what might initially be thought of as a single tone with a slightly changing timbre is in no way easy to define and describe in physical terms. Furthermore, since Krumhansl and Iverson (1992) found that pitch and timbre actually interact for isolated tones, I agree with Houtsma (1997) in suggesting that the concepts of pitch and timbre should never be presented as independent variables. Developing perceptual models that take the multi-dimensional characteristics of sound into account is therefore important, and the IPEM-toolbox is a good example of such an implementation.

6 Artificial Neural Networks and Music

This chapter will discuss neural networks as one specific type of computer model that learns to process information in a way that may be similar to human perception. Theoretical background for neural networks and an example of training feedforward networks with timbre will be presented.

6.1 Connectionist vs. Symbolic Models

Previous chapters have shown the multi-dimensionality and complexity of auditory signals, and some of the difficulties when it comes to analysis and representation of music from such a sub-symbolic input. But since music perception seems to be a quite "easy" task for humans, we should try to make computer models work in a way similar to the human brain. Therefore I chose to look at artificial neural networks, and how they can be used to simulate neural activity. Before going into more detail about neural networks, it is worth mentioning that such networks are but one of many different models of "intelligent" computational systems. Such models in the world of "artificial intelligence" seem to be divided in two major directions; symbolic and rule-based models on one side, and connectionist models on the other.

Symbolic models are based on sets of rules for describing structures and processes. Take for example language, where grammar is the set of rules that govern how sentences should be generated and interpreted. To be able to speak or understand a language, it is necessary to know the basics of syntax, and semantics. Thus knowledge of the rules is essential for understanding or creating meaningful sentences. A computer model can therefore be "intelligent" if it knows the rules and will be able to both interpret and produce valid sequences within the boundaries of the system. Examples of rule-based systems in music composition are counterpoint, Bach choral harmonization and dodecaphony. There are also many examples of rule-based systems for music analysis, for instance (Lerdahl and Jackendoff 1983). The problem with most such systems, though, is that they will always be incomplete in the sense that they often assume additional knowledge which is not formalized.

Connectionism, on the other hand, is a direction in cognitive science with the aim of explaining human intellectual abilities using *artificial neural networks* (often referred to as *neural networks*). It is a relatively new direction, if one counts the work of (Rumelhart, McClelland, and others 1988) as when this research program came to the front stage (de

Sousa 1995). Neural networks are simplified models of the brain, composed of a large number of units connected together with weights. Those weights are measurements of the strength of connections between units in the network (Garson 2002). This is why connectionist models are still often referred to as *parallel distributed processing,* because the information that is contained in the system is not localized in single units in memory, but rather "stored" by activation throughout the whole network (Rumelhart, McClelland, and others 1988). This is similar to how the human brain is believed to function, and is thus also sometimes referred to as "neuromimetic" modelling.

Spangler (1999) argues that, when dealing with music, rule-based algorithms have several advantages. First of all, he claims that almost all music is rhythmic and tonal, and can therefore be measured in terms of quantized pitch and duration. Since rule-based systems are inherently discrete they will be able to account for this. He further argues that in a rule-based system it is straightforward to find the "reasoning" behind an output of the system, and therefore it is easier to determine where mistakes are made and what changes should be made to the algorithm. This is as opposed to a connectionist system where it might be very difficult to determine why a multi-layer network, based on dynamical principles, makes a given decision. He therefore concludes that rule-based systems are better when it comes to music analysis and creation.

A weakness with a rule based system, however, is the fact that it is limited by its rules, so the system will never be able to go beyond its boundaries. As such, it might work well in making strict counterpoint or analyse Bach chorales, but it will have serious problems when presented with elements that have not been, and/or cannot be, formalized in the system.

Another problem with rule-based systems is that they work *serially*. Just like a serial computer it can only do one operation at a time (Edelman 1992). This means that even though computers are getting faster and more powerful, the system will be slowed down because it always has to compare a target with each item in "memory" before finding a solution. This might secure accuracy within the system, but it will always be bound by its inability to "reason" and learn from experience.

These weaknesses support the claim that such serial processes are fundamentally different from processes of the human brain, since a rule based model is serial and symbolic, while the brain is believed to be a parallel, and distributed system (Smolensky, Mozer, and Rumelhart 1996). Even though the brain cannot perform calculations as fast as computers, it still has the ability to reason across all boundaries and quickly recognize and understand

complex structures. And if there are "holes" in our memory, by for example lack of information, we are still able to reason and understand structures.

In recent years, attempts to combine the central principles of computation in connectionist networks with that of symbolic computation, has resulted in the *optimality theory* (Prince and Smolensky 1993). It will be interesting to see this theory applied on a musical material.

6.2   The Self-Organizing Map

Before going into more detail about feedforward neural networks, I will just briefly mention one of the more popular connectionist models in recent years, the Self-Organizing Map (SOM). The SOM was intended as an effective software tool for conversion of nonlinear and high-dimensional data into simple geometric relationships on a low-dimensional display (Kohonen 2001: 106). This is interesting since it allows us to "see" for example an eight-dimensional structure when it is represented in a two-dimensional map.

The SOM is a method of unsupervised learning, meaning that the model is given a set of inputs and has to organize the content based on similarity within the data sets. Its great strength is the ability to learn structures in highly scattered and nonlinear material, and organize such large and complex sets into maps where items with similar features are plotted close to each other. In such cases where the data cannot be easily described in terms of mathematical functions, the SOM may relatively easily "see" the structures. A SOM is thus a simple abstraction of complex data, and has proven to be valuable in a number of complicated tasks. Figure 44 shows an example of a SOM where countries are mapped according to living conditions based on a 39-dimensional data set (with information such as state of health, nutrition, educational services, etc) from the World Bank statistics of 1992.

The popularity of the SOM algorithm has increased considerably the last years, and today it seems to be used in a wide variety of disciplines. Related to the field of this project is the use of SOMs for categorizing timbre in (Cosi, De Poli, and Lauzzana 1994) and (Feiten and Günzel 1994), in the development of analytical tools for speech processing (De Poli and Prandoni 1997), and in various types of music analysis, for example (Leman 1995). For this project, however, I decided to test out feedforward neural networks.
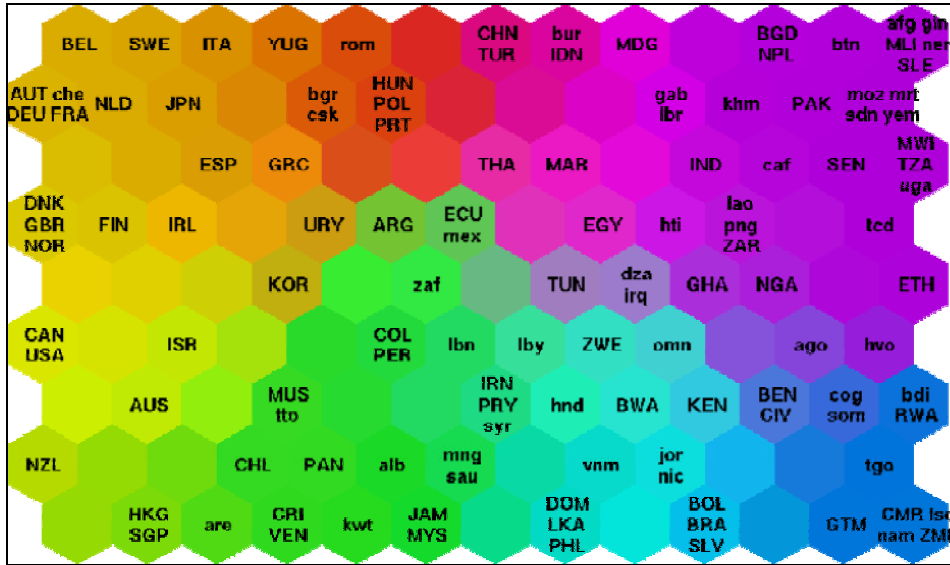
Figure 44. Example of a SOM where countries are organized according to a 39-dimensional data set indicating living conditions. Reproduced from (Kaski 1997).

6.3 Feedforward Neural Networks

Another popular connectionist model is the feedforward neural network (Figure 45). As opposed to SOMs, feedforward networks are based on *supervised* learning. That is, the network is trained with sets of both input and output data, so it learns specific outputs for given sets of input values. The following presentation is based on (Wasserman 1989).
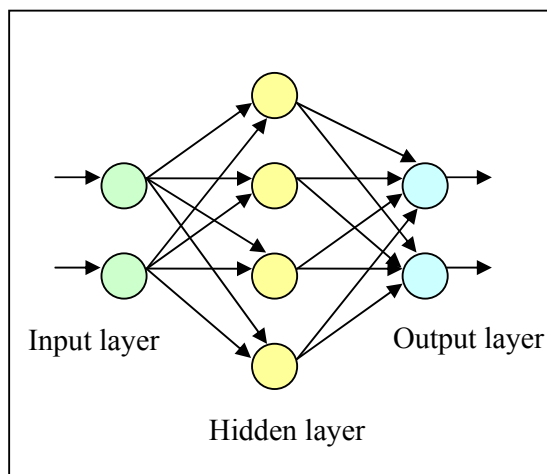
Figure 45. A fully connected multi-layer feedforward neural network consisting of an input, hidden and output layer.

A fully connected feedforward network is shown in Figure 45. Such a network is based on layers of neurons connected with *weights*. The *neuron* can be thought of as a very simple

"computer", since it has the ability to receive, process and transmit signals to other neurons. The *weights* between neurons govern how the network will process information, and a network is learning by changing these weights. In a *feedforward* network, all connections between neurons are going in the same direction, and the network therefore "feeds" its information forward.

The input to the neuron comes from the neurons preceding it in the net. Receiving an input, it sums its input and automatically makes an output when that input reaches a certain level. When the neuron outputs, or "fires", it influences the neurons it is connected to further on in the chain. A sketch of a simple artificial neuron is shown in Figure 46.
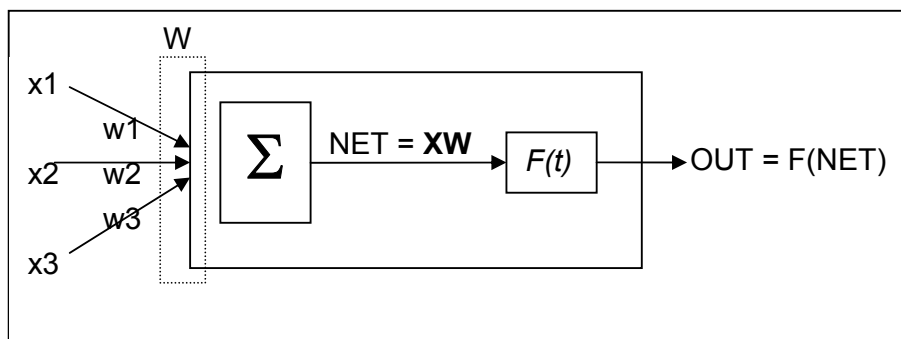
Figure 46. Artificial neuron with activation function (Wasserman 1989).

The inputs $x_1, x_2, ..., x_n$ applied to the neuron come from other neurons preceding it in the network. Each input is multiplied by a corresponding weight $w_1, w_2, ..., w_n$, as an analogy to the strength of that link. In the neuron, the weighted inputs are summed to find the activation level of the neuron. In compact vector form[30] this is shown in the figure as NET = **XW**. The function governing the output of the neuron is represented with *F(t)*. This can either be a linear function or a nonlinear threshold function, where *t* is some constant threshold value. Exactly what type of function that can be used depends on the paradigm and the algorithm being used (Kartalopoulos 1996). For now, I will just mention that two of the most popular threshold functions are the *sigmoid* and the *hard limiter* (Figure 47). A sigmoid function will output any value between 0 and 1 and is popular because of this monotony and because it has a simple derivative.

---

[30] *Scalars* (a quantity that has magnitude, but not direction) are shown in normal type, *vectors* (a quantity that has both magnitude and direction) in **bold**.
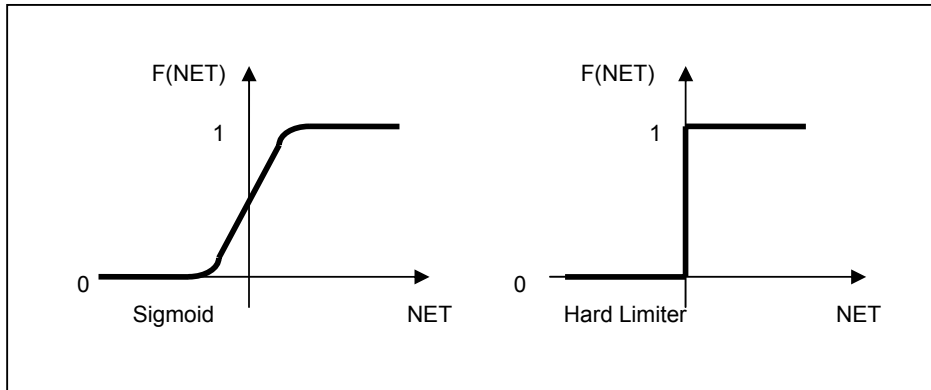
Figure 47. Two different types of threshold functions, the sigmoid and hard limiter.

A *hard limiter*, on the other side, is discontinuous at the origin, and is linear between its upper and lower bounds. It will output either 0 or 1, for example such that

OUT = 1   if NET > t

OUT = 0   otherwise

A neuron based on such a hard limiter function is called a *perceptron*. The perceptron is the basis for the backpropagation algorithm that will be discussed in more detail in the next section.

Most of the training algorithms in use today have evolved from the ideas of Hebb (Wasserman 1989). In his model from 1949, Hebb suggested that the *synaptic strength*, or the weight between two neurons, would be increased if both the source and destination neuron were activated. The idea behind this is simply that paths of activated neurons that occur often will also tend to happen often in the future, the classical idea of learning through experience. This type of learning is often called *Hebbian learning* (Kartalopoulos 1996).

An example from human perception of music may help to clarify this concept. Cadences have been mentioned earlier in this thesis as an example of how expectation arises if we hear a *IIm7-V7* progression. But since we sometimes also encounter the *IIm7-V7-VIm* progression, this will also be "assigned" a relatively higher probability than other progressions. This way we are trained in recognizing patterns and their possible outcomes. So a *IIm7-V7* will generate high expectations for either a *I* or a *Vm* chord. In a network such outcomes are governed by the strength of the weights between neurons, and the network will automatically "load" the sequence that is most likely to match the input.

A learning algorithm for a perceptron works like this:

1. Apply an input pattern and calculate the output *Y*
2. Evaluation:
   a. If the output is correct, go to step 1;
   b. If the output is incorrect and is zero, add each input to its corresponding weight
   c. If the output is incorrect and is one, subtract each input from its corresponding weight
3. Go to step 1;

For continuous inputs and outputs, this method is generalized to what is called the *Delta Rule*. From step 2 of the perceptron learning algorithm, the difference between the target output *T* and the actual output *A* may be represented as

$$\delta = (T - A)$$
(Equation 2.)

Notice how step 2.a corresponds to $\delta = 0$, 2.b corresponds to $\delta > 0$, and 2.c corresponds to $\delta < 0$. For all these cases, the algorithm is satisfied if δ is multiplied by the value of each input x_i to the perceptron, and this product is added to the corresponding weight. Introducing the coefficient η as a learning rate to control the average size of weight changes we get

$$\Delta_i = \eta \delta x_i$$
$$w_i(n+1) = w_i(n) + \Delta_i$$
(Equation 3.)

where

$\Delta_i =$ the correction associated with the *i*th input x_i

$w_i(n+1) =$ the value of weight *i* after adjustment

$w_i(n) =$ the value of weight *i* before adjustment

This rule works appropriately for target and actual outputs, for both continuous and binary inputs and outputs. A problem, however, is that there is no way to know the number of training cycles that is required, except that it is a finite number.

6.4 Backpropagation

One of the most popular training processes for feedforward neural networks is the backpropagation algorithm. This algorithm was presented in 1986 by Rumelhart, Hinton and Williams, only to discover that it had been anticipated several times before, and as early as 1974 by Werbos (Wasserman 1989). This algorithm works with multi-layer networks such as shown in Figure 45, where there is one input, one hidden and one output layer of connected neurons. The neuron used in the backpropagation algorithm is shown in Figure 48, and is a slightly modified version of the neuron shown in the previous section.
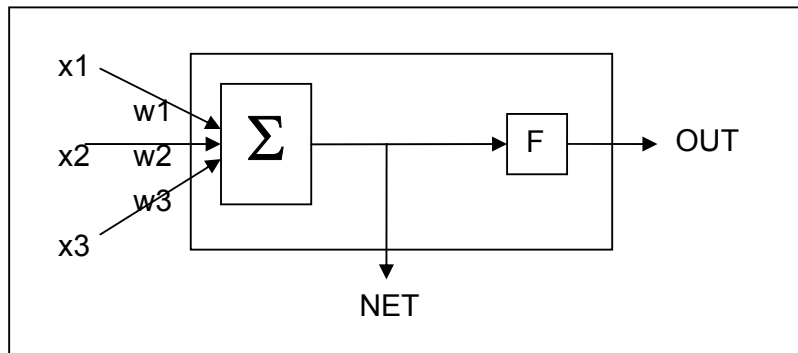
Figure 48. Sketch of an artificial neuron with activation function. The inputs are multiplied with the corresponding weights, summed and sent out as NET *and through the threshold function* F.

This neuron produces both NET and OUT signals, where NET is simply a sum of all the input values multiplied with the corresponding weight, such that

$$NET = x_1 w_1 + x_2 w_2 + ... + x_n w_n = \sum_{i=1}^{n} x_i w_i$$

Since the neuron used in the backpropagation is the perceptron, it is governed by a hard limiter threshold function, outputting either 0 or 1, where

$$OUT = F(NET)$$

An overview of the training process for the backpropagation network is as follows:

1. Select the next training pair and apply the input vector to the network input.
2. Calculate the output of the network.
3. Calculate the error between the network output and the desired output.
4. Adjust the weights of the network to minimize the error.
5. Repeat steps 1-4 for each vector in the training set. Stop when the sum error is low enough.

The error found in step 4 is the basis for tracing how the total error of the network is changing over time. This is important for deciding when to stop the training process.

The weights of the output layer are adjusted using an equation quite similar to the delta rule presented earlier, so

$$\Delta w_{pq,k} = \eta \delta_{q,k} OUT_{p.j}$$ 　　　　　　(Equation 4.)

and

$$w_{pq,k}(n+1) = w_{pq,k}(n) + \Delta w_{pq,k}$$ 　　　　　　(Equation 5.)

The hidden layers have no target vector, but backpropagation solves this by propagating the output error back through the network layer by layer, adjusting each layer on its way. So for the hidden layers, $\delta_{q,k}$ will not be present and must be calculated by

$$\delta_{pj} = OUT_{pj}(1 - OUT_{pj})(\sum_{q} \delta_{q,k} w_{pq,k})$$ 　　　　　(Equation 6.)

When the network is training, the general error function will (usually) decrease as the weights are getting adjusted. Eventually the network will come to a point where the error function does not get smaller, and that is called a minimum. However, there might be cases where there can be several local minima, but only one global. Such an error graph as a function of weight is shown Figure 49.

Even though there are two cases that result in a low error in Figure 49, there is only one global minimum. The problem is that when the learning algorithm reaches a local minimum (point A) it will not be able to reach the global minimum (point B). Recall from the previously presented equations that the weights are adjusted slightly in a direction that further decreases the error. If a global minimum is reached, we will necessarily have to increase the error function somewhat to be able to get to the global minimum. Since it is not possible to increase the error, we will either have to be satisfied with the result or start a new training session. The reason that a new session might overcome the local minimum, is because starting with "new" randomized weights for the whole network might result in lots of weights to vary. This might make it easier to "jump" over the local minimum and get to the global minimum (Dolson 1991).
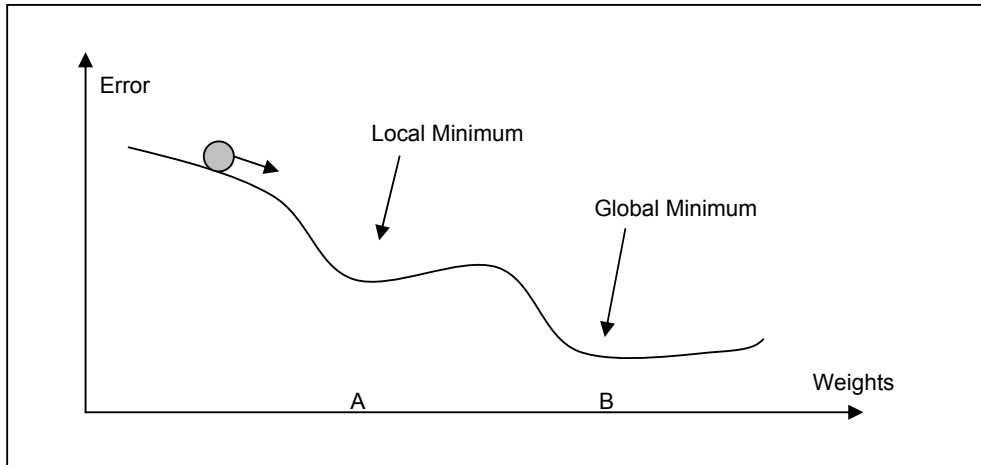
Figure 49. Error as function of weights. There might be several local minima, but only one global minimum. An analogy to the training progression can be that of a ball rolling down a slope. Ideally it should roll all the way down to the global minimum, but it might also get stuck in the local minimum, and the training will have to start over.

Even though the backpropagation algorithm has been one of the most popular neural networks algorithms, it has also been criticized for its nonbiological behaviour. Kartalopoulos (1996) argues that biological neurons do not seem to work backwards to adjust their synaptic weights. As such, the algorithm can not be seen as a learning process simulating biological behaviours but rather as a method to design a network with learning. It has also been argued that the algorithm involves a lot of calculations and trains slowly for large networks, since the calculation time is proportional to the size of the network. Training is much faster when updating of the weights occurs after each training vector rather than for the whole training set (Tørresen 1997), but still the algorithm is not very well suited for real-time calculations. For my purpose, however, it has worked fine.

6.5   Simulating Timbre Recognition in a Neural Network

As a test, I decided to see if neural networks can be "trained" with timbre. If this is the case, such a trained network could then be used for either analysis or synthesis of instruments. The reasoning behind this is that humans have no problems of recognizing or identifying timbre, and since neural networks are meant to resemble the human brain they should also be able to do this. This idea was first suggested by (Dolson 1991), and later experiments by Wessel, Drame and Wright (1998) showed that this is indeed possible.

But what does it actually mean to "learn" the timbre of an instrument? What values should be used to train with? How can the training values be related to our perception? Since the interest is on investigating human perception, we should start by remembering that we

perceive a tone with an associated pitch, loudness and timbre. So a neural network should therefore learn to recognize these three components. The first problem, however, is to figure out how these three elements can be described in terms of the physical signal.

In the following I will refer to the Rollins tone (Example 15a). Let us start by investigating the output of the pitch tracker in Addan. This is a file containing the values for F0 over time and looks like this:

```
0.020000      202.519958
0.030000      204.513733
0.040000      205.732803
0.050000      206.172501
0.060000      206.209244
0.070000      206.219498
0.080000      206.302322
0.090000      206.415588
0.100000      206.585770
...           ...
```

Here the time window of 10 milliseconds is shown in the left column (in seconds), while the fundamental frequencies (in Hz) are in the right column. So this file helps solving the first problem, namely that of finding the pitch of the signal.

Next, a file with the spectral analysis of the sound shows information about the partials for each time window:

```
108    0.020000
1      206.744980    0.0133795282   2.397758
2      408.126007    0.0262175743   1.841558
3      609.051758    0.0387362503   1.888313
4      803.625549    0.1311578155   -2.563423
5      1010.885498   0.0567401312   2.021351
6      1215.241211   0.0564972423   0.464872
7      1428.672852   0.0234822202   0.049138
8      1623.785889   0.0651926547   -2.553016
9      1824.361450   0.0508930534   2.791992
10     2022.632690   0.0816930383   1.236977
...    ...           ...            ...
```

This file is organized with a "header" before each new list of partials. One such header is shown above, and consists of a line with the numbers 108 and 0.020000. The first number (108) is the total number of partial frequencies found, and the second number (0.02) is the start of the time window in seconds. This indicates that there will be 108 lines containing information about the partials following this header. Each line of information about the partials contains four columns: partial number, frequency (Hz), loudness (relative amplitude), and phase (radians).

Based on this information we need to extract the features that correspond to what is perceived as loudness and timbre of the tone. Since the amplitude of each partial frequency is given in this file, it should be possible to estimate the perceived overall loudness of the tone from these values. When it comes to timbre, it can be adequately described with reference to about the 60 first partials and their amplitudes (as shown in Section 5.3). To simplify things, we can assume that the partial frequencies are harmonic, and can therefore be calculated by multiplying the value of F0 with the corresponding harmonic number. So when F0 is already in the training set, it is sufficient to include the amplitudes of each harmonic to be able to represent the timbre of the tone.

To summarize what the training data looks like, these are the following relationships between perceived attributes and physical signal:

- Pitch ⇔ Fundamental frequency (F0)
- Loudness ⇔ Sum of partial amplitudes
- Timbre ⇔ Set of partial amplitudes

Since feedforward networks are based on supervised learning, they are trained by applying both "question" and "answer" in the training process. As such, they learn to associate relationships between different sets of values. In this simulation I wanted the network to learn relationships between F0 and the sum of partial amplitudes on one side, and the set of partial amplitudes on the other. This means that a trained network would output the amplitude values of 60 harmonics when controlled by F0 and overall loudness.

There are some drawbacks with this approach. First, the assumption that all partials are harmonics will necessarily remove information about any inharmonic movement and minor transient changes. However, to reduce the training data to a feasible size, this seems to be the best option. Second, the networks will only be trained with "stationary" spectra, so there will be no information connecting consecutive time windows in the trained network. This means that the trained network will only be able to reproduce sound successfully if it is controlled with values of F0 and loudness similar to what it has been trained with. However, since my interest in this simulation was only to see whether networks can actually learn complex data structures such as timbre, this was not really a problem for this project.

6.6 Training the Neural Network

All the tools used in this simulation is shown in Figure 50. In the following I will briefly go through the various parts.

Sound In: Sony Rollins saxophone sounds
Addan: IRCAM-software for doing spectral analysis
Addan2Pat: Perl-script for preparing training data
JavaNNS: Software for training neural networks
SNNS2C: Program for converting trained network to C-code
C-wrapper: C-code to compile trained network to MAX-objects
MAX/MSP: Object to control trained network
Sound Out
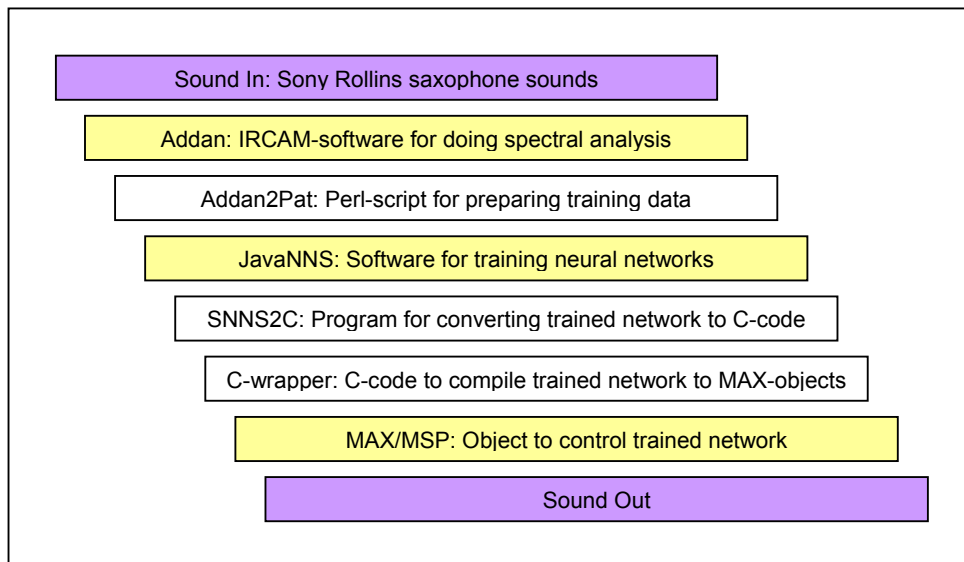
Figure 50. Work chain from original sound in to network-synthesized sound out.

First of all, it was necessary to decide on the sound input. To simplify things I decided to use saxophone sounds only, taken from *The Solo Album* by Sony Rollins. This CD features nearly an hour of solo tenor saxophone, and is therefore a good source for finding examples from the entire range of the instrument. The recording quality is also quite good, with little noise and external sounds. From this CD five segments were selected (Examples 17a-e) of varying length and complexity, covering a large dynamic range and register of the instrument.

Second, pitch tracking and spectral analysis was done in *Addan* and output as text files. The content of these files were shown in the previous section, and formed the basis for extracting material to be used for the training.

When it comes to the training of the networks, I never intended to implement the backpropagation algorithm myself, and therefore relied upon finding available software. The Neural Network Toolbox in Matlab could have been used, but I finally selected the *Stuttgart Neural Network Simulator* (SNNS). With the newly released Java implementation *JavaNNS*, this runs smoothly on both Windows and Linux computers, and it offers far more options than necessary. However, the most important factor for choosing JavaNNS was a small Windows-program called *SNNS2C,* which was distributed with the software. This program takes a trained network file and converts it to a C-function, which can be

compiled to a MAX/MSP object. This way it would be possible to write a MAX/MSP patch controlling the trained neural network.

Since JavaNNS was chosen for the training, the next problem was to merge and format the analysis data from Addan into the Pattern-files required by JavaNNS. This turned out to be the most time-consuming part of the whole project, since I ended up learning Perl-scripting from scratch. In retrospect, this could probably have been solved more easily in for example Matlab or even MAX/MSP, but I ended up spending weeks of debugging Perl-code before finally getting the little program *Addan2Pat* to work correctly. The source code for this program can be found in Appendix 3, and in the following its operations will be briefly explained.

Addan2pat takes two files as input, one containing the values of F0 over time and another with results from the spectral analysis (see Section 6.5), and outputs a file in standard SNNS pattern-format:

```
SNNS pattern definition file V3.2
generated at Fri Apr 22 13:25:23 2002

No. of patterns : 279
No. of input units : 2
No. of output units : 60

# Input pattern 1:
0.853751817703236 0.9146766973
# Output pattern 1:
0.0133795282   0.0262175743   0.0387362503   0.1311578155   0.0567401312   0.0564972423
0.0234822202   0.0651926547   0.0508930534   0.0816930383   0.0626092851   0.0177947525
0.0163455717 0.0173242018 0.0184137784 0 0.0192239378 0.024825735 0 0.0105436686 0 0
0.0119776605 0.017275244 0.0108059598 0.016561918 0.0206235815 0.0120163495 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

This pattern-file contains a header with information about when the file was created, the number of patterns (training sets), and the number of input and output values that the network will be asked to adjust its weights according to. The two values shown under "Input pattern 1" are F0 and overall loudness, while the values shown under "Output pattern 1" are the 60 harmonic amplitudes. These input and output pattern values make up one training set.

Notice that all the values in the pattern-file are in the range between 0 and 1. This is due to the dynamics of the network, where faster training will be obtained if the values are in a similar range. So the values were therefore normalized and linearly scaled to these limits in Addan2Pat.

The pattern-files created by Addan2Pat were opened in JavaNNS and training was done with the learning rate coefficient η set to 0.2 (see Equation 3). The weights of the network were randomized prior to training, and the patterns of training data were shuffled for each training cycle. Due to the quite large training sets, I had to run up to 3000 cycles before the network was well trained. Figure 51 shows a screenshot from JavaNNS with the fully connected network in the background, the error graph and control panel.
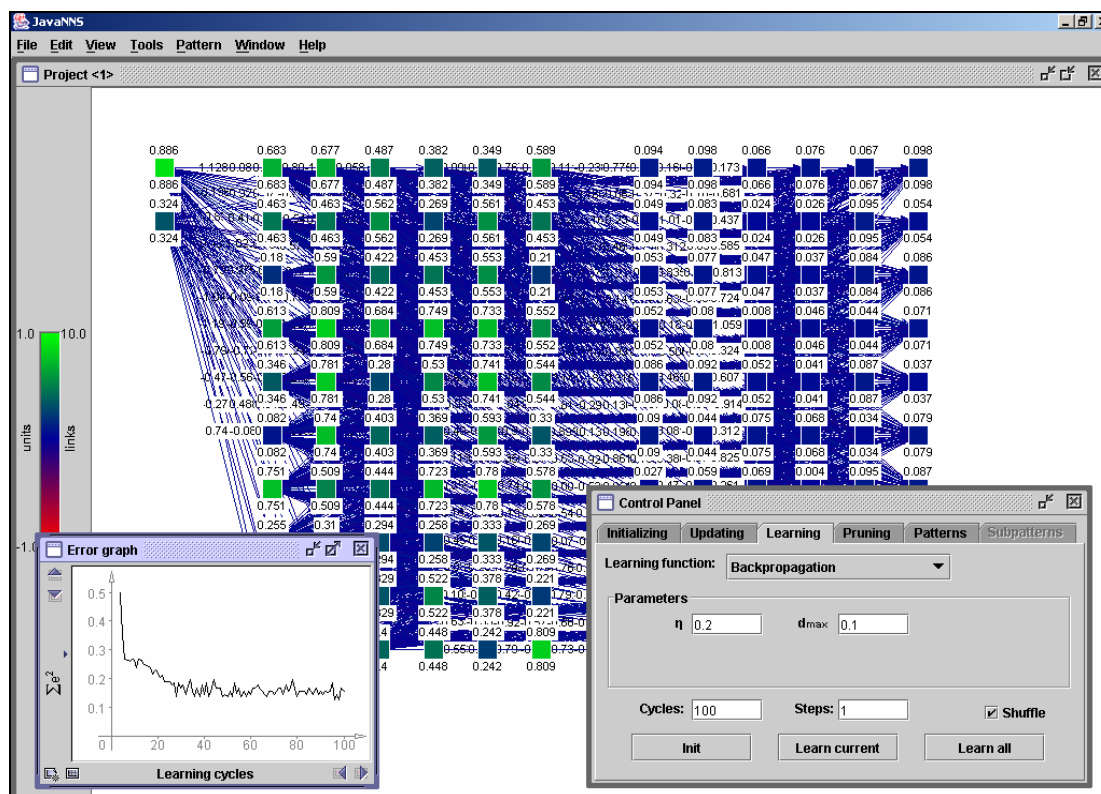
Figure 51. Interface of JavaNNS. The 2-60-60 network is shown in the background. A plot of the decreasing error function is shown in the bottom left corner, and a control panel for the training is shown in the bottom right corner.

The trained network was saved to a text file containing information about connections of the network and the values of the weights between neurons. This text file was converted to a C-function with the little program *SNNS2C*, and this function was used together with a "C-wrapper" [31] when compiling a MAX/MSP object of the trained network. Compilation was done with the commercially available *Metrowerks CodeWarrior 7.0* under Mac OS 9.2. So finally, after all these various stages, the compiled object could be used for timbre synthesis.

---

[31] The C-wrapper was written by Matthew Wright at CNMAT, and is available in Appendix 4.

6.7   The Trained Neural Network Object

The compiled object of the trained network can be used as any other MAX/MSP object. As shown in Figure 52, the object takes two inputs (F0 and overall loudness) and outputs a list of 60 values (amplitudes for each harmonic). It is important to remember that even though the network has been trained to learn relationships between F0/Loudness and sets of amplitudes, the actual output of the network will be based on the overall activation of the network. This means that applying a certain F0/Loudness to the network will not necessarily result in a set of amplitudes that is an exact match with the training data, since the weights have been adjusted to give the best *overall* performance. On the other hand, this flexibility of the network makes it able to generalize beyond the data sets it has been trained with.
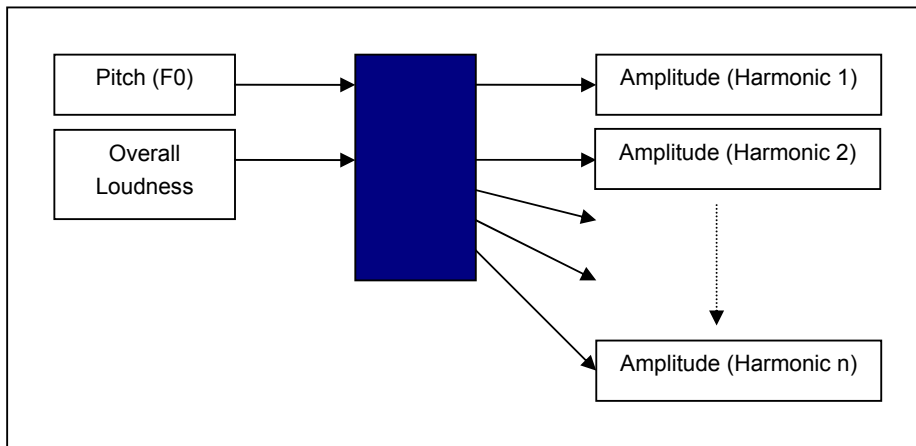
Figure 52. The trained network object takes two inputs (pitch, loudness) and outputs 60 values as a list (amplitudes for each harmonic).

To control the object I made the patch *NN-Control*. The "interior" of the user interface is shown in Figure 53, and shows how a *multislider* object with 60 sliders is connected to the network object. The harmonic frequencies are found by simply multiplying the value for F0 with the corresponding harmonic number, such as described in some of the previously described patches (see Section 2.3). The list of the harmonic frequencies is merged with a list of the harmonic amplitudes coming from the network, and sent to the *sinusoids~* object for the additive synthesis. The input values to the network can be controlled by changing the F0 and Loudness sliders, or by using the two-dimensional "control space" allowing control of pitch (horizontal) and loudness (vertical) with a mouse or graphical tablet.

I recommend the reader to try and play with the object. Notice that even though the network only produces "stationary" spectra there is certainly some saxophone quality of the output sound.

Figure 53. Control patch for the trained neural network object. The network is connected to a multislider showing the amplitudes of each harmonic.

The fact that the network actually managed to train, and that the resultant sound has some saxophone-like quality, can be taken as an indication of a successful simulation. So it can be concluded that the network did indeed manage to "learn" timbre. However, there are also a number of elements that can be improved. First, the simple addition of harmonic amplitudes to find the overall loudness does not take into account the non-linearity of the auditory system. So implementing an auditory model would be much better. Second, the normalization and scaling of the training values were rather rough, and I suspect this to be a reason for problems with too high values for the upper partials of the output sound.

A more fundamental weakness of this model is that it does not take into account the development of partials (harmonic and inharmonic) through time. This might be improved in the future by training a second network with some time-varying parameters. Yet another improvement would be to train networks with timbre from various instruments, and also to associate instrument names with these different training sets. This could again be used to compile reversed network objects, where timbre can be input and the network can relate

this to the correct instrument name. These and many other improvements will have to wait for future projects. As for now, I am quite satisfied to have gotten this far.

6.8   Conclusions

This chapter started with a discussion of the differences between rule-based and connectionist models. A problem with rule-based systems is that they work serially and will therefore be bound by always doing processing sequentially. They will also always be limited to the rules they are set up with, and can never learn to generalize or come up with solutions outside of its "domain". Connectionist models or artificial neural networks, on the other hand, are parallel and distributed systems working by activation of neurons. They can therefore process different types of information in parallel throughout the network, and can learn from experience by adjusting weights between the neurons.

My interest was, besides learning the basic theories of neural networks, to see if it is possible to train a neural network with timbre. I decided to train networks to learn relationships between fundamental frequency (F0) and overall loudness as input values, and sets of harmonic amplitudes as output values. The focus of the simulation was more on understanding the concepts behind neural networks and getting everything to work, than the actual results. Thus the expectations were moderate and the fact that I actually managed to get every chain in the process to work, was in itself satisfactory. This involved doing spectral analysis in *Addan*, writing the Perl-script *Addan2pat*, training networks in *JavaNNS*, converting the trained networks to C-functions with *SNNS2C,* compiling a MAX/MSP object with a "C-wrapper", and finally making a patch running the object in MAX/MSP. This setup seems to work well and might be suitable for future experiments.

The fact that the networks managed to train well and that the output sound might be characterized as "saxophone-like" seems like a good indication that the simulation was successful. The reason that the sound output of the network object is not very pleasing is due to the fact that the networks are only trained to learn "stationary" spectra. So an improvement of the model would be to add some way to control instrumental-specific envelopes. More training data, better normalization, and implementation of an auditory model would also be good improvements for future simulations.

Despite the seemingly "uselessness" of the trained object as it is now, the results might be taken as an indication that neural networks can be used for understanding more of how the human brain is recognizing timbre. With the improvements mentioned above, I believe such a model can be used both for recognition of different instruments directly from the sound source and also as a control structure for instrument synthesis.

7 Conclusions

In this chapter discussions and findings in previous chapters are summarized, and some directions for future research are suggested.

7.1 Summary

The project started with the observation that we can recognize a song in one second. This problem has been approached from three different perspectives. First, theories and problems related to physical and perceptual aspects were discussed. Second, music recognition was discussed and illustrated, and various musical examples were analysed with respect to traditional musical parameters. Third, different methods of analysing, visualizing and synthesizing timbre were presented, including a simulation of training neural networks.

An important part of this project has been the approach to music theory through perceptual models, using the sounding music as source of analysis. There are a number of reasons why I advocate such an approach. First of all, musical notation is a symbolic system that is mostly a description of actions to be performed on instruments. The music we hear is highly dependent on the interpretation by a performer, and these sounding qualities should therefore be taken into account when doing analysis. Second, a lot of music was actually never notated before it was played, and can therefore only be studied from an auditory perspective.

Considering the processes that govern music recognition, I suggested that they can roughly be divided in two groups:

- Segregation of sensory input
- Recognition of musical content

The segregation of auditory input is often referred to as *auditory scene analysis,* and involves grouping by primitive processes and/or schemata activated in voluntary or automatic ways. Perception of pitch and timbre, for example, is based on the grouping of frequencies coming from the same spatial location and starting at the same time. Actually, when dealing with audio it is important to remember that there are no simple and well-defined correspondences between traditional musical concepts and the physical signal. One example of this is the sensitivity region of the auditory system to certain frequencies. Such

non-linearity calls for caution when reading visual displays of auditory information and when making computer models.

The actual recognition of musical content is highly individual and subjective. Exactly what is recognized from a song has been of less interest for this study than the underlying musical features that make recognition possible. A question posed in the introduction was that of how much time is actually needed to recognize a song. There does not seem to be a clear answer to this, and I found that some songs are recognized after less than a second, while others might require more than a minute of listening before they are recognized. What governs the recognition process is, of course, musical content, and I think a key concept in this respect is that of *salience*. Generally it seems that musical excerpts containing highly salient features are much easier to recognize quickly.

Trying to figure out what might contribute to perceptual salience, I argued that traditional musical parameters such as melody, harmony, rhythm, dynamics and timbre, can all be salient, either alone or in various combinations. What also emerged from the analysis was that that the *sound* of the music seems to be significant for music perception. This might seem obvious, but unfortunately this has so far received little attention in traditional music analysis.

I have also argued that timbre is the main constituent of musical sound, and a reason why it is so difficult to study lies in its multi-dimensional nature. Even though we usually have no problems recognizing or categorizing timbre, it is not well-defined in either normal language or physical terms. But with the advent of faster computers and better models for signal processing, it is now possible to analyse, visualize and synthesize timbre in many different ways. Some issues related to this were presented and discussed in Chapter 5, and a conclusion to be drawn from this was that pitch, timbre and loudness should preferably not be studied separately. This is so because these phenomena all rely on the grouping of frequencies in time, and do not seem to be easily defined and separable in terms of physical units.

The complexity and multi-dimensionality of timbre calls for alternative ways of doing analysis and synthesis. One way of studying this is by using artificial neural networks modelling neural activity in the brain. To test this, I showed an example of how neural networks can be trained with saxophone sounds. Although there are a number of elements that can be improved in future simulations, the important result is that the network actually managed to learn the large and complex data set it was trained with.

When it comes to how computers can learn to listen like humans, I think it is important that such models are based on perceptual processes. It seems that considerable effort has been put into making computers transcribe music before doing traditional analysis. That such methods are not the best way to approach the problem, is supported by the fact that even though advances in computer science have been enormous the last decades, there are still no computer systems that come close to any human in terms of recognition of musical content. I believe that a greater knowledge about our own processing of music can help in developing better systems. This can help in both understanding more about music perception, and also in developing new and better methods for analysis and synthesis.

7.2   Future Directions

Ending this thesis, I would like to point out a couple of interesting research projects that have been presented recently. George Tzanetakis (2002) has finished the MARSYAS framework, a collection of tools for doing analysis and retrieval of music, including "a general multifeature audio texture segmentation methodology, feature extraction from mp3 compressed data, automatic beat detection and analysis based on the Discrete Wavelet Transform and musical genre classification combining timbral, rhythmic and harmonic features" (Tzanetakis 2002: iv). He has also developed some novel graphical user interfaces that allows for browsing and visualizing large audio collections. As part of this work, he has obtained some amazing results of style recognition by only using beat-tracking (Tzanetakis, Ermolinskyi, and Cook 2002).

Another interesting study is the *Sound Spotting* techniques by Christian Spevak (2002). This method is based on finding perceptually similar sounds by query-by-example. The algorithms are implemented in Matlab, and allow the user to manually select a musical passage from a sound file, and search for similar excerpts in the rest of the file. The audio data is pre-processed with an auditory model and the signal is divided into frames, each with an associated feature vector. Vector quantization and mapping is performed in a Self-Organizing Map (SOM), and then finally a pattern matching is applied (Spevak, Polfreman, and Loomes 2001). Such a method seems very promising since it is incorporating both auditory models and also neural processing. The problem, however, is to make it fast enough to be able to search through large sound databases. Another problem is that the user has to be very specific when choosing the example to search for.

Finally, I think that making computational models that could combine such methods for finding similar sounds with that of automatic recognition of salience points, may solve many problems. Not only could it be possible to make music "thumbnails" based on salience, but it could also revolutionize music information retrieval.

References

Aksnes, Hallgjerd. 2002. Perspectives of Musical Meaning. A Study Based on Selected Works by Geirr Tveitt. Dr. art. thesis, Department of Music and Theatre, University of Oslo, Oslo.

Beauchamp, J. W. 1982. Synthesis by Spectral Amplitude and "Brightness" Matching of Analyzed Musical Instrument Tones. *Journal of Audio Engineering Society* 30 (6):396-406.

Bent, Ian. 1980. Analysis. In *The New Grove Dictionary of Music and Musicians*, edited by S. Sadie. London: MacMillan Press.

Bordwell, David, and Kristin Thompson. 1997. *Film Art. An Introduction*. 5th ed. New York: McGraw-Hill Companies, Inc.

Bregman, Albert S. 1990. *Auditory Scene Analysis. The Perceptual Organization of Sound*. Cambridge, Massachusetts: The MIT Press.

———. 1993. Auditory Scene Analysis: Hearing in Complex Environments. In *Thinking in Sound: The Cognitive Psychology of Human Audition*, edited by S. McAdams and E. Bigand. Oxford: Clarendon Press.

Bregman, Albert S., and A. Rudnicky. 1975. Auditory Segregation: Stream or Streams? *Journal of Experimental Psychology: Human Perception and Performance* 1:263-267.

Carr, Ian. 1991. *Keith Jarrett. The Man and His Music*. London: Grafton.

Cosi, Piero, Giovanni De Poli, and Giampaolo Lauzzana. 1994. Auditory Modelling and Self-Organizing Neural Networks for Timbre Classification. *Journal of New Music Research* 23 (1):71-98.

Crowther, Jonathan, ed. 1995. *Oxford Advanced Learner's Dictionary of Current English*. Fifth ed. Oxford: Oxford University Press.

Darwin, C. J., and R. P. Carlyon. 1995. Auditory Grouping. In *Hearing*, edited by B. C. J. Moore. San Diego: Academic Press.

De Poli, Giovanni, and Paolo Prandoni. 1997. Sonological Models for Timbre Characterization. *Journal of New Music Research* 26 (2):170-197.

de Sousa, Ronald. 1995. Turns of Minds. *Semiotic Review of Books* 7 (1).

Dixon, Simon, and Emilios Cambouropoulos. 2000. Beat Tracking with Musical Knowledge. Paper read at ECAI.

Dolson, Mark. 1991. Machine Tongues XII: Neural Networks. In *Music and Connectionism*, edited by P. M. Todd and D. G. Loy. Cambridge, MA: The MIT Press.

Edelman, Gerald M. 1992. *Bright Air, Brilliant Fire. On the Matter of the Mind*. New York: BasicBooks.

Eitan, Zohar. 1997. *Highpoints. A Study of Melodic Peaks*. Philadelphia: University of Pennsylvania Press.

Feiten, Bernhard, and Stefan Günzel. 1994. Automatic Indexing of a Sound Database Using Self-Organizing Neural Nets. *Computer Music Journal* 18 (3):53-65.

Fiske, Harold. 1996. *Selected Theories of Music Perception*. Vol. 49, *Studies in the History and Interpretation of Music*. Lewiston: The Edwin Mellen Press.

Garson, James. 2002. *Connectionism. The Stanford Encyclopedia of Philosophy (Winter 2002 Edition)*. Edward N. Zalta 2002 [cited 2002]. Available from http://plato.stanford.edu/archives/win2002/entries/connectionism/.

Godøy, Rolf Inge. 1997a. *Formalization and Epistemology*. Oslo: Scandinavian University Press.

———. 1997b. Knowledge in Music Theory by Shapes of Musical Objects and Sound-Producing Actions. In *Music, Gestalt, and Computing: Studies in Cognitive and Systematic Musicology*, edited by M. Leman. Berlin: Springer-Verlag.

———. 1999. *Shapes and Spaces in Musical Thinking*. Oslo: Department of Music and Theatre.

———. 2001. Imagined Action, Excitation, and Resonance. In *Musical Imagery*, edited by R. I. Godøy and H. Jørgensen. Lisse: Swets and Zeitlinger.

Goto, Masataka, and Satoru Hayamizu. 1999. A Real-time Music Scene Description System: Detecting Melody and Bass Lines in Audio Signals. Paper read at IJCAI Workshop on Computational Auditory Scene Analysis.

Grey, John. 1977. Multidimensional Perceptual Scaling of Musical Timbres. *Journal of the Acoustical Society of America* 61:1270-1277.

Hanna, Patricia. 2001. Indeterminacy of Translation. In *Word and World*, edited by P. Hanna and B. Harrison.

Helmholtz, Hermann. 1885/1954. *On the Sensations of Tone*. Translated by A. J. Ellis. 2nd ed. New York: Longmanns & Co./Dover Publications. Original edition, Die Lehre von den Tonempfindungen.

Houtsma, A. J. M. 1997. Pitch and Timbre: Definition, Meaning and Use. *Journal of New Music Research* 26 (2):104-115.

Huron, David. 1996. The Melodic Arch in Western Folksongs. *Computing in Musicology* 10:3-23.

Illingworth, Valerie, ed. 1991. *The Penguin Dictionary of Physics*. Second ed. London: Penguin Books.

Jensenius, Alexander Refsum. 1999. Digitalisering av pianolyd - Noen problemområder, med vekt på fysisk signal og menneskelig oppfatning. Semester Paper, Department of Music and Theatre, University of Oslo, Oslo.

———. 2000. Datamaskin og lyder - bruker og påvirkning. Semester Paper, Department of Music and Theatre, University of Oslo, Oslo.

Kartalopoulos, Stamatios V. 1996. *Understanding Neural Networks and Fuzzy Logic. Basic Concepts and Applications*. New York: IEEE Press.

Kaski, Samuel. 2002. *World Poverty Map* [Web page] 1997 [cited 2002]. Available from http://www.cis.hut.fi/research/som-research/worldmap.html.

Klein, Julie Thompson. 1990. *Interdisciplinarity. History, Theory, and Practice*. Detroit, Michigan: Wayne State University Press.

Kohonen, Teuvo. 2001. *Self-Organizing Maps*. 3. ed. Berlin: Springer-Verlag.

Krumhansl, Carol L. 1989. Why is Musical Timbre so Hard to Understand? In *Structure and Perception of Electroacoustic Sound and Music*, edited by S. Nielzén and O. Olsson. Amsterdam: Elsevier.

———. 1995. Music Psychology and Music Theory. Problems and Prospects. *Music Theory Spectrum* 17 (1):53-80.

Krumhansl, Carol L., and Paul Iverson. 1992. Perceptual Interactions Between Musical Pitch and Timbre. *Journal of Experimental Psychology: Human Perception and Performance* 18 (3):739-751.

Large, Edward W., and Caroline Palmer. 2002. Perceiving Temporal Regularity in Music. *Cognitive Science* 26:1-37.

Lartillot, Olivier. 2002. Musical Analysis by Computer Following Cognitive Model of Induction of Analogies. Paper read at International Computer Music Conference, at Göteborg, Sweden.

Leman, Marc. 1995. *Music and Schema Theory*. Edited by T. S. Huang, T. Kohonen and M. R. Schroeder, *Spring Series in Information Sciences*. Berlin: Springer.

———. 2000. Visualization and Calculation of the Roughness of Acoustical Musical Signals Using the Synchronization Index Model (SIM). Paper read at COST G-6 Conference on Digital Audio Effects (DAFX-00), at Verona, Italy.

Leman, Marc, Micheline Lesaffre, and Koen Tanghe. 2001a. Introduction to the IPEM Toolbox for Perception- based Music Analysis. *Mikropolyphonie - The Online Contemporary Music Journal* 7.

———. 2001b. *Toolbox for Perception-Based Music Analysis - Concepts, Demos, and Reference Manual*.

Lerdahl, Fred, and R. Jackendoff. 1983. *A Generative Theory of Tonal Music*. Cambridge, Massachusetts: The MIT Press.

Levitin, Daniel J. 2000. In Search of the Musical Mind. *Cerebrum: The Dana Forum on Brain Science* 2 (4).

Mathews, Max. 1999. The Ear and How it Works. In *Music, Cognition and Computerized Sound : An Introduction to Psychoacoustics*, edited by P. R. Cook. Cambridge, Massachusetts: The MIT Press.

McAdams, Stephen, Suzanne Winsberg, Sophie Donnadieu, Geert De Soete, and Jochen Krimphoff. 1995. Perceptual Scaling of Synthesized Musical Timbres: Common Dimensions, Specificities, and Latent Subject Classes. *Psychological Research* 58:177-192.

Meyer, Leonard B. 1956. *Emotion and Meaning in Music*. Chicago: The University of Chicago University Press.

———. 1989. *Style and Music. Theory, History, and Ideology*. Chicago: The University of Chicago Press.

Narmour, Eugene. 1990. *The Analysis and Cognition of Basic Melodic Structures. The Implication-Realization Model*. Chicago: University of Chicago Press.

Parncutt, R. 1987. The Perception of Pulse in Musical Rhythm. In *Action and Perception in Rhythm and Music*, edited by A. Gabrielsson. Stockholm: Royal Swedish Academy of Music.

Peretz, Isabelle. 1993. Auditory Agnosia: a Functional Analysis. In *Thinking in Sound: The Cognitive Psychology of Human Audition*, edited by S. McAdams and E. Bigand. Oxford: Clarendon Press.

Plack, Christopher J., and Robert B. Carlyon. 1995. Loudness Perception and Intensity Coding. In *Hearing*, edited by B. C. J. Moore. San Diego: Academic Press.

Prince, Alan, and Paul Smolensky. 1993. Optimality Theory: Constraint interaction in generative grammar: Department of Computer Science, University of Colorado at Boulder and Rutgers Center for Cognitive Science, Rutgers University, New Brunswick, NJ.

Puckette, Miller. 1985. *A Real-Time Music Performance System*. Cambridge, Massachusetts: The MIT Department of Electrical Engineering.

———. 1988. The Patcher. Paper read at The International Computer Music Conference, at San Francisco.

Puckette, Miller, and David Zicarelli. 1990. *MAX - An Interactive Graphical Programming Environment*. Menlo Park: Opcode Systems.

Risset, Jean-Claude, and David L. Wessel. 1999. Exploration of Timbre by Analysis and Synthesis. In *The Psychology of Music*, edited by D. Deutsch. San Diego: Academic Press.

Roads, Curtis. 1996. *The Computer Music Tutorial*. Cambridge, Massachusetts: The MIT Press.

Rothgeb, John. 1997. Salient Features. *Music Theory in Concept and Practice*:181-196.

Rumelhart, David E., James L. McClelland, and others. 1988. *Parallel Distributed Processing. Explorations in the Microstructure of Cognition*. Vol. 1: Foundations. Cambridge, Massachusetts: The MIT Press.

Schaeffer, Pierre. 1966. *Traité des objets musicaux*. Paris: Éditions du Seuil.

Schafer, R. Murray. 1977. *The Soundscape: Our Sonic Environment and the Tuning of the World*. New York: Knopf.

Schneider, Albrecht, and Rolf Inge Godøy. 2001. Perspectives and Challenges of Musical Imagery. In *Musical Imagery*, edited by R. I. Godøy and H. Jørgensen. Lisse: Swets and Zeitlinger.

Schwarz, Diemo, and Matthew Wright. 2000. Extensions and Applications of the SDIF Sound Description Interchange Format. Paper read at International Computer Music Conference, at Berlin, Germany.

Serafine, M. L. 1988. *Music and Cognition: The Development of Thought in Sound*. New York: Columbia University Press.

Shepard, Roger. 1999. Cognitive Psychology and Music. In *Music, Cognition and Computerized Sound : An Introduction to Psychoacoustics*, edited by P. R. Cook. Cambridge, Massachusetts: The MIT Press.

Small, Christopher. 1998. *Musicking. The Meanings of Performing and Listening*. Hanover, New Hampshire: Wesleyan University Press.

Smolensky, Paul, M. C. Mozer, and David E. Rumelhart, eds. 1996. *Mathematical Perspectives on Neural Networks*. Mahwah, NJ: Lawrence Erlbaum Publishers.

Snyder, Bob. 2000. *Music and Memory. An Introduction*. Cambridge, Massachusetts: The MIT Press.

Spangler, Randall Richard. 1999. Rule-Based Analysis and Generation of Music. PhD thesis, California Institute of Technology, Pasadena, California.

Spevak, Christian. 2002. Sound Spotting Techniques Using Auditory Models and Self-Organizing Maps. PhD thesis, Department of Music, University of Hertfordshire.

Spevak, Christian, Richard Polfreman, and Martin Loomes. 2001. Towards Detection of Perceptually Similar Sounds: Investigating Self-Organizing Maps. Paper read at AISB '01 Symposium on Artificial Intelligence and Creativity in Arts and Science, at York.

Sundberg, Johan. 1999. The Perception of Singing. In *The Psychology of Music*, edited by D. Deutsch. San Diego: Academic Press.

Toiviainen, Petri, and Tuomas Eerola. 2001. A Method for Comparative Analysis of Folk Music Based on Musical Feature Extraction and Neural Networks. Paper read at VII International Symposium on Systematic and Comparative Musicology, at Jyväskylä, Finland.

Turicchia, L., Giovanni De Poli, and G. A. Mian. 2000. Audio Analysis by a Model of the Physiological Auditory System. Paper read at COST G-6 Conference on Digital Audio Effects (DAFX-00), at Verona, Italy.

Tzanetakis, George. 2002. Manipulation, Analysis and Retrieval Systems for Audio Signals. PhD thesis, Department of Computer Science, Princeton University.

Tzanetakis, George, Andreye Ermolinskyi, and Perry R. Cook. 2002. Beyond the Query-by-Example Paradigm: New Query Interfaces for Music Information Retrieval. Paper read at International Computer Music Conference, at Göteborg, Sweden.

Tørresen, Jim. 1997. The Convergence of Backpropagation Trained Neural Networks for Various Weight Update Frequencies. *International Journal of Neural Networks* 8 (3).

Wasserman, Philip D. 1989. *Neural Computing: Theory and Practice*. New York: Van Nostrand Reinhold.

Webster's Revised Unabridged Dictionary 2002. 1913 [cited 2002]. Available from http://smac.ucsd.edu/cgi-bin/http_webster?

Wessel, David L. 1979. Timbre Space as a Musical Control Structure. *Computer Music Journal* 3 (2):45-52.

Wessel, David L., Cyril Drame, and Matthew Wright. 1998. Removing the Time Axis from Spectral Model Analysis-based Additive Synthesis: Neural Networks versus Memory-Based Machine Learning. Paper read at International Computer Music Conference, at Ann Arbor, Michigan, ICMA.

Wright, Matthew, Amar Chaudhary, Adrian Freed, S. Khoury, and David L. Wessel. 1999. Audio Applications of the Sound Description Interchange Format Standard. Paper read at AES 107th Convention.

Wright, Matthew, Richard Dudas, S. Khoury, R. Wang, and David Zicarelli. 1999. Supporting the Sound Description Interchange Format in the MAX/MSP Environment. Paper read at International Computer Music Conference, at Beijing, China.

Appendix 1: Contents on the CD-ROM

Quick Start

Please double click the 'start.html' file on the CD-ROM to access the sound examples. The various applications can be run by double clicking the icons in the 'Applications' folder.

Folders and files on the CD-ROM

- **Applications**: This folder contains stand-alone applications compiled from the various MAX/MSP patches presented throughout the thesis. These applications should work on any Macintosh computer running Mac OS 9 (not OS X!). The MP3-files named 1-5 in the folder are used by the various applications, and may be exchanged with other sound files.
- **HTML**: This folder contains the HTML-files used for accessing the sounds. To open these files, please double click the 'start.html' file in the root directory.
- **Patches**: These are the original MAX/MSP patches as shown in the thesis. The patches require MAX/MSP 4 to run properly. The subfolder 'RequiredObjects' needs to be put into the MAX/MSP folder or the search path for the patches to work properly.
- **Sounds**: This folder contains the sound files referred to as "Example #" in the thesis. These files can be accessed most easily be opening the 'start.html' file in the root directory. The sound files can also be accessed directly from this folder, and have logical names "ex#.wav"
- **start.html**: This file can be opened in any web browser, and gives easy access to all the sound examples on the CD-ROM.

Patches / Applications

- **Harmonics**: Synthesis of a complex tone. Allows the user to control the loudness of up to 60 harmonics. Remember to turn on the sound and adjust the volume in the patch!
- **ST-Perception**: Measures short term perception. Choose whether the songs should start at the beginning (intro) or at a random start position. Clicking the big button starts playback of a song, and clicking once more stops playback. Recognition time is shown. The next song is loaded automatically. Results can be saved to a text file.
- **Measure-Salience**: Start a song and use the mouse to change the slider value.

- **Music-Trailer**: Select number and time of short segments, and hit the button to play a "music trailer".

- **Play-Add-Files**: Play saxophone tones with various numbers of harmonics. This file is only in the 'Patches' folders since it could not be compiled because of some of the external objects. However, if the folder 'RequiredObjects' is in the MAX/MSP search path, it will work fine when opened in MAX/MSP 4.

- **NN-Control**: Allows the user to control the trained neural network object. Remember to turn on the sound and adjust the volume in the patch! There are two modes: either continuous output or with an envelope.

Appendix 2: Matlab Code

Below is the source code (m-file) that I wrote to create the various plots and graphs presented in the thesis. Notice that some of the plots require the IPEM-toolbox to work.

```matlab
%-------------------------------------------%
%      Plot and Spectrogram of Sound File        %
%                                                %
%       Alexander Refsum Jensenius, 2002          %
%-------------------------------------------%


%----- Read file and Initialize Values ----
name='rollins3.wav';

sr=44100;                         % Set sample rate

a=wavread(name);                  % Import audio file
b=a(:,1);                         % Extract one channel from stereo files
b_samples=length(b);              % Find number of samples of file
b_seconds=b_samples/sr;           % Find duration of sound in seconds
b_t=([0:1:(b_samples-1)]')/sr;    % Make list of time in seconds to plot


%---- Play sound file ----
sound(b,sr);                      % Play sound file


%---- Time-domain plot ----
figure;                           % Opens figure
plot(b_t,b);                      % Simple plot of sample values against time in
seconds
%grid on;                         % Uncomment to draw grid in plot
xlabel('Time (s)');
ylabel('Amplitude');
title([int2str(b_samples) ' Samples - ' [num2str(b_seconds,2)] ' Seconds ']);


%---- Spectrogram ----
figure;                           % Opens new figure
specgram(b);
ylabel('Frequency');

xlabel('Time (s)');
title([int2str(length(b)) ' Samples - ' [num2str(b_seconds,2)] ' Seconds ']);
set(gca,'ytick', []);
set(gca,'xtick', [0:(sr/2):b_samples]);
set(gca,'XTickLabel',[0:1:((b_samples)/sr)]);


%---- Spectrum ----
figure;

b_spectr=spectrum(b,sr);
plot(b_spectr);
ylabel('Relative Amplitude');
xlabel('Frequency Hz');
title([int2str(length(b)) ' Samples - ' [num2str(b_seconds,2)] ' Seconds ']);
set(gca,'ytick', []);
%set(gca,'xtick', [0:(sr/2):b_samples]);
%set(gca,'XTickLabel',[0:1:((b_samples)/sr)]);


%---- ANI
figure;
[ANI, ANIFreq, ANIFilterFreqs] = IPEMCalcANI(b,sr,[],1);
```

```
%---- Roughness
figure;
[outRoughness, outSampleFreq, outFFTMatrix1, outFFTMatrix2] = ...
    IPEMRoughnessFFT(ANI, ANIFreq, ANIFilterFreqs, 5,300,0.20,0.02,1);


%---- Spectrogram
figure;
[a_s,a_t,a_f] = IPEMCalcSpectrogram(b,sr);


%---- Spectral Centroid
figure;
[Centroid,CentroidFreq] = ...
    IPEMCalcCentroid(ANI,ANIFreq,0.05,0.01, ANIFilterFreqs);
```

Appendix 3: Addan2Pat

Below is the source code of the Perl-script *Addan2Pat* that was used to prepare training data for JavaNNS.

```perl
#!/usr/local/bin/perl

#-------------------------------------------------------------#
#                    --- ADDAN2PAT ---                        #
#                         v.0.9                               #
#                                                             #
#                 Alexander Refsum Jensenius                  #
#                       ARJ (c) 2002                          #
#                                                             #
# Mission:  Read ADDAN F0 and ADD ASCII-files                 #
#           Split lines, find max-values                      #
#           Normalize and apply threshold (MUST be improved)  #
#           Write SNNS Pattern files                          #
#                                                             #
# Usage:    addan2pat <infile_f0> <infile_add> <outfile.pat>  #
#-------------------------------------------------------------#

use warnings;
#use strict;

#-------------------------------------------------------------#
# Define variables                                            #
#-------------------------------------------------------------#

# Read Command Line arguments
$file_in_f0   = $ARGV[0];
$file_in_add  = $ARGV[1];
$file_out     = $ARGV[2];

# Other variables to be used
$patterns     = 0;
$pattern_no   = 1;
$input_units  = 2;
$output_units = 60;
$add_line     = 1;
$max_f0       = 0;
$max_add      = 0;

#$theDate      = `date`;       # Calls the date from command line. Only works on
UNIX/Linux
$theDate      = "Mon Jan 01 00:00:00 2000";    # Manual date under Windows


#-------------------------------------------------------------#
# CHECK Correct Arguments                                     #
#-------------------------------------------------------------#

$argcnt = $#ARGV + 1;
if ($argcnt != 3) { die "Usage: addan2pat <infile_f0> <infile_add> <outfile.pat>\n";
}


#-------------------------------------------------------------#
# OPEN FILES - Die if not found                               #
#-------------------------------------------------------------#

open (INFILE_F0, $file_in_f0) or die "Problem with file: $file_in_f0";
open (INFILE_ADD, $file_in_add) or die "Problem with file: $file_in_add";
open (OUTFILE, ">$file_out") or die "Problem with file: $file_out";


#-------------------------------------------------------------#
```

```perl
# READ FILES - Check number of patterns                        #
#----------------------------------------------------------------#

@file_f0 = <INFILE_F0>;
foreach (@file_f0) { $patterns++; }
@file_add = <INFILE_ADD>;


#----------------------------------------------------------------#
# FIND MAX-VALUES                                                #
#----------------------------------------------------------------#

foreach $max_f0_line (@file_f0) {
    my @values = split /\s+/,$max_f0_line;
    $max_f0 = $values[1] if $max_f0 < $values[1];
}

# Reads the whole ADD-file into memory.
# Could not find any better way to do this.
foreach $max_add_line (@file_add) {
    my @values = split /\s+/,$max_add_line;
    $max_add = $values[2] if $max_add < $values[2];
}


#----------------------------------------------------------------#
# WRITE HEADER - Standard SNNS Pattern file header               #
#----------------------------------------------------------------#

print OUTFILE "SNNS pattern definition file V3.2\n";
print OUTFILE "generated at $theDate\n\n\n";
print OUTFILE "No. of patterns : $patterns\n";
print OUTFILE "No. of input units : $input_units\n";
print OUTFILE "No. of output units : $output_units\n";


#----------------------------------------------------------------#
# WRITE PATTERNS - Input and Output Patterns                     #
#----------------------------------------------------------------#

foreach $line (@file_f0) {

    # Find the F0-value
    print OUTFILE "\n\# Input pattern $pattern_no:\n";
    my @values = split /\s+/,$line;

    # -----> Normalization
    my $value_norm = $values[1]*0.9/$max_f0;

    print OUTFILE "$value_norm ";

    # Check for header (undefined $values_add[2]), scroll down to next
    @values_add = split /\s+/,$file_add[$add_line];

    while (defined($values_add[2])) {
        @values_add = split /\s+/,$file_add[$add_line];
        $add_line++;
        }

    # Calculate and output Loudness
    my $loudness_line=$add_line;
    $loudness = 0;

    for (my $x=0; $x < $output_units; $x++) {
        my @loudness_values = split /\s+/,$file_add[$loudness_line];
        $loudness=($loudness+$loudness_values[2]);
        $loudness_line++;
    }

    # -----> Normalization
    $loudness = $loudness;

    # -----> Threshold
    if ($loudness < 0.01) { $loudness = 0; }
    print OUTFILE "$loudness";
```

```perl
    # Find the Amplitudes and output
    print OUTFILE "\n\# Output pattern $pattern_no:\n";


    for (my $x=0; $x < $output_units; $x++) {
        my @values_add = split /\s+/,$file_add[$add_line];
        my $amplitude=1*$values_add[2];

        # -----> Normalization
        #$amplitude = 0.9*$amplitude/$max_add;

        # -----> Threshold
        if ($amplitude < 0.01) { $amplitude = 0; }
        print OUTFILE "$amplitude ";
        $add_line++;
        }


    $pattern_no++;
}


#---------------------------------------------------------------#
# Close Files - Write Message                                   #
#---------------------------------------------------------------#

close(OUTFILE);

print "FileInF0 : $file_in_f0\n";
print "FileInAdd: $file_in_add\n";
print "FileOut  : $file_out\n\n";

print "No. of patterns : $patterns\n";
print "No. of input units : $input_units\n";
print "No. of output units : $output_units\n";
```

Appendix 4: MAX/MSP C-Wrapper

To compile the C-functions with the trained networks to a MAX/MSP object, Matthew Wright at CNMAT wrote this C-wrapper. It takes a trained JavaNNS network file as .c and .h files as arguments and outputs the MAX/MSP object.

```c
/*
Copyright (c) 2002.  The Regents of the University of California (Regents). All
Rights Reserved.

Permission to use, copy, modify, and distribute this software and its documentation
for educational, research, and not-for-profit purposes, without fee and without a
signed licensing agreement, is hereby granted, provided that the above copyright
notice, this paragraph and the following two paragraphs appear in all copies,
modifications, and distributions. Contact The Office of Technology Licensing, UC
Berkeley, 2150 Shattuck Avenue, Suite 510, Berkeley, CA 94720-1620, (510) 643-7201,
for commercial licensing opportunities.

Written by Matt Wright, The Center for New Music and Audio Technologies, University
of California, Berkeley.

IN NO EVENT SHALL REGENTS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL,
INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE
OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF REGENTS HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

REGENTS SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
SOFTWARE AND ACCOMPANYING DOCUMENTATION, IF ANY, PROVIDED HEREUNDER IS PROVIDED "AS
IS". REGENTS HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES,
ENHANCEMENTS, OR MODIFICATIONS.

SNNS-wrapper.c
Turn an SNNS-generated network-forward-pass C file into a Max object.

To change the SNNS file, do this:
1) Add the new .c file to the project
2) Under Edit / SNNSPPC Settings, go to the "PPC Target" panel and change the "File
Name" to whatever you want your external to be called.
3) Search for all the comments in this file that say "HEY!" and change the code.

*/

#define SNNS_VERSION "0.0"

/* the required include files */
#include "ext.h"

/* HEY!  put your .h file here: */
#include "rollins_mix-t1000.h"

/* HEY!  put the numbers of inputs and outputs here: */
#define NUM_INPUTS 2
#define NUM_OUTPUTS 60

/* structure definition of your object */

typedef struct SNNS
{
        Object o_ob;                            // required header
        void *outlet;
        float inputFloats[NUM_INPUTS];
        float outputFloats[NUM_OUTPUTS];
        Atom outputMaxList[NUM_OUTPUTS];
} SNNS;
```

```
void *SNNS_class;

/* prototypes  */
void SNNS_float(SNNS *x, float f);
void SNNS_int(SNNS *x, long n);
void SNNS_list(SNNS *x, Symbol *s, short argc, Atom *argv);
void SNNS_anything(SNNS *x, Symbol *s, short argc, Atom *argv);
void *SNNS_new(Symbol *s);
void SNNS_version (SNNS *x);
void SNNS_assist (SNNS *x, void *box, long msg, long arg, char *dstString);


/* initialization routine */

void main(fptr *f)
{
        setup((t_messlist **)&SNNS_class, (method)SNNS_new,0L, (short)sizeof(SNNS),
0L, 0);
        /* bind your methods to symbols */
        addmess((method)SNNS_anything, "anything", A_GIMME, 0);
        addmess((method)SNNS_list, "list", A_GIMME, 0);
        addmess((method)SNNS_assist, "assist", A_CANT, 0);
        addmess((method)SNNS_version, "version", 0);
        addint((method)SNNS_int);
        addfloat((method)SNNS_float);

        post("ADDAN2PAT and SNNS training by Alexander Refsum Jensenius. ");
        post("SNNS wrapping object version " SNNS_VERSION " by Matt Wright. ");
        post("Copyright © 2002 Regents of the University of California. All Rights
Reserved.");

}

/* instance creation routine */

void *SNNS_new(Symbol *s)
{
        SNNS *x;
        int i;
        x = newobject(SNNS_class);              // get memory for a new object &
initialize

        x->outlet = listout(x);

        for (i = 0; i < NUM_OUTPUTS; ++i) {
                x->outputMaxList[i].a_type = A_FLOAT;
                x->outputMaxList[i].a_w.w_float = -99999.9;
        }
        return (x);
}

void SNNS_version (SNNS *x) {
        post("SNNS wrapper version " SNNS_VERSION
                ", by Matt Wright. Compiled " __TIME__ " " __DATE__ );
}

/* I don't know why these aren't defined in some Max #include file. */
#define ASSIST_INLET 1
#define ASSIST_OUTLET 2

void SNNS_assist (SNNS *x, void *box, long msg, long arg, char *dstString) {
        if (msg==ASSIST_INLET) {
                sprintf(dstString, "List of network input values");
        } else if (msg==ASSIST_OUTLET) {
                sprintf(dstString, "List of network output values");
        } else {
                post("¥ SNNS_assist: unrecognized message %ld", msg);
        }
}

#define ATOM_AS_FLOAT(a)  (((a).a_type == A_LONG) ? ((float) (a).a_w.w_long) :
((a).a_w.w_float))

void SNNS_list(SNNS *x, Symbol *s, short argc, Atom *argv) {
        int i;
```

```c
        for (i = 0; i < argc; ++i) {
                if (argv[i].a_type == A_SYM) {
                        post("¥ SNNS: error: symbols are not allowed in the input
list.");
                        return;
                }
        }

        if (argc < NUM_INPUTS) {
                post("¥ SNNS: warning: network has %ld inputs but you sent only %ld
numbers.",
                        NUM_INPUTS, argc);
                post("  Setting extra network inputs to zero.");
                for (i = 0; i < argc; ++i) {
                        x->inputFloats[i] = ATOM_AS_FLOAT(argv[i]);
                }
                for (i = argc; i < NUM_INPUTS; ++i) {
                        x->inputFloats[i] = 0.0f;
                }
        } else {
                if (argc > NUM_INPUTS) {
                        post("¥ SNNS: warning: input list has %ld numbers, but
network has ", argc);
                        post("  only %ld inputs.  Ignoring extra list elements.",
NUM_INPUTS);
                }
                for (i = 0; i < NUM_INPUTS; ++i) {
                        x->inputFloats[i] = ATOM_AS_FLOAT(argv[i]);
                }
        }

        // post("*** Input list:");
        // for (i = 0; i < NUM_INPUTS; ++i) {
        //      post("   %f", x->inputFloats[i]);
        // }

        // post("*** calling network...");

        /* HEY!  Make sure the right function is being called here */
        project_trained_mix1000(x->inputFloats, x->outputFloats, 0);

        // post("*** done calling network.");

        for (i = 0; i < NUM_OUTPUTS; ++i) {
                x->outputMaxList[i].a_w.w_float = x->outputFloats[i];
        }

        outlet_list(x->outlet, 0L, NUM_OUTPUTS, x->outputMaxList);
}

void SNNS_anything(SNNS *x, Symbol *s, short argc, Atom *argv) {
        SNNS_list(x, s, argc, argv);
}

void SNNS_int(SNNS *x, long n) {
        Atom a[1];

        a[0].a_type = A_FLOAT;
        a[0].a_w.w_float = (float) n;

        SNNS_list(x, 0, 1, a);
}

void SNNS_float(SNNS *x, float f) {
        Atom a[1];

        a[0].a_type = A_FLOAT;
        a[0].a_w.w_float = f;

        SNNS_list(x, 0, 1, a);
}
```