Classification of Sound Textures

Nicolas Saint-Arnaud

Bachelor of Electrical Engineering
Université Laval, Québec, 1991
Master of Science in Telecommunications
INRS-Télécommunications, 1995

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Master of Science at the
Massachusetts Institute of Technology

September 1995

Author
Nicolas Saint-Arnaud
Program in Media Arts and Sciences
August 23, 1995

Certified by
Barry L. Vercoe
Professor, Program in Media Arts & Sciences
Thesis Supervisor

Accepted by
Stephen A. Benton
Chair, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Classification of Sound Textures

by

Nicolas Saint-Arnaud

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on August 23, 1995
in partial fulfillment of the requirements for the degree of
Master of Science

Abstract

There is a class of sounds that is often overlooked, like the sound of a crowd or the rain. We refer to such sounds with constant long term characteristics as Sound Textures (ST). This thesis describes aspects of the human perception and machine processing of sound textures.

Most people are not adept at putting words on auditory impressions, but they can compare and identify sound textures. We performed three perception experiments on ST's. A Similarity Experiment let us build a map where symbols that are close correspond to ST that are perceived similar. A Grouping Experiment provided groups of ST's that share a characteristic and a name; we call these groups P-classes. Finally, an Trajectory Map(Richards) Experiment points to some features used by subjects to compare ST's.

For machine processing, we look at two levels: local features, which we call Sound Atoms, and the distribution of Sound Atoms in time. In the implemented classification system, the low-level analysis is done by a 21-band constant-Q transform. The Sound Atoms are magnitude of points in the resulting time-frequency domain. The high level first builds vectors which encode observed transitions of atoms. The transition vectors are then summarized using a Cluster-based Probability Model (Popat and Picard). Classification is done by comparing the clusters of models built from different sound textures. On a set of 15 ST's, the classifier was able to match the samples from different parts of the same recording 14 times. A preliminary system to make a machine name the P-class of an unknown texture shows promise.

Thesis Advisor: Barry Vercoe
Title: Professor, Program in Media Arts and Sciences
Funding provided in part by the Television of Tomorrow Consortium.

Classification of Sound Textures

by

Nicolas Saint-Arnaud

Thesis Reader Rosalind W. Picard
Associate Professor
Program in Media Arts & Sciences

Thesis Reader Whitman A. Richards
Head
Program in Media Arts & Sciences

Acknowledgments

So many people had a tangible effect on this work that it seems impossible to thank them all and enough. Thanks to:

- my parents for their patience and support,
- the Aardvarks who put the fun in the work,
- all my friends, for the smiles and support,
- prof. W.F. Schreiber for continued support and interesting conversations, and Andy Lippman for a decisive interview,
- Linda Peterson, Santina Tonelli and Karen Hein without whom we would probably fall of the edge of the world,
- the Machine Listening Group, a fun bunch,
- my advisor, Barry Vercoe,
- my readers for their enthusiasm and helpful hints,
- Kris Popat and Roz Picard for the numerous and inspiring discussions and ideas for research topics, and for their help with the cluster-based probability model,
- Stephen Gilbert and Whitman Richards for their help with the Trajectory Mapping experiments,
- Dan Ellis, for tools, figures and sound advice,
- all who volunteered as human subjects,
- and all the people at the Media Lab who make it what it is.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

This thesis is about human perception and machine classification of sound textures. We define a sound texture as an acoustic signal characterized by its collective long-term properties. Some examples of sound textures are: copy machine, fish tank bubbling, waterfall, fans, wind, waves, rain, applause, etc.

1.1 Motivation

Sound textures are an interesting class of sounds, yet different from the other classes of sound studied by most researchers. Contrary to speech or music, sound textures do not carry a "message" which can be decoded. They are part of the sonic environment: people will usually identify them and extract relevant information (their meaning in the current context) and then forget about them as long as the sonic environment does not change.

There currently exists no meaningful way to describe a sound texture to someone other than have them listen to it. The vocabulary to qualify sound textures is imprecise and insufficient, so humans tend to identify them by comparison to a known sound source ("it sounds like a motor, like a fan, like a group of people"). This method of qualification does not transpose easily to machine classification.

Some current models for analysis of sounds by machine extract features like harmonics that occur in a deterministic fashion: the sound is defined as the specific features in the specific order. These models are too rigid for sounds that are stochastic in nature, sounds that have no definite start or end. There are other deterministic analysis methods like embedding, stochastic methods like Markov models, models that assume sound is filtered white noise, and so on. These methods are not well suited for analysis of sound textures.

Our analysis of sound textures assumes that sounds are made up of sound atoms (features) that occur according to a probabilistic rule. Under this assumption, analysis involves extracting the sound atoms and building a probabilistic model. These probability models are high-order statistics, and classification with those models is not a trivial task. Feature selection is an important part of the design of the analyzer; sound atoms should carry perceptual meaning and should aggregate data to reduce the complexity of the probabilistic model.

We devised some psycho-acoustic experiments to find out into which classes people sort sound textures. We will later compare human classification with machine classification.

1.2 Overview

Chapter two summarizes what we found about human perception of sound textures. Chapter three explains the machine classification scheme. Chapter four gives results for machine classification. Chapter five suggests some areas of further research in sound textures.

Our work with human perception of sound textures was roughly divided in two parts: first an informal part with small "focus groups", then some more formal psycho-acoustic experiments. We formed small focus groups to come up with examples, qualifiers, classes, perceived parameters and properties of sound textures. We did brainstorms, group discussions and listening sessions. Then we ran three psycho-acoustic experiments: a similarity experiment to see which sounds are perceived similar to each other, a classifying experiment to find preferred groupings of sound textures, and an ordering experiment to find the relations between textures. Finally, classes of textures that are considered to belong together by most subjects are formed using the groupings observed in the experiments. We call those human classifications P-classes to distinguish them from the machine classifications to be presented later. Each P-class has a name that should be indicative enough that most subjects would make the same groupings when given only the P-class name and a set of unidentified textures.

The machine analysis system first extracts sound atoms from textures, and then builds a probabilistic model of the transitions of atoms. Atom selection is critical: atoms must be perceptually salient features of the sound, but they must not make too strong assumptions about the kind of sound at the input. The system currently uses points of the magnitude of a spectrogram as sound atoms. Vectors containing observed transitions of the parameters of atoms are then formed, and these transition vectors are clustered to extract the most likely transitions. Classification is done by comparing the likely transitions of unknown textures with the transitions stored in templates. The templates are formed by analyzing typical textures from one P-class.

Samples from twelve textures are used to train the machine classifier. Each sample is assigned one or more of the six P-classes labels: periodic, random, smooth-noise, machine, water and voices. Fifteen samples are used to test the classifier, twelve of which are taken from different segment of the textures used for training, and three samples from new textures. The dissimilarity measure is consistent for different samples of the same texture. The classifier clearly discriminates its training textures. For the testing set used, the classification done by machine was the same as the classification into P-classes in more than 85% of the cases.

1.3 Background

The first concept in our planned work is modeling a complex signal using a representation that addresses the underlying cognitive structure of the signal rather than a particular instance of the signal. A lot of work has been done on modeling, most of which assumes some knowledge of the signal source. Examples of physical modeling of sound sources can be found in [SCH83], [SMI83], [SMI86] and [GAR94].

The probabilistic model used in this thesis does not assume such knowledge about the source, at least at a high level; it can encode arbitrary transitions of any features of a signal. It does assume stationarity of the signal in a probabilistic sense, which we equate to the perceptual similarity of any time portion of the texture. With our proposed method the (physical) modeling is replaced by a high-dimensional probability density estimation. This is a newer field, but some of the groundwork has been done [SCO83][SIL86]. For the probabilistic modeling, we will use the work of Popat and Picard [POP93] as a starting point.

Sound textures are not new, but they have received less attention than other classes of sounds, for example timbres. In fact, most studies treat sounds as specific events limited in time, not as textures. One problem with researching sound textures is that they are referenced under many other names. Our library search for related work in the library has met with very little success.

Feature selection for sound is a very wide topic, which can be (and has been) the subject of countless theses. The human cochlea is known to do a log-frequency decomposition [RIC88], p.312

and [MOO92]. This provides a good starting point for feature selection, given what is known now. We will explore a few alternatives, like simple forms of filtering [VAI93] for preprocessing the sampled (PCM) signal.

Using a high-level model to characterize the transitions of low-level features in sound is not entirely new; it has been done to some extent for breaking, bouncing and spilling sounds [GAV94][WAR84]

Connection to Other Groups at the Media Lab

The cluster-based probability model used for machine analysis was developed at the Media Lab by Picard and Popat of the Vision and Modeling group. It has been used to classify and resynthesize visual textures. The cluster-based probability model has also been shortly explored for modeling timbres by Eric Metois (music group) [MET94].

Whitman Richards developed the Trajectory Mapping (TM) technique, and used it to show paths between visual textures. Stephen Gilbert used TM on musical intervals. Whitman Richards expressed interest in mapping from visual textures to sound textures [RIC88]. Michael Hawley has addressed the problem of classification of sounds in his Ph. D. Thesis [HAW93].

In the Machine Listening group, Dan Ellis worked intensively on extracting low-level features from sounds and grouping them [ELL94]. Michael Casey works on physical modeling for sound production [CAS94]. Eric Scheirer has done some work on probabilistic modelling of transitions of musical symbols.

1.4 Applications

Sound texture identification is an important cue for the awareness of the environment. Humans and animals constantly listen to the ambient texture and compare with similar textures previously experienced. Sound textures can have a strong impact on the emotions of the listener (e.g. fear) and can also influence their actions (the sound of rain prompts to avoid getting wet, the sound of voices to prepare for social contact...)

Similarly, machines can get information about the environment by listening to sound textures. Autonomous creatures (real or virtual) can use this information to direct their actions.

Successful models for machine representation and classification of sound textures could help us understand how humans perceive and classify a wide range of complex, changing sounds.

Sound textures are almost always present in movies, in one form or another. This makes sound texture identification a privileged tool in sound track annotation. The sound texture classifier might also be trained to recognize the presence of other classes of signal, and signal other systems to start acting: for example the detection of speech may be made to trigger a speech recognizer and a face recognition program.

An optimal method for analysis of a sound texture should produce a representation that is small compared to the original sampled sound. Furthermore, the extracted model can be used to generate new textures of indefinite length. This points to very efficient compression. Just as speech can be compressed to the text and music to a MIDI stream, ambiance could be compressed to a compact representation [GAR94].

Creating and modifying sound textures would be of use for sound track ambiance creation, or virtual environments. Addition of a surrounding sound ambiance could add information to a user interface in a non-monotonous way. The ability to modify a texture using semantic parameters greatly increases the usability of the texture editor. Interpolation between two or more textures could produce new textures that combine some of the perceived meaning of the original textures, possibly in an unnatural but interesting fashion (e.g. a voice engine: voice phonemes with an engine-like high-level structure).

1.5 Summary of Results

Human Perception

We performed some perception experiments which show that people can compare sound textures, although they lack the vocabulary to express formally their perception.

We have also seen that the subjects share some groupings of sound textures. There seem to be two major ways used to discriminate and group textures: the characteristics of the sound (e.g. periodic, random, smooth) and the assumed source of the sound (e.g. voices, water, machine).

Machine Classification

The machine classifier we implemented is able to accurately identify samples of the same sound texture for more that 85% of the samples. The dissimilarity measure is smaller between periodic sounds, and between random sounds, which agrees with human perception of similarity. Frequency content of the samples also has an influence on classification.

Chapter 2 Human Perception of Sound Textures

Chapter Overview In the first section of this chapter we summarize our early efforts at collecting examples of Sound Textures and finding their perceived qualifiers and parameters. We proceed to propose a definition of sound textures to be used in the scope of this work. Section 2.3 introduces the three psychoacoustic experiments on similarity (§ 2.4), grouping (§ 2.5) and ordering (§ 2.6) of Sound Textures, and report findings. Finally, we present a simple set of classes for Sound Textures based on the human experiments, which we call P-classes.

2.1 What do You Think is a Sound Texture?

The first logical step in our research on Sound Textures was to find out how people would characterize and define them. For this we had many informal meetings with "focus groups" of four to six people of different background, where we would brainstorm about sound textures, make several listening tests, discuss and record the interventions. The concept of a sound texture was at first very difficult to pin down, and discussing in groups brought many simultaneous points of view on the subject. One of the first suggestions was to look in the dictionary.

2.1 What do You Think is a Sound Texture?

Dictionary Definition

The on-line Webster dictionary has many interesting definitions for texture:

- 1a: something composed of closely interwoven elements
- 1b: the structure formed by the threads of a fabric
- 2b: identifying quality: CHARACTER
- 3: the disposition or manner of union of the particles of a body or substance
- 4a: basic scheme or structure: FABRIC
- 4b: overall structure: BODY

From definitions 1a and 3, we could propose that sound textures are composed of closely interwoven sound particles. Definition 1b and 4 suggests the existence of a structure. Definition 2b suggests that the texture helps to identify a sound. These pointers are surprisingly adequate for a lot of our work on sound textures. In fact, the three themes (sound particles, structure, identification) are key concepts in this thesis.

2.1.1 Examples

On the first meeting the groups were asked to give examples of sound textures. The format was a brainstorm, where the ideas were written down on the board, and discussion was not allowed at first, to let every one speak with spontaneity. On a second pass, some of the sounds were rejected from the "Texture" list, and some other items were added to the "Not a Texture" list. Table 2.1.1 is a compilation of results.

TABLE 2.1.1 Brainstorm: Examples of Sound Textures

Texture		Not a Texture
rain	running water	one voice
voices	whisper	telephone ring
fan	jungle	music
traffic	crickets	radio station
waves	ice skating	single laugh
wind	city ambiance	single hand clap
hum	bar, cocktail	sine wave
refrigerator	amplifier hum	
engine	60 Hz	
radio static	coffee grinder	
laugh track (in TV show)	bubbles	
applause	fire	
electric crackle	whispers	
babble	snare drum roll	
murmur	heart beat	

Texture or Not?

One very dynamic activity of the focus groups was an informal experiment where the author would play a lot of different sounds and ask the group whether each can be called a texture. This experiment was very important in showing that each person has specific criteria of what is a texture, and those criteria can vary quite a lot from one person to the other.

2.1.2 Perceived Characteristics and Parameters

The groups were then asked to come up with possible characteristics and qualifiers of sound textures, and properties that determine if a sound can be called a texture. Sorting the answers produced the following tables, split into a few loose conceptual classes: characteristics, qualifiers, other determinants and properties. Characteristics (usually nouns) apply to all textures to some degree; Qualifiers (usually adjectives) may or may not apply at a certain degree to each texture; Other Determinants have a strong impact on how the sound is produced or perceived; Properties are required of all sound textures. Table 2.1.2 show the first three lists. Lists can overlap: for example, the characteristic randomness can be seen as an axis, and the qualifier random points to the high end of that axis.

TABLE 2.1.2 Brainstorm: Sound Texture Characteristics, Qualifiers and Other Determinants

Characteristic	Qualifier		Other Determinant
volume	loud	is machine	spectrum content
randomness	repetitive	pitched	"tone color"
regularity	random	noisy	granular shape
periodicity	voiced	environmental	"could be produced with this model"
frequency	chaotic	ambient	
smoothness	steady	periodic	
irritability	dangerous	harsh	
grain size	quiet	human	
time scale	pleasant	granular	
period length	is voice	rhythmic	
brightness	rough	smooth	
density	man-made	discrete	
complexity	natural	continuous	
contrast	is water	"has sharp onsets"	
spaciousness	annoying	violent	
size of source			
energy			
force			

There is obviously a lot of overlap within the Characteristics, and also between the Qualifiers and Characteristics. If we consider Characteristic to be akin to axes of the texture space, they could hardly form an orthogonal basis. Furthermore a lot of them are highly subjective, which removes even more of their usability as a basis.

A few of the more subjective characteristics and qualifiers are specially interesting. "Annoying" is a qualifier that surfaced often when participants were listening to recorded textures; it was often associated with a strong periodicity, a sharp volume contrast, and the presence of high frequencies. However, sounds qualified "annoying" could have all kinds of other qualifiers; they were not confined to a specific sub-space of the Characteristic "axes" either. Similar remarks could be made about most of the other subjective qualifiers: environmental, pleasant, violent, natural, etc.

Another preeminent qualifier is the sensation of danger. It was often associated with a large estimated source size, which in turn was associated with the presence of low frequencies. Listeners were rather unanimous in their evaluation of what textures sounded dangerous (more unanimous than for almost all other appreciations on sound textures). The "danger" sensation from sound seems to be wired in the human brain, as it is probably for all animals with hearing capabilities.

This brings us to a very important observation: *sound textures carry a lot of emotion.* With the exception of smell, they are probably the strongest carrier of emotions for human beings; even in speech or music the emotional content is not so much dependant on words or melodies but to a great extent on a "mood" more linked to the "textural" aspect than the semantics.

Properties

A few properties were mentioned as essential for a sound to be a texture; they are summarized in Table 2.1.3. They will be used to form a working definition of sound textures in Section 2.2.

TABLE 2.1.3 Brainstorm: Properties of Sound Textures

cannot have a strong semantic content
"ergodicity": perceptual similarity of any (long enough) segment in time
long time duration

Parallel with Visual Texture Parameters

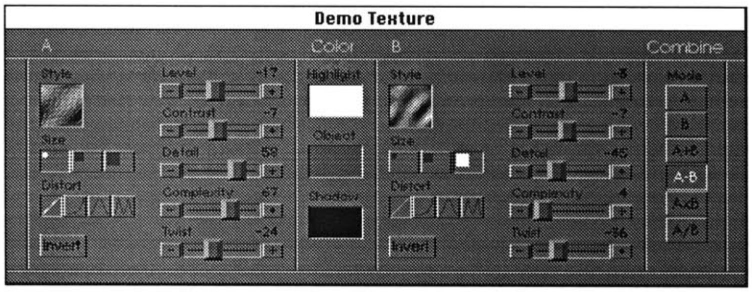
Some of the perceived characteristics of sound textures have obvious parallels in visual textures: periodicity and period length, complexity, randomness, granular shape, and "could be produced with this model". Sound volume can be associated with visual intensity, and visual contrast with volume contrast. Color is in some ways similar to sound frequency content (sometimes referred to as tone color). Visual orientation is difficult to equate with a sound characteristic, although some sounds are characterized by a shift in frequency content with time. In a fishtank bubble texture, for example, each bubble starts with a low frequency moving upwards (see Figure 4.1.4), so that upwards or downwards movement in frequency could be associated with visual orientation, although this effect in sound is less important than in vision.

Even some of the more subjective qualifiers, like natural, annoying, man-made and annoying have parallels in the visual domain.

In the course of our texture exploration, we also experimented with two pieces of Macintosh software for synthesizing visual textures. The first KPT Texture Explorer ™ obviously has some internal model of visual textures, with an unknown number of parameters, but the user controls are very limited: at any point one can only choose only one of 12 textures shown, choose a mutation level, and initiate a mutation which produces 12 "new" textures. There is also a control for mutating color. Despite the poorness of the controls, the program includes a good selection of seed textures, which helps to get visually interesting textures.

The second program, TextureSynth ™ Demo, is more interesting in the way it lets users modify textures. It offers 12 controls, described on Table 2.1.4. The control window has two sets of 9 controls (sliders, buttons and pop-up menus) to produce two textures (A and B) which are then combined (A,B, A+B, A-B, AxB, A÷B). The results are visually interesting, and the controls are usable. Still, it is obvious that modifying textures is a difficult problem, both from the synthesis model point of view and the user control point of view.

FIGURE 2.1.1 TextureSynth ™ Demo Control Window



Demo Texture

A: Level -17, Contrast -7, Detail 53, Complexity 67, Twist -24

B: Level -5, Contrast -7, Detail -45, Complexity 4, Twist -56

There are many parallels between the controls in TextureSynth ™ Demo and possible parameters of sound textures. The last column of Table 2.1.4 points to some possible choices.

TABLE 2.1.4 TextureSynth ™ Demo Controls

Name	Type	Description	Parallel with Sound Textures
Style	popup: 8 choices	basic style: plaster, waves, fiber, plaid, etc.	basic class: machine, voices, rain, etc.
Size	buttons: 3 choices	fine, medium or coarse grain	time scale
Distort	buttons: 3 choices	lookup table from function intensity	–
Invert	on/off button	reverses intensity scale	–
Level	slider	mid-point of intensity	volume
Contrast	slider	dynamic range of intensity	volume contrast
Detail	slider	low/high frequency ratio	number of harmonics, low/high frequency ratio
Complexity	slider	randomness	randomness
Twist	slider	orientation	–
Shadow	popup: 25 choices	color mapped to low intensity	Frequency contents
Object	popup: 25 choices	color mapped to medium intensity	
Highlight	popup: 25 choices	color mapped to high intensity	

2.1.3 Classes and Categories

The participants in one focus group had many reservations about whether it was possible, or even desirable, to come up with classes or categories of sound textures. One foreseen difficulty was that there are many possible kinds of classes:

- by meaning
- by generation model
- by sound characteristics

Also, people had great difficulties in coming up with names for categories without first having an explicit set of textures to work with. This would support views that for one class of signals, people use many different models and sets of parameters, and that identification is context-dependent.

Because of those difficulties, we decided to do classification experiments in which a set of sound textures are provided; this is described in the rest of this chapter. The first obvious problem was to try to collect a set of textures that would well represent the space of all possible textures. It is not possible to do an exhaustive search of this infinitely complex space, so we tried to collect as many examples as possible and then limit the redundancy. The examples were taken by asking many people, by recording ambient sounds, and by searching databases of sounds. Still, any collection of sound textures is bound to be incomplete.

2.2 Working Definition of Sound Textures

Too Many Definitions

During the focus group meetings, it quickly became obvious that there are many definitions of a sound texture, depending on whom you ask. However, most people agreed on a middling definition which included some repetitiveness over time, and the absence of a complex message.

John Stautner suggests that textures are made of "many distinguishing features, none of which draws attention to itself. A texture is made of individual events (similar or different) occurring at a rate lower than fusion; using an analysis window, we can define a texture as having the same statistics as the window is moved." [STA95]

This would seem a reasonable definition, but it is made difficult because it depends on many variable concepts: "distinguishing", "drawing attention", fusion rate, window size, etc. The concepts of "distinguishing" and "drawing attention" are totally subjective. The acceptable range for fusion rate and window size are once again variable.

When asked to draw the boundary for fusion, participants in the focus groups all agreed that the rate was lower than 30Hz. As for the window size, the upper boundary was limited by an "attention span" of a few (1-5) seconds, with the argument that events too far apart are heard as independent of each other.

2.2.1 Working Definition of Sound Texture

A natural step at this point was to refine a definition for Sound Texture to be used in the scope of this work.

Defining "Sound Texture" is no easy task. Most people will agree that the noise of fan is a likely "sound texture". Some other people would say that a fan is too bland, that it is only a noise. The sound of rain, or of a crowd are perhaps better textures. But few will say that one voice makes a texture (except maybe high-rate Chinese speech for someone who does not speak Chinese).

First Time Constraint: Constant Long-term Characteristics

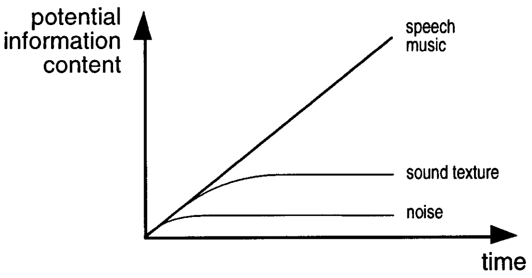
A definition for a sound texture could be quite wide, but we chose to restrict our working definition for many perceptual and conceptual reasons. First of all, there is no consensus among people as to what a sound texture might be; and more people will accept sounds that fit a more restrictive definition.

The first constraint we put on our definition of a sound textures is that it should exhibit similar characteristics over time; that is, a two-second snippet of a texture should not differ significantly from another two-second snippet. To use another metaphor, one could say that any two snippets of a sound texture seem to be cut from the same rug [RIC79]. A sound texture is like wallpaper: it can have local structure and randomness, but the characteristics of the structure and randomness must remain constant on the large scale.

This means that the pitch should not change like in a racing car, the rhythm should not increase or decrease, etc. This constraint also means that sounds in which the attack plays a great part (like many timbres) cannot be sound textures. A sound texture is characterized by its sustain.

Figure 2.2.1 shows an interesting way of segregating sound textures from other sounds, by showing how the "potential information content" increases with time. "Information" is taken here in the cognitive sense rather then the information theory sense. Speech or music can provide new information at any time, and their "potential information content" is shown here as a continuously increasing function of time. Textures, on the other hand, have constant long term characteristics, which translates into a flattening of the potential information increase. Noise (in the auditory cognitive sense) has somewhat less information than textures.

FIGURE 2.2.1 Potential Information Content of A Sound Texture vs. Time



Sounds that carry a lot of meaning are usually perceived as a message. The semantics take the foremost position in the cognition, downplaying the characteristics of the sound proper. We choose to work with sounds which are not primarily perceived as a message.

Note that the first time constraint about the required uniformity of high level characteristics over long times precludes any lengthy message.

Two-level Representation

Sounds can be broken down into many levels, from a very fine (local in time) to a broad view, passing through many groupings suggested by physical, physiological and semantic properties of sound. We choose, however, to work with only two levels: a low level of simple atomic elements distributed in time, and a high level describing the distribution in time of the atomic elements.

We will bring more justification for this choice in Chapter 3, when we talk about our definition of sound textures for machine processing.

Second Time Constraint: Attention Span

The sound of cars passing in the street brings an interesting problem: if there is a lot of traffic, people will say it is a texture, while if cars are sparse, the sound of each one is perceived as a separate event. We call "attention span" the maximum time between events before they become distinct. A few seconds is a reasonable value for the attention span.

We therefore put a second time constraint on sound textures: high-level characteristics must be exposed or exemplified (in the case of stochastic distributions) within the attention span of a few seconds.

This constraint also has a good computational effect: it makes it easier to collect enough data to characterize the texture. By contrast, if a sound has a cycle of one minute, several minutes of that sound are required to collect a significant training set. This would translate into a lot of machine storage, and a lot of computation.

Summary of our Working Definition

- Our sound textures are formed of basic sound elements, or atoms;
- atoms occur according to a higher-level pattern, that can be periodic or random, or both;
- the high-level characteristics must remain the same over long time periods (which implies that there can be no complex message);
- the high-level pattern must be completely exposed within a few seconds ("attention span");
- high level randomness is also acceptable, as long as there are enough occurrences within the attention span to make a good example of the random properties.

2.3 Experiments with Human Subjects

We conducted three sound texture classification experiments, each with 4-8 subjects. The experiments were approved by the MIT Committee on the Use of Human Experiment Subject (COUHES, request 2258).

Goal

In doing experiments with human subjects, we want to confront the subjects with actual sound textures and explore their reactions. We first want to see what kind of parameters are actually used by naive subjects to sort the sound textures. The second important goal is to find the kinds of groupings that subjects make, and how they "label" these groupings. This last information was then used to build P-Classes, which are the subject of a further section.

2.3.1 Common Protocol

The interface is based on the Macintosh finder, without any special programming. Sound textures are each identified with an icon, and can be played (double-click), moved (drag) and copied (option-drag). Each experiment trial is self-contained in a window, with the required texture icons in the right place. Special care is taken to insure that windows show up on the right place on the screen, and don't overlap.

The sounds are played by SoundApp (Freeware by Norman Franke). The stimuli were taken from the Speed of Sound sound effect compact disk.

Shortcomings

Using the Finder greatly reduces the time required to set the experiments up, but it also reduces the flexibility of the system as compared to programmed systems. The main drawbacks are:

- the data is not automatically collected in machine form,
- copying texture icons can be awkward, and
- playback cannot be robustly interrupted.

2.4 Similarity Experiment

Goal

In the first experiment we measure the perception of similarity between sound textures and form a map where the distance between any two textures is roughly inversely proportional their similarity. The resulting map should bring together the textures which are perceived similar, if it is possible on a two-dimensional map. Because MDS is an optimization procedure that tries to achieve a best fit, the distance on the map may not always reflect the perceived similarity.

2.4.1 Method: Multidimensional Scaling (MDS)

Multidimensional Scaling is a well-known method to build a low-dimensional map of the perceived distances between a set of stimuli [JOH82]. It is based on an iterative algorithm to optimize the relative position of the stimuli on the map to match to the perceived similarities or dissimilarities.

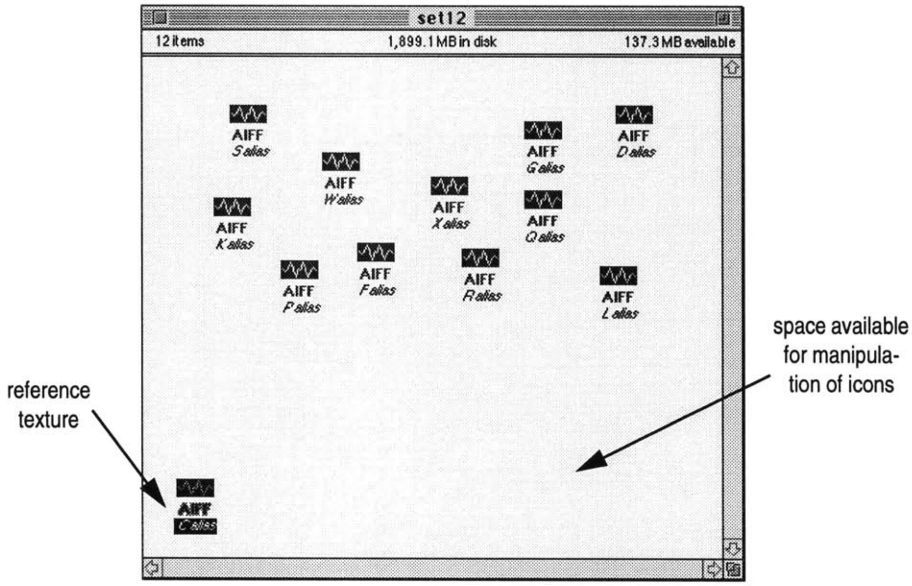
In our experiment, we collect rank-ordered perceived similarities (as opposed to perceived distances). For example, with a set of 12 textures, we give a similarity rank of 11 to the texture perceived the closest to the reference texture, then a 10 for the next closest, and so on until the subject thought the remaining textures are perceived as not similar (rank 0). This measure of perceived similarity is valid for ordering but should not be taken as an absolute measurement.

These similarities are successively collected with each texture as a reference to form a similarity matrix. The similarity matrix is then fed to the well known Kyst2a program from AT&T [KRU77]. The data is specified to be non-metric similarities. We request a two-dimensional map, so Kyst produces a set of (x,y) coordinates for each texture.

Protocol

In this experiment, the subject is presented with a small set of sound textures which remains the same throughout all the trials of the experiment. As a trial begins, a reference texture is played and the subject must try to find the most similar texture in the rest of the set. The subject then looks for the next texture most similar to the reference texture and so on, until it is felt that the remaining textures are completely different from the reference texture.

FIGURE 2.4.1 Trial Window for the Similarity Experiment



space available for manipulation of icons

reference texture

There are as many trials as there are sounds in the set, so the subject gets to match each texture. There is a different window for each trial, and each window contains icons for all the sounds from the set, in random positions. Each icon is identified by a capital letter. These letters have no meaning, they are for identification only. The icon for the reference texture is always in the lower left corner of the window, and the others at the top of the window (see Figure 2.4.1). The subject listens to a sound by double-clicking on its icon, and can stop the playback by typing a dot (".") before clicking on anything else. A data sheet is provided to note the results at the end of each set. The icons can be moved around the window to help organize the sounds; only the data sheet is used to collect results, not the final window configuration.

The set contains the 12 textures shown on Table 2.4.1. The signals were scaled to have a maximum dynamic range, but the loudness perception varied. The sound textures are each identified by a random capital consonant. The experiment took an average of 35 minutes to complete.

TABLE 2.4.1 Set of Stimulus for the Similarity Experiment

identifier	name	description
C	bubbles	fish tank bubbles
D	honda	idling motorcycle
F	traffic	many individual traffic horns
G	stream	quiet water stream or brook
K	crickets	constant drone of many crickets outside
L	helicopt	closed-miked constant helicopter rotor
P	whispers	many people whispering, with reverb
Q	rain	constant heavy rain
R	air_cond	air conditioning compressor & fan
S	snare	constant snare drum roll
W	crowd	medium-sized crowd inside, with reverb
X	wind	constant wind outside

2.4.2 Results

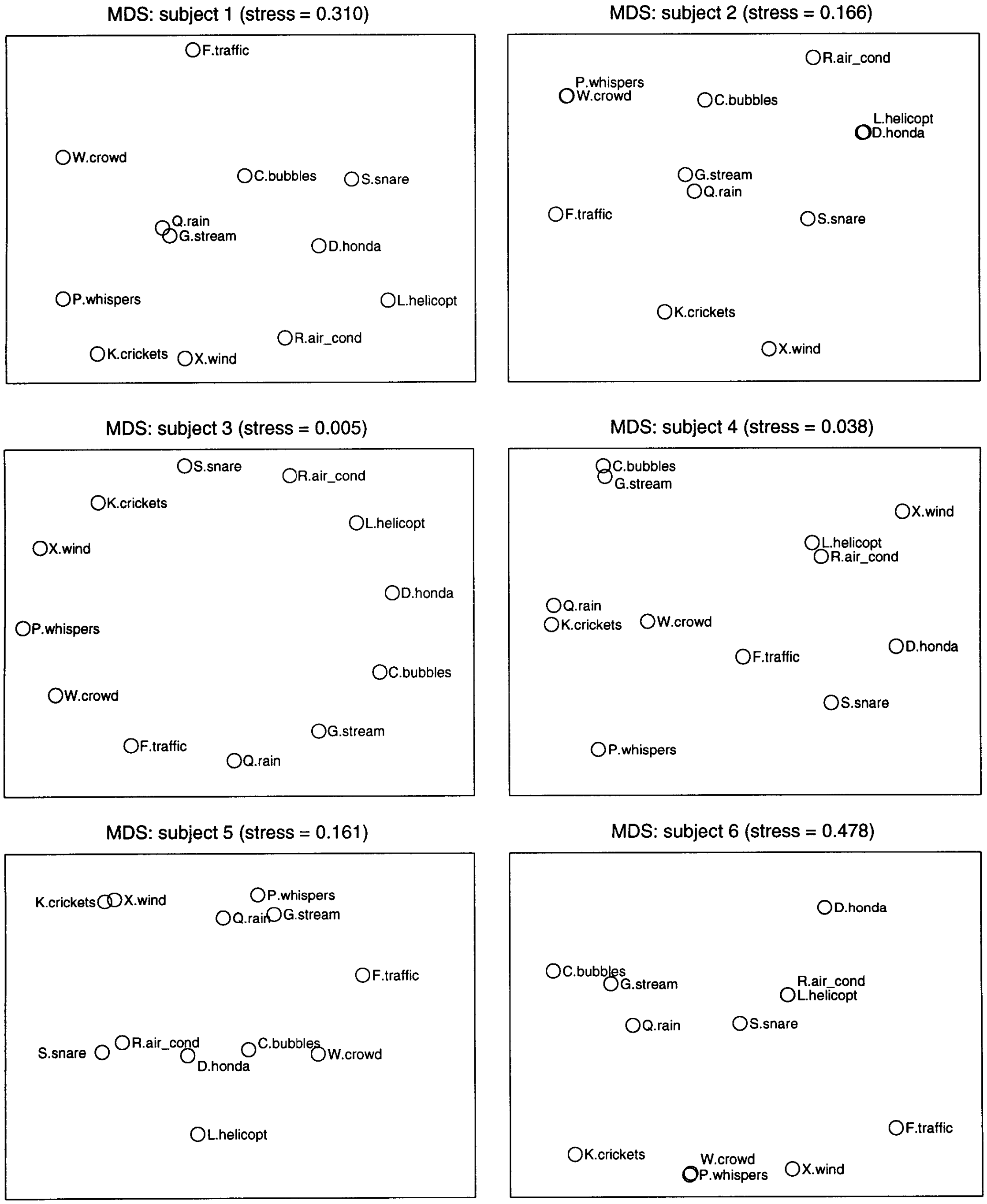
The similarity perception data was collected from the trials into a similarity matrix, as in Table 2.4.2. To visualize the data, MDS maps are formed using the kyst algorithm. The two dimensional maps are shown on Figure 2.4.2.

TABLE 2.4.2 Similarity Matrix for Subject 1

	bub	hon	tra	str	cri	hel	whi	rai	A/C	sna	cro	win
C.bubbles	12	9	0	11	6	7	0	10	5	8	0	0
D.honda	9	12	0	7	5	10	6	3	11	8	0	4
F.traffic	6	11	12	8	0	0	0	7	0	9	10	0
G.stream	10	0	0	12	8	0	7	11	6	4	5	9
K.crickets	0	0	0	9	12	0	11	8	7	0	6	10
L.helicopt	7	11	0	6	0	12	0	5	10	9	0	8
P.whispers	0	0	0	8	9	0	12	7	6	0	11	10
Q.rain	8	0	0	11	9	0	10	12	5	6	4	7
R.air_cond	3	8	0	7	6	10	2	5	12	11	4	9
S.snare	6	9	0	7	0	11	0	8	10	12	0	0
W.crowd	7	0	10	9	6	0	11	8	0	0	12	5
X.wind	0	0	0	8	10	0	11	7	9	5	6	12

For all six subjects, four sounds (helicopter, air conditioning, motorcycle and snare) formed a localized group, with no other sound mixed in. Four subjects brought bubbles close to that latter group. The three sounds produced by water (bubbles, stream and rain) were also in a loose local group for all six subjects. Subjects 2 and 6 gave a high similarity to whispers and crowd, and three other subjects con-

FIGURE 2.4.2 Individual MDS Results (Subjects 1, 2, 3, 4, 5 and 6)



MDS: subject 1 (stress = 0.310)

MDS: subject 2 (stress = 0.166)

MDS: subject 3 (stress = 0.005)

MDS: subject 4 (stress = 0.038)

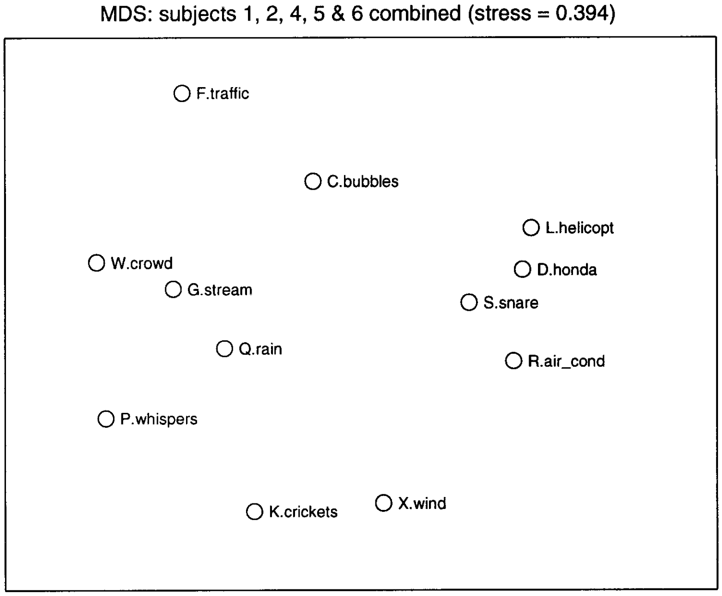
MDS: subject 5 (stress = 0.161)

MDS: subject 6 (stress = 0.478)

sidered them rather similar. Four people considered crickets and wind similar. Crowd is a neighbor to traffic in five of the MDS results.

Not surprisingly these observations apply to the MDS analysis of the combined data, shown on Figure 2.4.3. Note that the data for subject 3 was not included because the experiment protocol was not followed as closely in that case, as discussed in the next paragraph.

FIGURE 2.4.3 Combined MDS Results for Subjects 1, 2, 4, 5 & 6



MDS in an approximation procedure which minimizes a measure of stress. In this case, the stress indicates the deviation of the distances on the map from the distances computed from the input data. Because the input data has many more points than there are dimensions in the mapping space, the stress for the final map will not be zero. The final stress for the MDS maps are indicated at the top of the figures. They range from 0.005 to 0.478. A map with stress below 0.1 is considered to be a good fit, so only subjects 3 and 4 meet that criterion. Subjects 2 and 5 have a slightly higher but acceptable fit. Subjects 1 and 6 have a high stress which indicates an improper fit. In those cases, increasing the dimensionality of the mapping space can help produce a better fit. Table 2.4.3 shows the final stress for 2- and 3-dimensional mapping of the subject data.

TABLE 2.4.3 Final Stress of the MDS Mappings

	final stress	
subject	**2-D**	**3-D**
1	0.310	0.207
2	0.166	0.100
3	0.005	0.010
4	0.038	0.006
5	0.161	0.102
6	0.478	0.398
subjects 1, 2, 4, 5 & 6 combined	0.394	0.264

In general, the final stress is reduced by mapping onto three dimensions, but not enough to match our criterion for a "good" fit (stress<0.1) in the difficult cases (subjects 1, 6 and combined map). This may indicate that the data for subjects 1 and 6 is less consistent across trials. It is expected that the final stress for combined map is higher because the combination is bound to introduce variance and possibly inconsistencies. Figure 2.4.4 shows 3-dimensional MDS mappings for subjects 1 and 6; Figure 2.4.5 shows such a mapping for the combined data. The plots show a perspective view of the points in 3-D space, as well as a projection of the points on the (x,y) plane.

FIGURE 2.4.4 Three-Dimensional MDS Mappings (subjects 1 and 6)

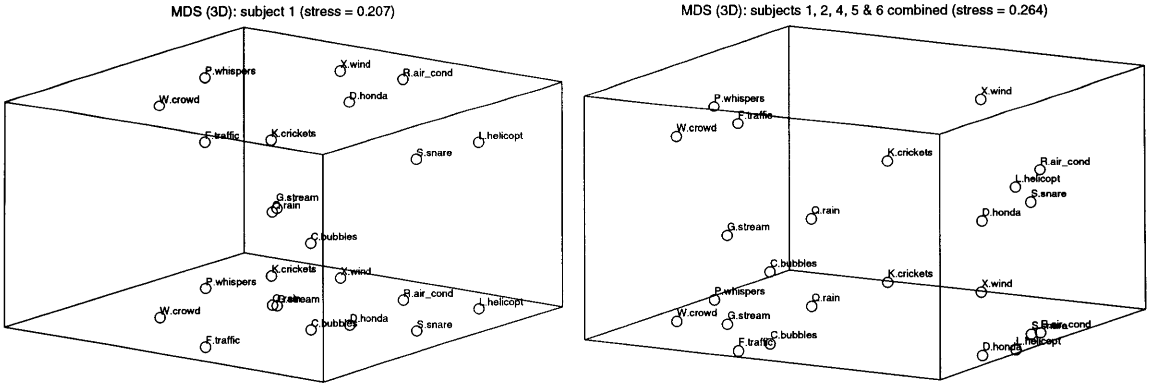
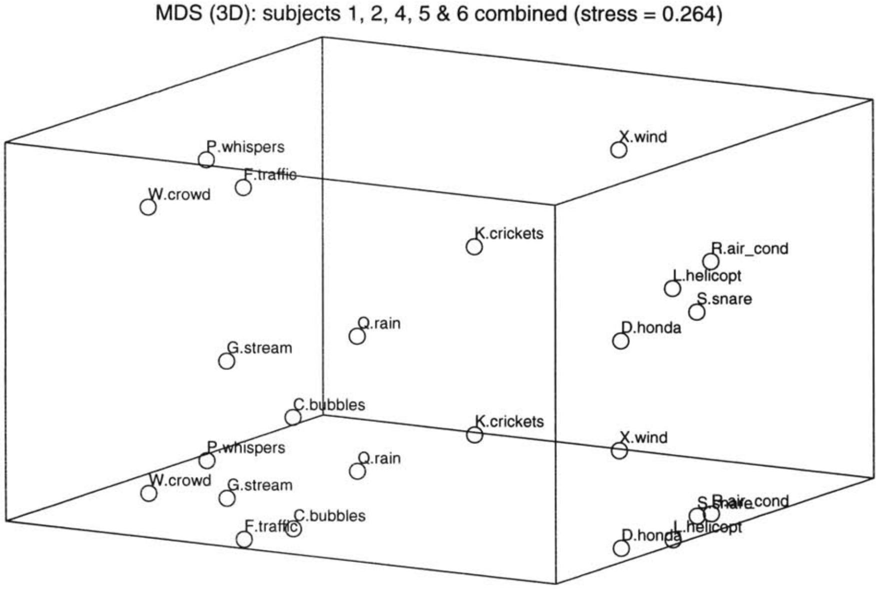


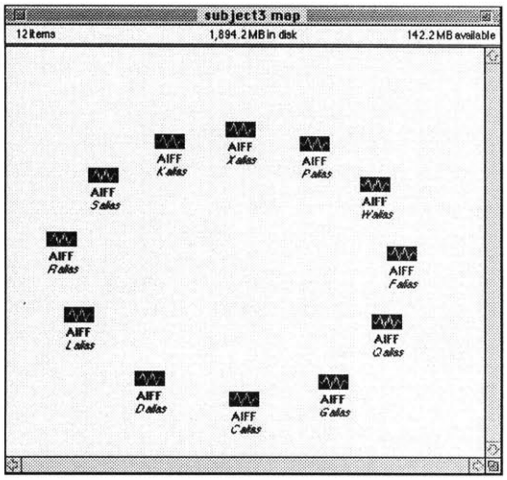
FIGURE 2.4.5 Three-Dimensional MDS Mappings (Subjects 1, 2, 4, 5 & 6 Combined)

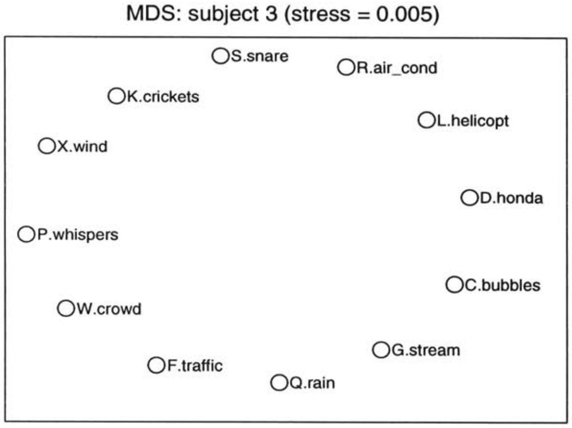


Proof of Principle of the MDS Algorithm

An interesting case should be brought to your attention: Subject 3 first ordered the textures in a circle (Figure 2.4.6, left side), and then used that ordering to rank the similarities. The MDS algorithm extracted exactly the same order (Figure 2.4.6, right side), which shows that it works as expected. Note that the MDS algorithm has a tendency to arrange data points in a circle, which helped conserve the shape.

FIGURE 2.4.6 Proof of Principle of the MDS Algorithm





2.5 Grouping Experiment

Goal

The grouping experiments aimed at discovering what sounds people think go together, and what names they give to the various groups they make. One question of interest was: "Do people group sound textures by their origin, or do they group them by common sound characteristics?".

We hoped to find a common, natural grouping scheme (P-classes), and later compare the computer classification results with P-classes.

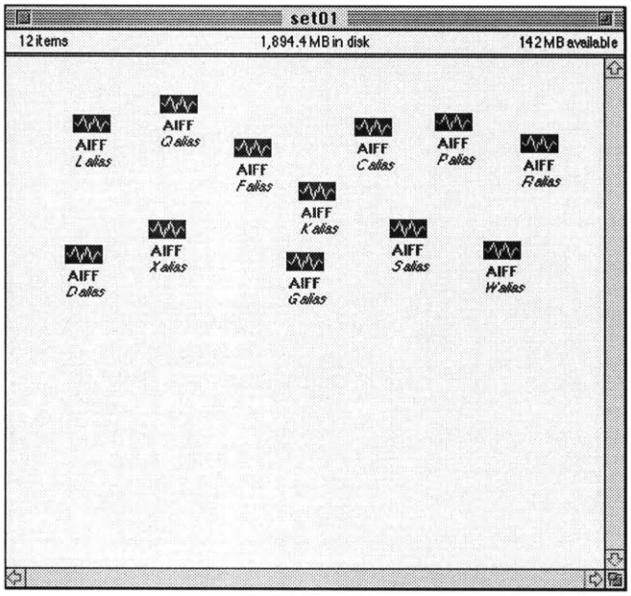
2.5.1 Protocol

In the grouping experiment, the subject is asked to cluster the textures from a small set into a few groups. The requirements for the clusters are intentionally left vague:

> "Cluster the sounds into groups of sounds that you think belong together. Make as many groups as you think are necessary, but not more. Groups can have only one sound, or many."

The subject is presented with a window containing the icons for all the texture in the set (see Figure 2.5.1). The grouping is done by dragging the icons into groups within the window. The window with the clustered icons is then printed, and the subject is asked to give a qualifier to each group. The experiment took an average of 6 minutes to complete, including the printout.

FIGURE 2.5.1 Initial Window for the Grouping Experiment



We did two sets of grouping experiments, with different subjects and slightly different stimulus sets.

First Grouping Experiment

A first run of the grouping experiment was done to test the usability of the interface. The set contained 10 sounds, which were identified with a random mixture of lower and upper case letters, numbers, and symbols (see Table 2.5.1). There were 5 subjects.

TABLE 2.5.1 Set of Stimulus for the First Grouping Experiment

identifier	name	description
*	bubbles	fish tank bubbles
2	crickets	constant drone of many crickets outside
K	crowd	medium-sized crowd inside, with reverb
%	forest	constant drone with distant birds
6	helicopter	closed-miked constant helicopter rotor
n	lawn mower	closed-miked constant lawn mower engine
+	rain	constant heavy rain
q	traffic	many individual traffic horns
y	transformer	strong 60 Hz buzz
F	whispers	many people whispering, with reverb

Second Grouping Experiment

In the second grouping experiment, the same 12 textures of the similarity experiment are used (see Table 2.4.1). The grouping experiment usually immediately follows the similarity experiment, so that the subject has had plenty of opportunity to get familiar with the sounds. There were 7 subjects.

2.5.2 Results

The first grouping experiment showed that subjects quickly accepted the user interface. The icon metaphor was grasped immediately: one icon for each sound texture, double-click the icon to play, drag the icon to organize the groups.

This first experiment also provided a number of groupings, which were in essence similar to those in the second grouping experiment; the results are thus combined.

Groupings provided by the subjects are shown on Figures 2.5.2 through 2.5.5. A list of the grouping names and contents is given on Table 2.5.2.

FIGURE 2.5.2 Result of Grouping Experiment for Subject 1

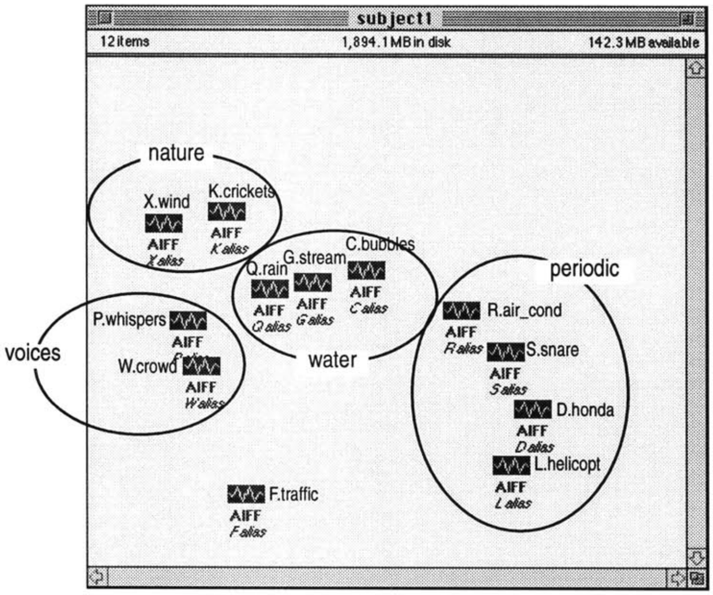


FIGURE 2.5.3 Result of Grouping Experiment for Subject 2

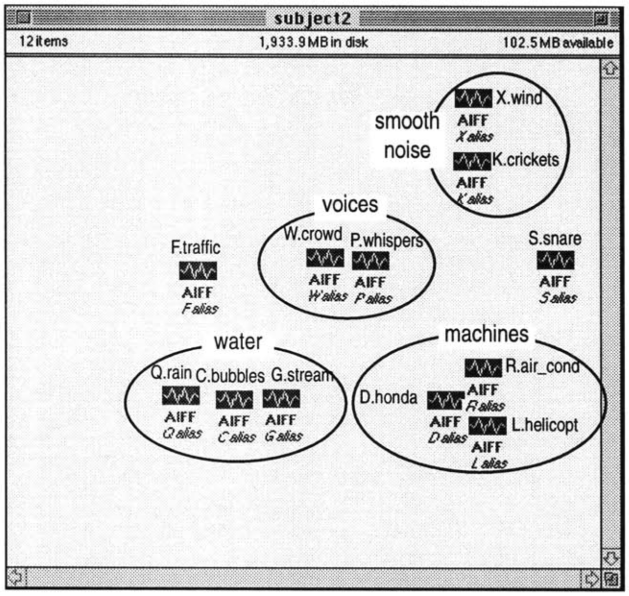


FIGURE 2.5.4 Result of Grouping Experiment for Subject 4

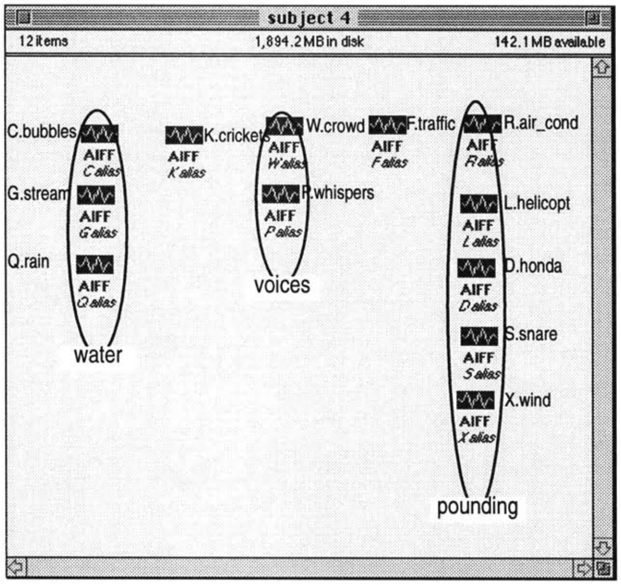
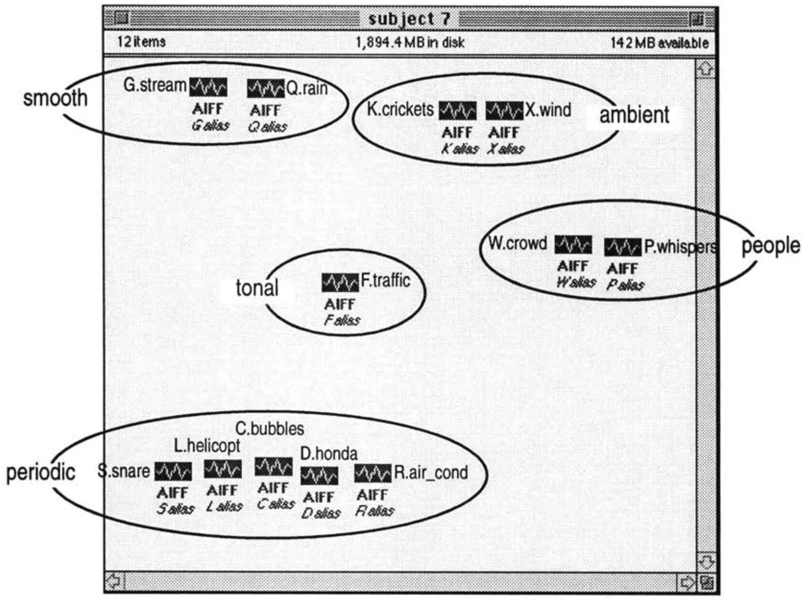


FIGURE 2.5.5 Result of Grouping Experiment for Subject 4



The kinds of grouping were quite varied, but some themes occurred more often. It is obvious that the groupings and their names are influenced by the selection of stimuli, but there is no way to avoid that.

TABLE 2.5.2 Compilation of the Groupings of Sound Textures Done by the Subjects

Subject(s)	Group name	Group
1,2,3,4,6	water^4, watery	bubbles, rain, stream
1,2,5,6,7	nature^2 constant, hum smooth, smooth noise "ambient"	crickets, wind
1,2,4,6	people^2, voices^2, speechy, spacious	crowd, whisper
2,3,5	machines^3	air_cond, helicopt, honda
1,5	periodic^3, mechanical	air_cond, helicopt, honda, snare
4	pounding	air_cond, helicopt, honda, snare, wind
7	periodic, hard, close	air_cond, bubbles, helicopt, honda, snare
5,7	constant background smooth, random	rain, stream
3	low volume variance	crickets, snare, whisper, wind
3	high volume variance	crowd, traffic
5	stochastic	whispers
5	irregular foreground	bubbles, crowd, traffic
7	tonal	traffic

2.6 Ordering Experiment

Goal

The third experiment seeks to find how sound textures are related to each other. The Similarity Experiment measured the perceived similarities between the sounds to set the *distances* on a map. On the other hand, the Ordering Experiment tries to find the *connections* between the textures, like the routes on a subway map.

2.6.1 Method: Trajectory Mapping (TM)

The Ordering Experiment is based on the Trajectory Mapping technique [RIC93]. The subject is presented with all possible pairs of stimuli, and asked to select a feature that changes between the two and then find an interpolator and two extrapolators according to the chosen feature. The collected data is split into triplets of stimuli which share a gradation of some perceived characteristic. The most common triplets are then used to built a two-dimensional map with links between stimuli (see Figure 2.6.4).

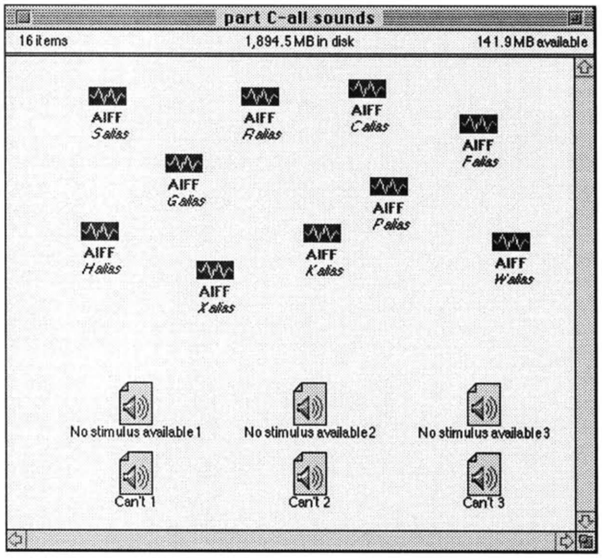
TM has been used to analyze musical interval [GIL94], and to order visual textures [RIC93].

Protocol

Once again, a small set of textures is chosen for stimuli. The subject is presented sequentially with each possible pair of textures in the set, and asked to choose an interpolator and two extrapolators, in as much as possible, for each pair. The instruction read as such:

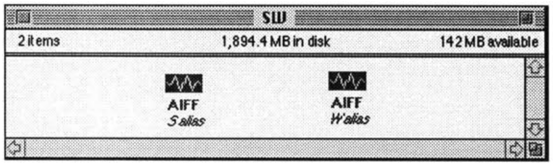
- During this experiment, you will work with a **set of 10 sound textures**. Each sound is represented by an icon, and identified by a capital letter, as in the top half of the **Set Window** shown on Figure 2.6.1.

FIGURE 2.6.1 Ordering Experiment Set Window



- **Listen** to all the sounds of the set, by double-clicking on their respective icon. Playback can be stopped by typing a dot *before you click on anything else.*
- On each trial of this experiment, you will be presented with a **Pair Window** that contains the icons corresponding to two **reference** sound textures, as in Figure 2.6.2.

FIGURE 2.6.2 Ordering Experiment Pair Window



- You will be asked to find one interpolator and two extrapolators for each pair.
- Listen to the two reference sounds, and choose a characteristic which differentiates the two.
- **Interpolator**: from the rest of the set, try to find a sound that fits between the reference sounds, according to the characteristic you chose.
- **Copy** the icon of the interpolator by holding the option key and dragging the icon from the Set Window to the center of the Pair Window.
- **Extrapolators**: in the same fashion, try to find a sound that would go after the two reference sounds. Use the sound that is closest to the second reference sound, according to the characteristic you chose. Copy its icon to the right side of the Pair Window.
- Finally, find a sound that would go before the two references, *as close to the first sound as possible* (another extrapolator). Copy its icon to the left side of the Pair Window.
- It is likely that some pairs are so **mismatched** that you will not find an interpolator. You should then copy a "**can't**" icon to the center of the pair window. You can then skip to the next trial.
- Similarly, You may find that one of the extrapolators can't possibly exist. You should also copy a "**can't**" icon in the appropriate space.
- If you think that there could be a sound that would fit between the reference sounds, but you can't find it in the Set Window, copy a "**No stimulus available**" icon. This also applies to the extrapolators.

The sound textures used are shown on Table 2.6.1. The set contained N=10 sounds, so there are $\frac{N(N-1)}{2} = 45$ pairs. The experiment lasted over an hour. There were 2 subjects.

TABLE 2.6.1 Set of Stimulus for the Ordering Experiment

identifier	name	description
C	bubbles	fish tank bubbles
F	traffic	many individual traffic horns
G	stream	quiet water stream or brook
H	applause	light applause in medium room
K	crickets	constant drone of many crickets outside
P	whispers	many people whispering, with reverb
R	air-cond	air conditioning compressor & fan
S	snare	constant snare drum roll
W	crowd	medium-sized crowd inside, with reverb
X	wind	constant wind outside

2.6.2 Results

A list of triplets of textures is extracted from the interpolator-extrapolator quintuplets. The most common triplets are then used to form the paths between the textures. The automatic procedure to extract the TM paths from the data and the actual processing for our subject data are a courtesy of Stephen Gilbert.

Results of the TM procedure for subjects 1 and 2 are shown on Figures 2.6.3 and 2.6.4. The simplest form of output for TM is a map of the connections between items without considerations for perceptual distance. This is shown on the left side of the figures. To integrate the concepts of distances between items as well as connection, it is possible to overlay the paths on a similarity map, like those produced by the MDS algorithm. The right side of the figures show such overlays, with the paths for subject 1 on a 3-D MDS map and the paths for subject 2 on a simpler 2-D map.

Figure 2.6.3 (subject 1) shows three paths originating from crowd, a moderately random texture. One path connects to progressively more periodic sounds: applause, snare and A/C. Another important path leads to smoother, less random textures: whispers, wins and crickets. A short path goes to traffic, a random texture with a lot of contrast. A minor path from wind through stream to snare, from quiet and smooth through more random and less quiet, to almost periodic end even less quiet. Another minor path connects whispers to streams to applause, all three random and progressively less quiet.

Subject 1 shows three main branches: periodic, smooth and (high-contrast) random, starting from a group of random and moderately quiet textures. The grouping experiments for the same subject (Figure 2.5.2) shows a similar periodic group, a "nature" group (corresponding to smooth), a voice group (present in the central random and moderately quiet group). The water group of the grouping experiment has disappeared in the TM, possibly because rain was removed from the latter experiment.

FIGURE 2.6.3 Trajectory Mappings for Subject 1

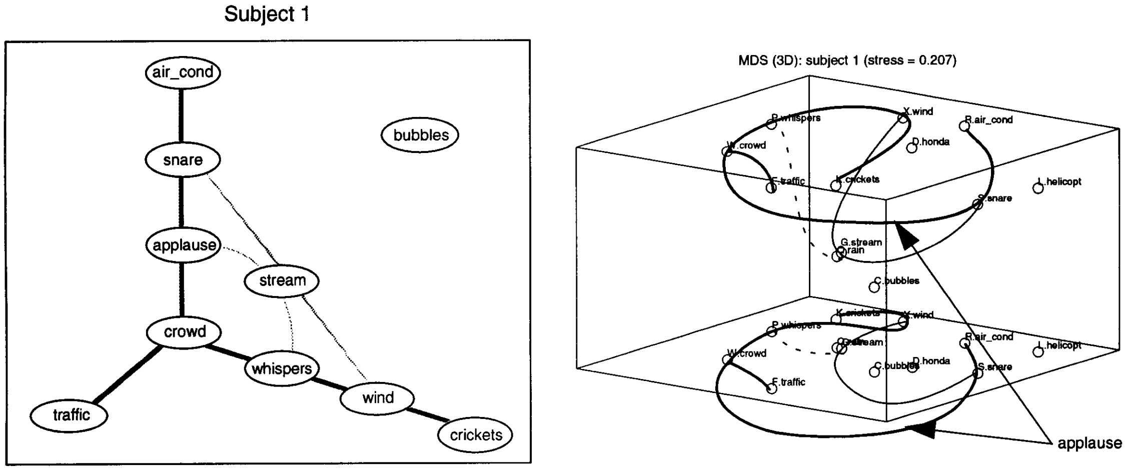
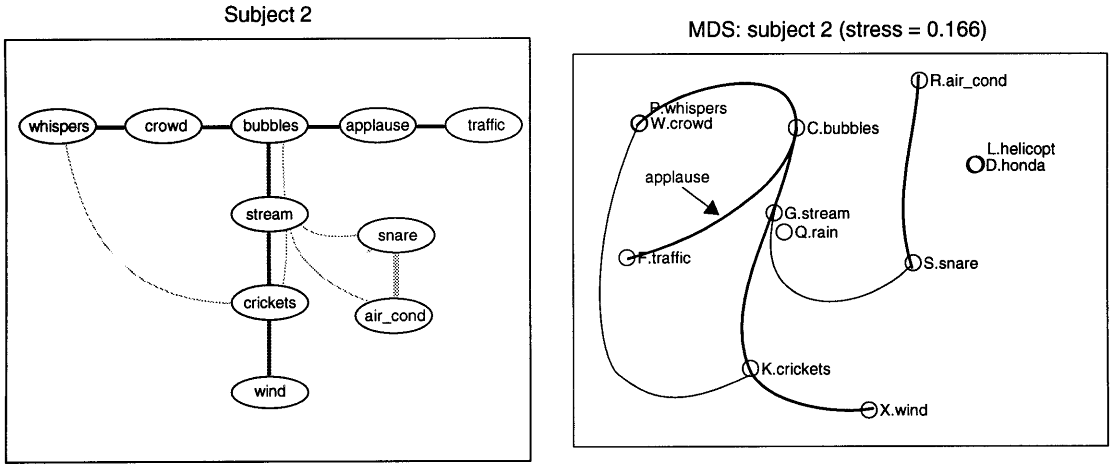


Figure 2.6.3 (subject 2) shows two main paths forming a T. The top bar (whispers, crowd, bubbles, applause, traffic) is a progression from quiet to busy of random and almost periodic sounds. The stem of the T (bubbles, stream, crickets, wind) is also increasingly quiet, ending with smooth sounds. Another strong path links the strongly periodic sounds (snare and A/C). Minor paths link whispers to crickets, and the periodic group to bubbles.

Comparing to the grouping results for subject 2 (Figure 2.5.3), we see that the base of the T is the smooth noise group, the left branch is the voices group and the node is almost the water group. Changes in the sets (removal of rain, helicopter and honda, addition of applause) changed the machines group to a more general periodic group.

FIGURE 2.6.4 Trajectory Mappings for Subject 2


Subject 2

MDS: subject 2 (stress = 0.166)

2.7 P-classes

One goal of the experiments was to find a small set of classes derived from actual groupings done by people listening to sound textures. They should also have names that are strong cues of the contents of the classes; the names should in fact enable the subjects to reproduce the groupings. The concept of a group of stimuli formed by subjects, with a representative name is called P-class.

The most common groups from the grouping experiment are an starting point choice for our P-classes. Note that this restricts the possible P-classes to subsets of the set of textures used in the experiments, but this is fine in the scope of this thesis, where a limited corpus of textures is acceptable, if not desirable. The chosen P-classes are shown on Table 2.5.2.

TABLE 2.7.1 Selected P-classes

	P-class name	Group
by kind of source	water	bubbles, rain, stream
	voices	crowd, whisper
	machines	air_cond, helicopt, honda
by sound characteristics	periodic	air_cond, bubbles, helicopt, honda, snare
	random	crowd, rain, stream, traffic, whispers
	smooth noise	crickets, wind

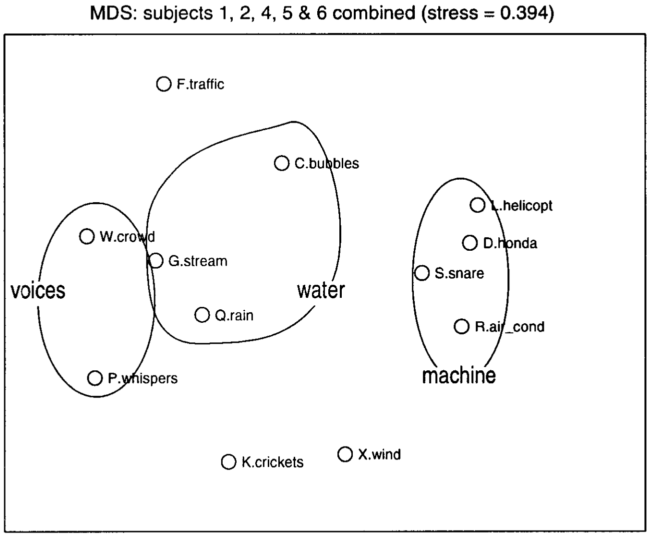
There are two kinds of groupings: by kind of source and by sound characteristics. We kept three P-classes by kind of source:

water and voice, both of which have agreement by the majority of subjects, and machines, which was almost as popular. The P-classes by sound characteristics are less clear-cut: periodic is a superset of machines, with the addition of bubbles and snare; smooth noise is the chosen name for the group of textures with a very low volume variance[1]; and random textures have more volume variance and a random arrival of features.

In the experiment, the subjects were asked to split the textures into clusters, which did not allow overlapping classes. This limitation may have had an influence on membership of the classes.

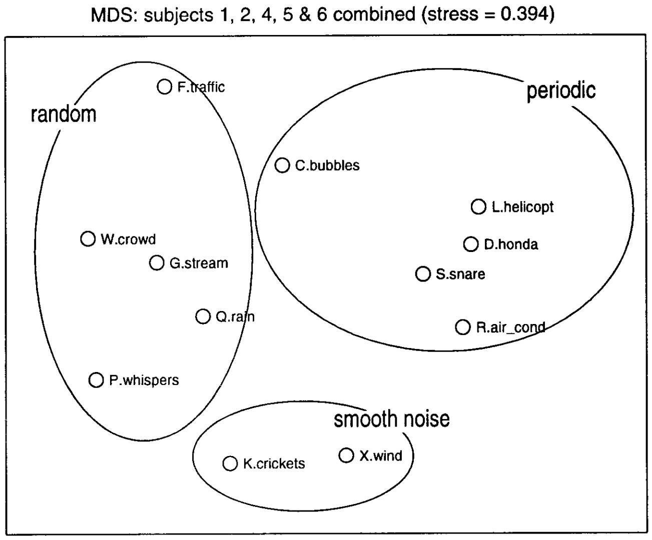
The combined MDS map (Figure 2.4.3) is of great help for visualizing the P-classes. Figure 2.7.1 show the groupings by kind of source, and Figure 2.7.2 show the groupings by sound characteristics.

FIGURE 2.7.1 P-classes by Kind of Source



[1] The "crickets" texture is the sound of hundreds of crickets in the distance; it has a low volume variance and it is not strongly periodic like a single cricket.

FIGURE 2.7.2 P-classes by Sound Characteristics



Chapter 3 Machine Classification of Sound Textures

Chapter Overview The first section of this chapter introduces a two-level approach for the analysis of sound textures. Section 3.2 introduces sound atoms, at the low level of the model. Section 3.3 describes atom extraction by a constant-Q filterbank. Section 3.4 introduces the high level of the approach: modeling transitions of atoms. The last section shortly describes a cluster-based probability model used to encode transitions of atoms.

3.1 A Two-Level Texture Model

Sounds can be broken down into many levels, from a very fine (local in time) to a broad view, passing through many groupings suggested by physical, physiological and semantic properties of sound. We choose, however, to work with only two levels: a low level of simple atomic elements distributed in time, and a high level describing the distribution in time of the atomic elements.

For many sound textures – applause, rain, fish-tank bubbles – the sound atom concept has physical grounds. Many more textures can also be usefully modeled as being made up of atoms. Without assuming that all sounds are built from atoms, we use the two-level representation as a model for the class of sound textures that we work on.

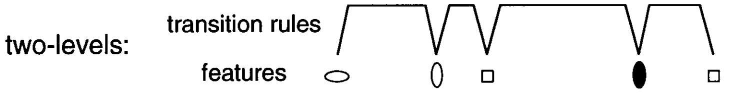
The boundary between low and high level is not universal: we could use very primitive atomic elements, or more complex atoms. Note that using simpler atoms leaves the high level to deal with more information and more complexity. On the other hand, one should be careful not to make too narrow assumptions – loosing generality – when choosing more complex atomic elements.

Such a two-level representation has some physical grounding, as explored in "Auditory Perception of Breaking and Bouncing Events" [WAR84]. In this paper, Warren and Verbrugge present a "structural" level characterized by the properties of the objects being hit, and a "transformational" level, characterized by the pattern of successive hits in breaking and bouncing events.

Key Assumption

Within the scope of this thesis, we assume that Sound Textures are made of sound atoms, and those atoms are subject to a distribution in time. A sound texture is defined by both the nature of its atoms, and their transition rules. We call this a two-level approach (Figure 3.1.1).

FIGURE 3.1.1 Transitions of Features

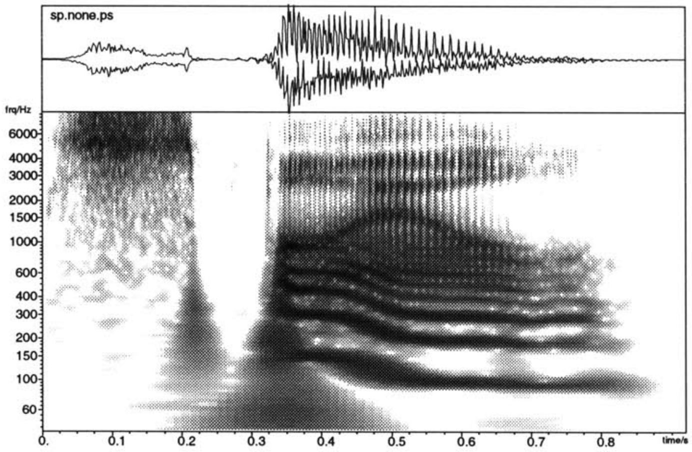


The atoms can be simple or complex, can be homogenous (all the same type) or different, can have one parameter (e.g. amplitude) or many. The transition rules can obey a simple period or a complex stochastic distribution. A model describing a texture must have sufficient information about the atoms, and also about the transition rules, but not more than necessary.

3.2 Low Level: Sound Atoms

The basic representation of sound is a time waveform of the sound pressure level in the air. In the human ear, the cochlea performs a time-frequency transform. Figure 3.2.1 shows the time waveform for an occurrence of the word "spoil", and an example of time-frequency transform (a constant-Q transform) underneath. A time-frequency representation is often called spectrogram.

FIGURE 3.2.1 Example of a Time-Frequency Representation: a Constant-Q Transform[1]



Sound atoms should form patterns in the proper representation domain. For example, the time-frequency representation of the phonemes of the word "spoil" (Figure 3.2.1) form horizontal stripes which could be used as atoms. Atom can be made complex, but we can also choose simpler atoms, such as groupings of energy contiguous in time and frequency.

3.2.1 Kinds of Atoms

Feature extraction from sound is a very rich field, and choosing features for sound (sound atoms) is no exception. A number of different features have been proposed for sound, ranging from the trivial (PCM samples) to the very complex (e.g. simulation of the filtering done by the cochlea followed by grouping of harmonics and noise bursts [ELL94]). Feature extraction is a very important step. In our case, the probabilistic model can sometimes make up for the lack of refinement of the features, but with the expense of more computational load.

Signal-Specific Atoms

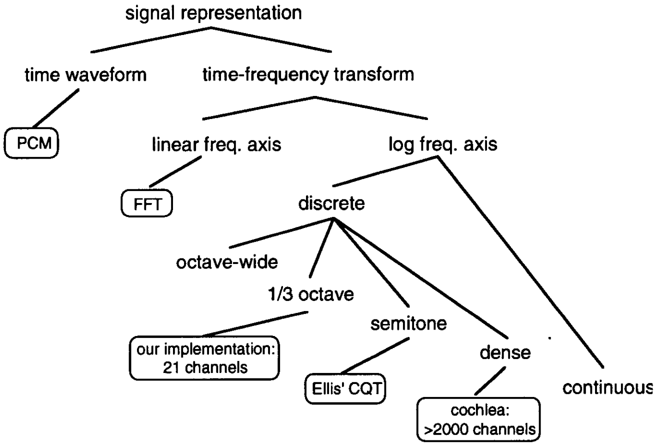
Some sound textures have obvious high-level sound atoms: claps in applause, drops in rain, bubbles in fishtank, rumble and clicks in copy machine, etc. This would seems ideal for our analysis, but there is a difficulty: which atoms to look for? Claps or bubbles are not general enough to be used as atoms for all textures.

[1] Courtesy of Dan Ellis

Generalized Atoms

In choosing the features, some characteristics are desirable. Features have to keep relative time location information, have limited computation time, and help reduce the complexity of the model. In some cases, it is desirable that feature extraction be reversible for resynthesis. It also makes sense to have features that mimic the ear's time-frequency sensitivity. Figure 3.2.2 shows a few possible sound representations.

FIGURE 3.2.2 A Variety of Sound Representations



The simplest feature we can choose is the amplitude of the waveform (PCM sample, [RED76]). We have shown that these minimal features can produce acceptable results for trivial signals [SAI94]; but a lot of clusters are wasted at the feature level, instead of being more intelligently used at a higher level.

To find general atoms for all sounds, it is helpful to look at the processing in the human hear. The sound pressure is captured by the eardrum and passed to the cochlea, where a time-frequency transform is done. It seems a good idea to do a frequency transform.

A typical feature space for sound is a time-frequency diagram, where time increases to the right, and frequency increases from the bottom (Figure 3.2.1). Such a representation can be produced by a variety of means, including filterbanks [VAI93] and transforms. Any sound can be exactly represented by the values of amplitude and phase at every point on this time-frequency diagram, although humans do not perceive such detail.

Filterbanks

The filterbank can be as simple as a few octave-wide filters, or a score of third-octave filters (as we will use), or a few dozen semitone-wide filters [ELL92], or thousands of frequency-tuned sensors, as in the cochlea. In [ELL92], Ellis uses the constant-Q transform; his implementation has filters spaced by one half-tone (log frequency axis). This seems to give a reasonable approximation of the filtering done by the human cochlea.

A straightforward atom is energy localized in time and frequency. We could for example use a filter bank to split the incoming sound in a number of frequency bands. Our atomic "features" are the energy level in each band. In a previous test [SAI94], we implemented a critically sampled, 6-band tree-structured filterbank for analysis and resynthesis of sound textures. And now we use a 21-bands filterbank.

Energy Groupings in Time and Frequency

On a spectrogram, the energy is not uniformly distributed, but tends to cluster. Instead of keeping all points on the time-frequency diagram, we can simplify the representation by grouping energy and keeping only the high-level parameters of the groupings. This is the subject of Section 5.1, "Better Atoms," on page 78.

3.3 A 21-band Constant-Q Filterbank

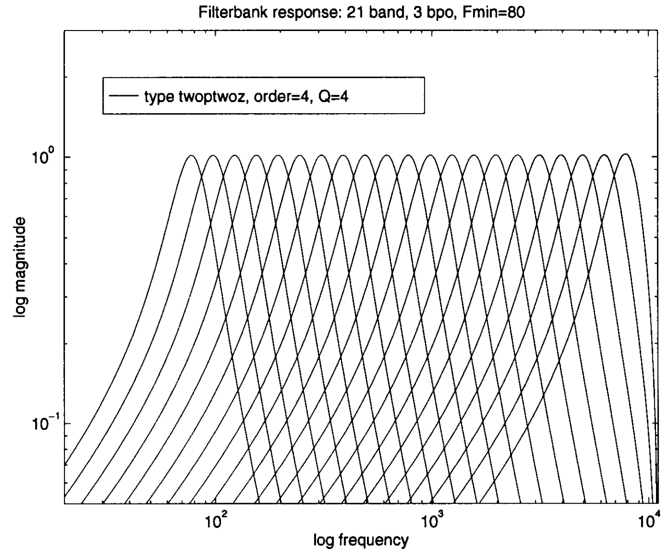
In the current classification system, we chose to use a third-octave filterbank, because it provides an adequate frequency resolution without the major computational burden of full cochlea models. Third-octave implies the log-frequency spacing of the filters.

21 frequency bands were required cover the range from 80 Hz to 8 kHz. The upper limit of 8 kHz is dictated by the Nyquist frequency of 11 kHz and the width of the filter. The lower limit of 80 Hz chosen because little energy was observed below that frequency in the available sound texture samples – while to reach the perceptual limit of 20 Hz would have required and additional six filters.

The filters have a selectivity factor Q=4, which is also a compromise between the need for selectivity (narrow frequency bands) and the time-lag introduced by the long filters required for selectivity. Still, this time-lag became important in the low frequencies, and had to be corrected to align on the spectrogram the energy which is simultaneous in time.

The frequency response for the 21 filters is shown on Figure 3.3.1. Note that the filters appear equally spaced because of the log-frequency scale.

FIGURE 3.3.1 Filterbank Response



Hilbert Transform

Each channel of the filterbank lets only components within a narrow frequency band go through. The resulting time-frequency map for an applause signal can be seen on Figure 3.3.2. The top part of the figure shows the long-time spectrum, where we can see the occurrences of the hand claps, their intensity and their frequency content. The lower part shown a highly magnified part of the same signal. The frequency decomposition is obvious: the period of the signal in each individual channel gets smaller as the frequency increases. Although this is a complete representation (and reversible) representation of the sound, the constant variations of the signal makes it harder to see the magnitude of the signal in each band.

The Hilbert transform is a way to extract the envelope of a narrow-band signal. It produces an out-of-phase version of the input signal (Figure 3.3.3), like a sine from a cosine. Combining the original signal and Hilbert transform into a complex signal and taking the magnitude leads to the energy envelope (Figure 3.3.4), which is much easier to read.

FIGURE 3.3.2 Original Signal

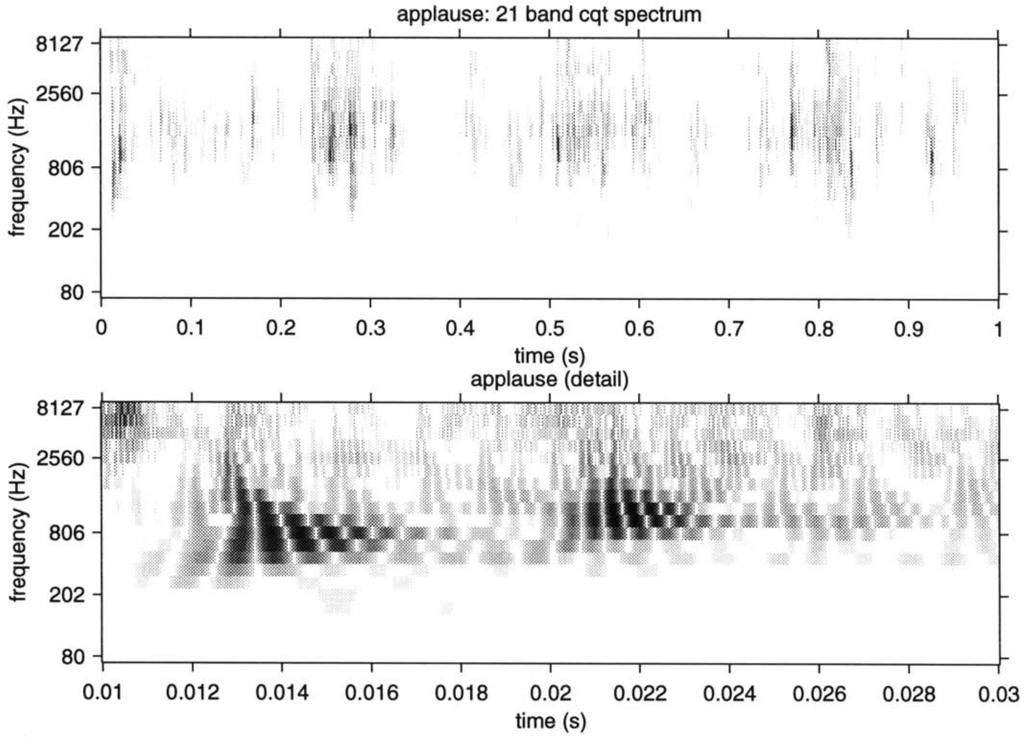
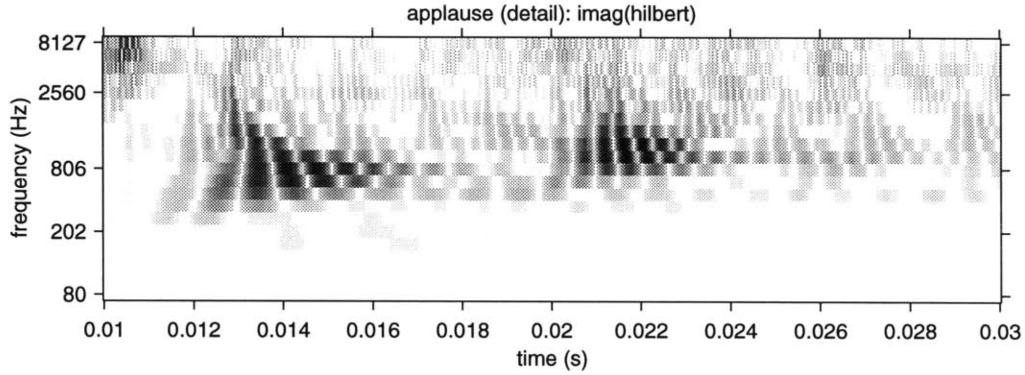


FIGURE 3.3.3 Imaginary Part of the Hilbert Transform



The envelope is much smoother than the original signal which should make it a domain where it is easier to track energy transition. It contains most of the information of the original signal, but not all of it: the phase (Figure 3.3.5) is left out[1]. In our system for classification, the atoms are the samples of the energy envelope.

[1] Using atoms that ignore the phase may impair resynthesis.

FIGURE 3.3.4 Magnitude of the Hilbert Transform (Energy Envelope)

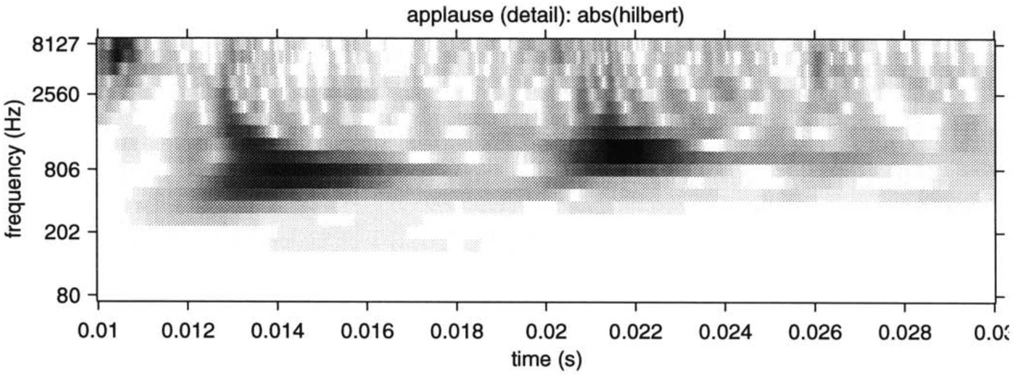
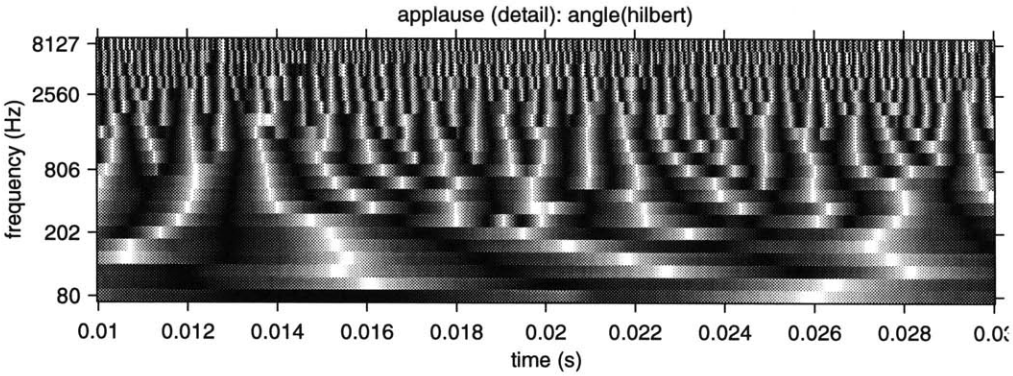


FIGURE 3.3.5 Phase of the Hilbert Transform [1]



3.4 High Level: Distribution of Sound Atoms

Periodic and Random

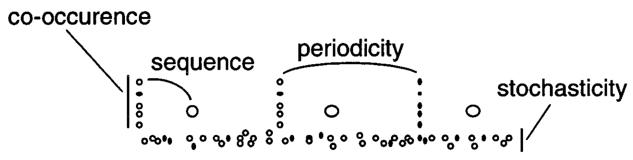
The high level of our two-level sound representation is concerned with the distribution of the sound atoms extracted at the low level. We identify periodic and stochastic (random) distributions of atoms, as well as co-occurrence and sequences of atoms. These different ways of distributing atoms are not exclusive of each other; they can be mixed and even combined in a hierarchy.

A sound in which similar atoms occur at regular interval in time is said to have a periodic distribution. Textures such as engine sounds have a periodic distribution. In a stochastic distribution, atoms occur at random times but obey some arrival rate. Rain and applause are examples of textures with a stochastic distribution of

[1] Notice how the phase aligns on the peaks of the magnitude: this could be used to create an artificial phase signal for resynthesis.

atoms. Different atoms that occur at the same time are said to co-occur. The impact of objects makes a sound where atoms co-occur in different frequency bands. Atoms also often occur in predictable sequences.

FIGURE 3.4.1 Example of Distributions of Atoms: the Copier



As an example, our photocopier makes a sucking sound in which many frequency components have high energy (co-occurrence). The sucking sound is followed (sequence) by a feeder sound. The suck-feed sequence is repeated sequentially (periodicity). At all times there is a low rumble (stochasticity). Figure 3.4.1 is a stylized representation of those four kinds of distributions.

Occurrences of atoms can be grouped into a hierarchy, e.g. the sucking sound (a co-occurrence) is periodic. The high level model should address all four kinds of distributions, as well as hierarchic distributions of distributions.

Cluster-Based Probability Model

The method we use for characterizing the distribution of atoms (the cluster-based probability model) does not assume that the texture takes a particular distribution. Instead, it tries to characterize the distribution of atoms by keeping statistics on the most likely transitions.

Modeling feature transitions (Analysis)

In the proposed work, the analysis of the texture will be done on the output of the feature extractor using a cluster-based probability model. We use the model to represent the transitions between the current state of the texture and the next output value. The cluster-based model approximates a complex, multi-dimensional probability mass function (PMF) by placing a limited number of clusters in the more dense areas of the PMF. If we assume that the clusters are separable along each dimension and that their shape is Gaussian, then for each cluster we need to know only the centroid (N-dimensional mean value) and variance.

We assume the nature, frequency, distribution and/or sequence of the features are picked up by the auditory system and that they determine the perception of the sound texture. Furthermore, we assume that the cluster-based probability model can represent enough significant transitions of the features to encode the behavior of the sound. We know that assumption is valid for deterministic transitions of sounds, although the process might require a lot of clusters.

3.5 Cluster-Based Probability Model

3.5.1 Overview

To characterize the high level transitions of sound atoms, we use the cluster-based probability model introduced by Popat and Picard [POP93]. This model summarizes a high dimensionality probability density function (PDF) by describing a set of clusters that approximate the PDF. We will use a notation similar to theirs. Popat and Picard have used the cluster-based probability model for visual textures and image processing with success.

The cluster-based probability model encodes the most likely transitions of ordered features. Features (in our case sound atoms) are put in vectors, and the order of features within the vector is determined by a neighborhood mask. The features used to encode the transitions are taken in the neighborhood of the current feature in the feature space, hence the name neighborhood mask. The vectors formed from a training signal are clustered to summarize the most likely transitions of features.

The input to the analyzer is a series of vectors in N-dimensional space representing the features of the training data. The vectors are clustered using a K-means algorithm [THE89], slightly modified to iteratively split its clusters. The centroid of each cluster, its variances and relative weight then form a lower-dimensionality estimate of the statistics of the training vectors.

The next section gives an example of the cluster-based analysis method, using a simplified feature space: the 1-D time series of PCM values of the amplitude of a waveform.

Cluster-Based Analysis Example in 1 Dimension

Let's take a sine wave as training signal, and the PCM values $y[t]$ of the intensity of the waveform as a feature. The training signal y is a time series:

$$y = \{y[0], y[0], \dots, y[t_0], \dots\} \qquad \textbf{(EQ 3.5.1)}$$

At observation time t_0, $y[t_0]$ is the PCM value of the current sample. Let's make a transition vector $\vec{x}[t_0]$ with the value of the current sample, the value of the previous sample and the value from 10 sample ago. The vectors have a dimension $d = 3$:

$$\vec{x}[t_0] = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = pick(y, \vec{h}, t_0) = \begin{bmatrix} y[t_0 - 10] \\ y[t_0 - 1] \\ y[t_0] \end{bmatrix} \qquad \text{(EQ 3.5.2)}$$

where $\vec{h}$ is the neighborhood mask:

$$\vec{h} = \begin{bmatrix} -10 \\ -1 \\ 0 \end{bmatrix} \qquad \text{(EQ 3.5.3)}$$

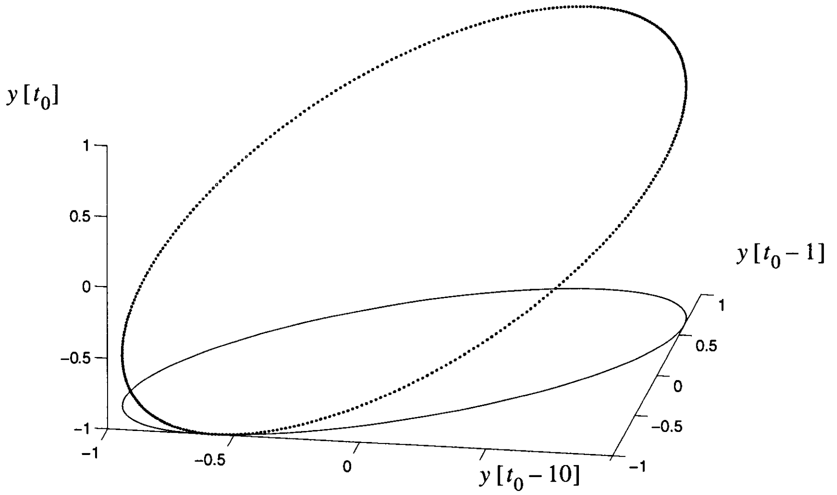
Think of a ruler with one hole for each position in the mask. As you slide the ruler along the signal, you can read a sequence of training vectors $\{\vec{x}\}$ through the holes in the ruler:

$$\{\vec{x}\} = \{\vec{x}[10], \vec{x}[11], \ldots, \vec{x}[t_0], \ldots\} = \left\{ \begin{bmatrix} y[0] \\ y[9] \\ y[10] \end{bmatrix}, \begin{bmatrix} y[1] \\ y[10] \\ y[11] \end{bmatrix}, \ldots, \begin{bmatrix} y[t_0 - 10] \\ y[t_0 - 1] \\ y[t_0] \end{bmatrix}, \ldots \right\} \qquad \text{(EQ 3.5.4)}$$

The vector $\vec{x}[t_0]$ contains a few values of the past of y which led to the current value $y[t_0]$, in a sense transitions of y which led to $y[t_0]$, hence the name transition vector.

Since the training signal is a sine wave, the current value $y[t_0]$ is completely determined by two values of the past. The transition vectors $\vec{x}[t]$ are a discrete representation of that fact. If we have a sufficient number of transition vectors, plotting $\{\vec{x}\}$ as points in 3-D space shows an single oval path: for any pair $(y[t_0 - 10], y[t_0 - 1])$, $y[t_0]$ can take only one value. This is illustrated on Figure 3.5.1, where the points are the transition vectors in the 3-D space, and the circle path underneath is the "shadow" of the points on the $(y[t_0 - 10], y[t_0 - 1])$ plane.

FIGURE 3.5.1 Transitions for a Sine Wave



Approximation by clusters

The sine wave signal is convenient because it is completely determined by two previous values. Furthermore, we know that it is a sine wave. But what if we did not know that? We could still make an estimate of the behavior of the training signal $y[t]$ by observing the transition vectors. To characterize an unknown training signal, we need to generalize the transitions exemplified in $\{\vec{x}\}$.

Let's suppose that the training vectors obey some rules. This is simple for the sine wave signal, which is completely deterministic: the transition vectors fall on an oval in 3-D space. In other words, the training vectors are samples of that oval.

Similarly, we assume that the sequence of training vectors from any signal is samples of a probability density function (PDF) in the transition vector space. For the sine wave, the PDF is zero everywhere outside an oval.

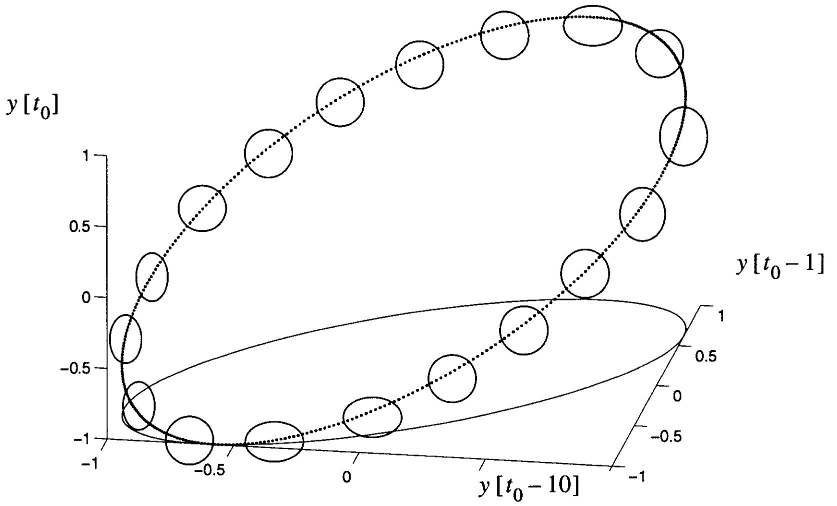
Since the PDF is usually much more complex than an oval, we need a general way to summarize the PDF. Actually, we need a way to estimate the PDF $f(\vec{x})$ from the PDF samples $\vec{x}$.

The cluster-based probability model uses clustering to form such an estimate of the PDF by clustering vectors in the dense parts of the vector space. The PDF estimate $\tilde{f}(\vec{x})$ is the sum of M clusters K_m:

$$f(\vec{x}) \cong \tilde{f}(\vec{x}) = \sum_{m-1}^{M} K_m \qquad \text{(EQ 3.5.5)}$$

In our example with the sine wave, we would see clusters all around the oval, like a pearl necklace. The estimate is granular because of the clusters, so we need a sufficient number M of clusters to properly estimate the PDF, and subsequently we need a sufficient number of transition vectors to cluster.

FIGURE 3.5.2 Cluster Approximation of the Transitions of a Sine Wave



A more detailed look at the cluster-based probability model follows.

3.5.2 Description

This section describes the cluster-based probability model. For a more detailed description and related work, the reader should read the papers by Popat and Picard referenced in the Bibliography.

We distinguish two basic cases: if we have continuous samples, then the vectors of features obey a probability density function (PDF); if the samples take only discrete values, we try to estimate a probability mass function (PMF).

Let $\vec{x} = [x_1,\ldots,x_N]'$ be a random vector in a N-dimensional space, and let $\vec{X}$ and X_n be instances of $\vec{x}$ and x_n respectively. Then $\vec{x}$ is the general form for a vector of features, and $\vec{X}$ is an instance vector of features.

If we build feature vectors from the training signal, we get a learning sample $\Im$ (also called training data):

$$\Im = \{\vec{x}_s\}, s = 1 \ldots S \qquad \text{(EQ 3.5.6)}$$

We assume that the N elements of $\vec{x}$ can each take any value in a range $\chi \subset \Re$; the range can be continuous or discrete. The space spanned by vector $\vec{x}$ is $\chi^N \subset \Re^N$. In the continuous case, χ is a bounded interval of the real numbers, and in the discrete case, χ is a set of K equally spaced values, usually integer. Digitized sound is quantized, thus discrete. For example, if the quantization is done to 8 bits, then there are $K = 2^8 = 256$ possible values for x_n, usually noted $\chi = \{0 \ldots 255\}$. A discrete vector of $N = 3$ dimensions could then take $K^N = 2^{8 \cdot 3} = 2^{24}$ distinct values.

We assume that successive instances of $\vec{x}$ are independent and that they follow the same probability law. This constrains signals to have stable statistics in time to be successfully analyzed. In the continuous case, $\vec{x}$ is governed by the probability density function (PDF) $f(x)$, with

$$\int_V f(\vec{X})\, d\vec{X} = Prob\{\vec{x} \in V\} \qquad \text{(EQ 3.5.7)}$$

for all measurable $V \subset \chi^N$. It is assumed that $f(\vec{x})$ is continuous and bounded. In the discrete case, $\vec{x}$ obeys a probability mass function (PMF) $p(\vec{x})$ defined as

$$p\vec{X} = Prob\{\vec{x} = \vec{X}\}, \vec{X} \in \chi^N \qquad \text{(EQ 3.5.8)}$$

By convention, x_N is the feature currently under observation, which is conditioned by the other features of the vector (see Figure 3.5.3). We call the other features $[x_1, \ldots, x_{N-1}]'$ the "conditioning neighborhood" of the feature under observation, because we assume that the current feature depends in part on the neighborhood:

$$p(\text{current feature} \,|\, \text{neighborhood}) = p(x_N | x_1, \ldots, x_{N-1}) \qquad \text{(EQ 3.5.9)}$$

If we call the feature under observation the "current" feature, then an observation vector $\vec{x}$ is the "current state" of the system.

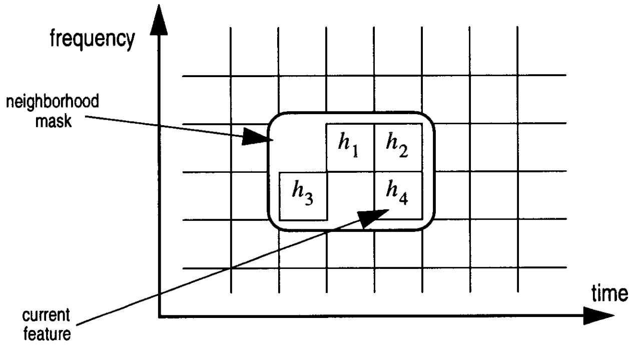
Two-dimensional Feature space

Picking a feature vector from a one-dimensional space is described in the previous example. In general, neighborhood masks will span as many dimensions as there are in the feature space. If the feature space is 2-dimensional, as for images or sound spectra, the notation is only slightly modified. As we can see in Table 3.5.1.

TABLE 3.5.1 Notation for a Time-Frequency Feature Space

training signal	$y = \{y(t, f)\}$
neighborhood mask	$\vec{h} = \begin{bmatrix} \Delta t_1, \Delta f_1 \\ \dots \\ \Delta t_{N-1}, \Delta f_{N-1} \\ 0, 0 \end{bmatrix}$
vector construction	$\vec{x}[t_0, f_0] = \text{pick}(y, \vec{h}, (t_0, f_0)) = \begin{bmatrix} y(t_0 - \Delta t_1, f_0 - \Delta f_1) \\ \dots \\ y(t_0 - \Delta t_N, f_0 - \Delta f_N) \\ y(t_0, f_0) \end{bmatrix}$

FIGURE 3.5.3 Example of a 2-D Conditioning Neighborhood



$$N = 4$$

$$\vec{h} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} = \begin{bmatrix} 1, 1 \\ 0, 1 \\ 2, 0 \\ 0, 0 \end{bmatrix}$$

Note that in the two examples, the neighborhood masks are causal, i.e. all the features in the neighborhood were taken from previous values of the time series. Neighborhood masks do not need to be causal, although causality is helpful in many real-world applications.

Conditional Probabilities

We equate $f(\vec{x}) = f(x_1, \ldots, x_N)$ and $p(\vec{x}) = p(x_1, \ldots, x_N)$, and write the one-dimensional[1] probability laws for the current feature conditioned on the neighborhood

$$f(x_N | X_1, \ldots, X_{N-1}) = \frac{f(X_1, \ldots, X_{N-1}, x_N)}{f(X_1, \ldots, X_{N-1})} \qquad \text{(EQ 3.5.10)}$$

and

$$p(x_N | X_1, \ldots, X_{N-1}) = \frac{p(X_1, \ldots, X_{N-1}, x_N)}{p(X_1, \ldots, X_{N-1})} \qquad \text{(EQ 3.5.11)}$$

Approximation by clusters

Estimates of $f(\vec{x})$ and $p(\vec{x})$ based on a learning set $\mathfrak{S}$ are noted $\tilde{f}(\vec{x})$ and $\tilde{p}(\vec{x})$. To get an estimate of the PDF[2], We approximate it by a finite number M of clusters:

$$f(\vec{x}) \cong \tilde{f}(\vec{x}) = \sum_{m=1}^{M} K_m \qquad \text{(EQ 3.5.12)}$$

where K_m is the *kernel* of the cluster. Kernels could have any shape, but is useful to define them as separable along each dimension, because then we can use the chain rule to estimate probabilities in a fashion suggested in Eq 3.5.10.

Using a single amplitude term w_m and a separable kernel:

$$K_m = w_m \cdot \prod_{n=1}^{N} k_{m,n} \qquad \text{(EQ 3.5.13)}$$

the PDF estimate becomes:[3]

$$\tilde{f}(\vec{x}) = \sum_{m=1}^{M} w_m \cdot \prod_{n=1}^{N} k_{m,n} \qquad \text{(EQ 3.5.14)}$$

[1] Here, "one-dimensional" is not related to the dimensionality of the feature space. It only means that the probability law applies to one feature, conditioned on all others.

[2] We will use only the continuous case here to spare the reader from the duplication.

[3] In the special case where a) the kernels are the same along each dimensions ($k_{m,n} = k_m$), b) the weights are equal at $w_m = 1/S$, and c) there is one "cluster" for each data point (i.e. $M = S$), we have a Parzen estimate of the PDF [TER89].

Finally, we limit the Kernels to be gaussian, so that we can completely specify each cluster by a scalar weight w_m, a N-dimensional centroid ${}_m\vec{\mu}$, and a N-dimensional variance ${}_m\vec{\sigma}$.

Typically, a grouping algorithm is used to form a set of clusters from a number of training vectors. The weight of each resulting cluster is proportional to the number of vectors in the cluster, the centroid along dimension m is the mean of the vectors along that dimension, and the variance along m is computed from the deviations of the vectors from the centroid along m.

3.5.3 Classification

Model Comparison

We chose to classify textures by comparing the clusters of the model produced for unknown sounds to clusters of models produced for known templates.

Let's consider comparing the model of one unknown texture with one template model. For each cluster in the model of the unknown, we first find the closest cluster in the template model, and compute a distance between the two. The distance between the unknown and the template is the sum of individual cluster distances.

Distance Measures

The ability to compare the models extracted from two textures depends on a distance measure between clusters. Finding distance measures between clusters is not trivial, and there are indeed many valid methods.

In the probability model used, clusters are separable and gaussian, i.e. the influence of one cluster on the PDF is:

$$K_m(\vec{x}) = w_m \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{(x_n-\mu_n)^2}{2\sigma_n^2}} \qquad \text{(EQ 3.5.15)}$$

We define a distance metric between clusters that takes into account the variance:

$$D(K_A, K_B) = \sqrt{\sum_n \frac{({}_A\mu_n - {}_B\mu_n)^2}{{}_A\sigma_n \cdot {}_B\sigma_n}} \qquad \text{(EQ 3.5.16)}$$

Dissimilarity Measure

We decided to compare models by taking the square root of the sum of the square of the distances of their clusters:

$$\Delta(T_t, T_u) = \sqrt{\sum_{m=1}^{M} D^2({}_uK_m, {}_tK_j)} \qquad j = \min_q \{D({}_uK_m, {}_tK_q)\} \qquad \text{(EQ 3.5.17)}$$

$\Delta(T_t, T_u)$ is not really a distance measure, because a it is not symmetric:

$$\Delta(T_t, T_u) \neq \Delta(T_u, T_t) \quad \text{(EQ 3.5.18)}$$

so we call $\Delta(T_t, T_u)$ a *dissimilarity* measure.

Closest Known Texture

The simplest classification method is to compare the clusters of the model of the unknown textured with all the models of the training textures, and choose the closest (less "dissimilar"). We call this *Identification.*

Classification into P-classes

A more interesting classification from the perceptual standpoint is to find in which perceptual class an unknown texture belongs. Here is the proposed algorithm:

Classifier training:

- obtain a set of training textures of the same perceptual class
- extract features from textures
- model feature transitions
- make templates with the extracted models

Classification:

- extract features from unknown texture
- model feature transitions
- compare the clusters of the unknown with the clusters of the template
- assign the class of the closest template to unknown

3.5.4 Evaluating Classification success

The first test of classification is to verify that all the instances used for training the textures that form a class get classified in the right class. Then we can take new textures, and compare the results of machine classification with that of human classification.

It can happen that a new texture does not belong to any of the classes known. Such cases should be detected to avoid giving erroneous classification results. For this purpose, we can run all of the instances of all the textures that form a class through the classifier and find the smallest likelihood ratio that should be accepted for this class. Psychoacoustic experiments could also be conducted to assess this idea of a maximum distance.

Chapter 4 Machine Classification Results

Chapter Overview This chapter presents and discusses the results obtained by our machine classifying system. The first section lists the sample sets used, describes the details of the atom extraction, shows spectrograms of the samples, and gives specifics about the transition modeling and dissimilarity measure. Section 4.2 gives dissimilarity results for the sample sets in tabular and graphical form. Section 4.3 presents a discussion of the results.

4.1 Protocol

4.1.1 Set of Signals

Training Set The training set is made of the same 12 sound textures used in the Similarity Experiment and the Grouping Experiment.

The sounds are extracted from the "Speed of Sound – volume I: SFX" CD-ROM[1]. The training samples are 3 seconds of sound taken from the start of the recording, but after any start-up phase, if present.

Testing Set The testing set is built from the same recordings as the training set, but the samples are taken later in the recording. In addition, three new textures are included. The samples are also 3 seconds long.

[1] published by Aware Inc., Cambridge MA. Sound effects by Sound Ideas.

TABLE 4.1.1 Set of Signals Used for Analysis and Classification

	name	ID	description	Speed-of-Sound file name
training and testing	air_cond	R	Air conditioning apparatus	air.01
	bubbles	C	flow of bubbles in water	bubbles.02
	crickets	K	constant drone of crickets	country.01
	crowd	W	several people with reverb	indoor.06
	helicopt	L	idling helicopter rotor	helicopt.03
	honda	D	big idling motorcycle	honda.01
	rain	Q	heavy rain on hard surface	rain.02
	snare	S	snare drum roll	snare.01
	stream	G	flow of brook on rocks	stream.01
	traffic	F	many car horns	traffic.01
	whispers	P	several people whispering	courtrm.01
	wind	X	constant wind outside	wind.01
testing only	airplane	B	idling cessna engine	single.02
	applause	H	light applause	applause.08
	crowd_02	M	several people talking	indoor.02

4.1.2 Extracting Atoms

The sound textures samples are decompressed from the CD-ROM to a 16-bit mono AIFF sound file with a 22kHz sampling rate. At all further points the data is kept in floating-point notation to avoid overflow or further quantization effects.

The samples are then processed by a 21-band constant-Q filterbank, with frequencies ranging from 50 to 8125 Hz, and Q=4 (see Section 3.3 on page 50). A Hilbert transform is then taken of each channel of the spectrum, and the magnitude (envelope) is used to build a spectrogram.

Figures 4.1.1 through 4.1.15 starting on page 66 show the spectrograms of the test samples. The training samples have very similar spectra. The top part of each graph shows the whole 3-second sample, and the bottom part shows a zoom on half a second of the same.

Dark areas on the spectrograms correspond to the zones of high energy in time and frequency. It is easy to see what frequencies are dominant for each sample; for example crickets is almost exclusively high frequencies, while honda is mostly low-frequencies. Periodicities in sound are visible as periodicities in the spectra: helicopt and honda are good examples. The spectrogram for applause shows the separate claps. It is also possible to see that the frequency content for each bubble goes up as time advances.

4.1.3 Spectrograms of the Sound Texture Samples

FIGURE 4.1.1 Spectrogram of "airplane"

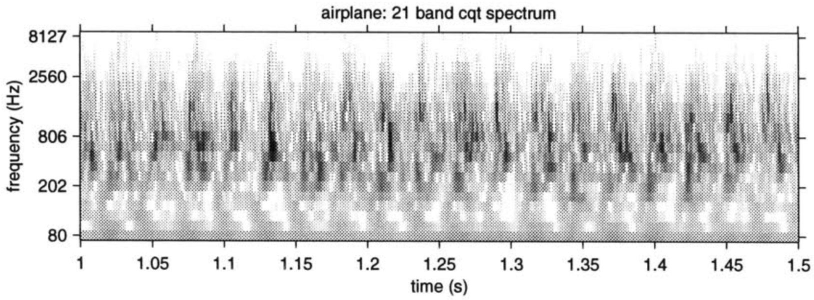


FIGURE 4.1.2 Spectrogram of "air_cond"

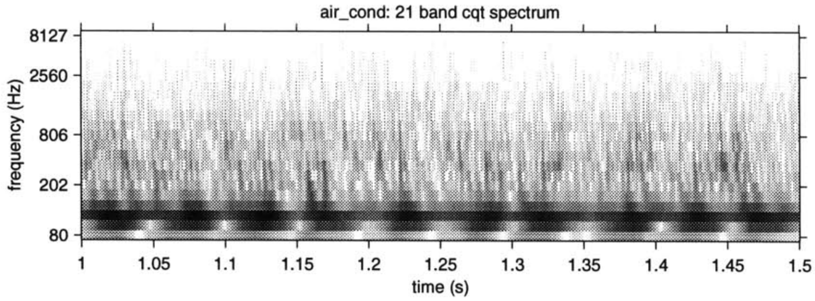


FIGURE 4.1.3 Spectrogram of "applause_08"

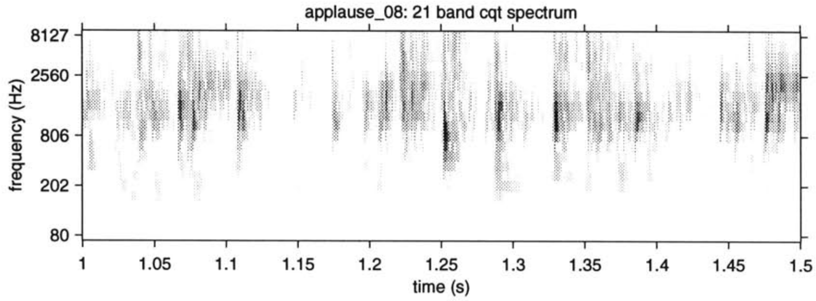


FIGURE 4.1.4 Spectrogram of "bubbles"

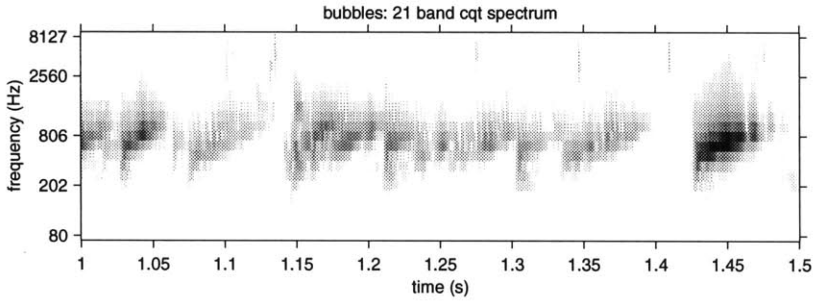


FIGURE 4.1.5 Spectrogram of "crickets"

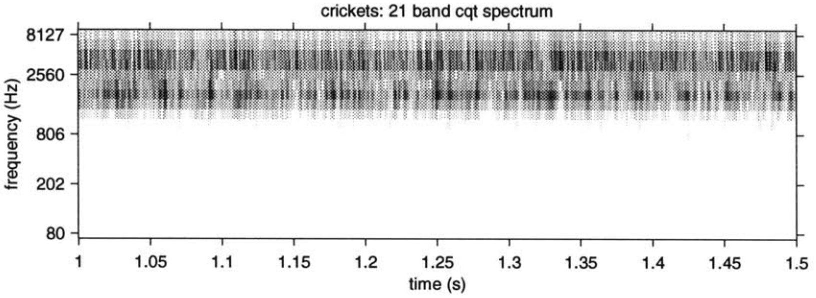


FIGURE 4.1.6 Spectrogram of "crowd_02"

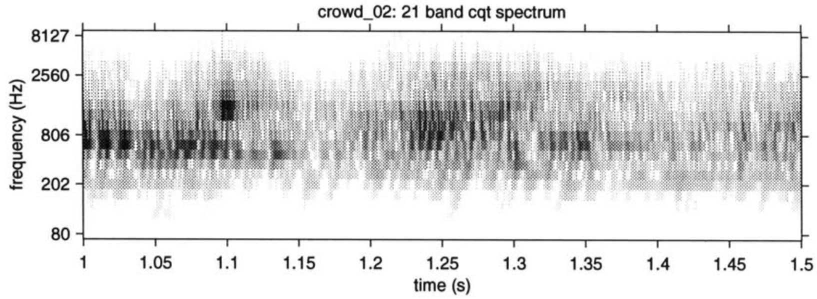


FIGURE 4.1.7 Spectrogram of "crowd_06"

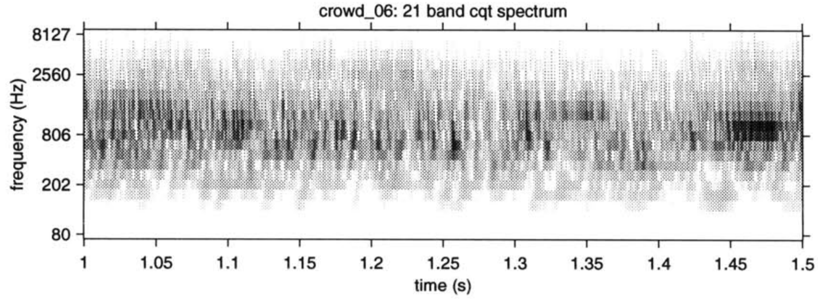


FIGURE 4.1.8 Spectrogram of "helicopt"

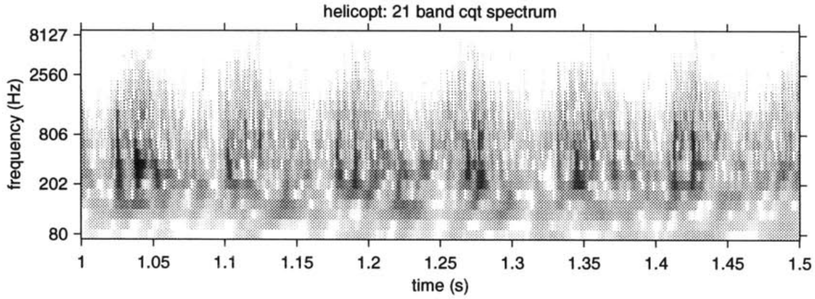


FIGURE 4.1.9 Spectrogram of "honda"

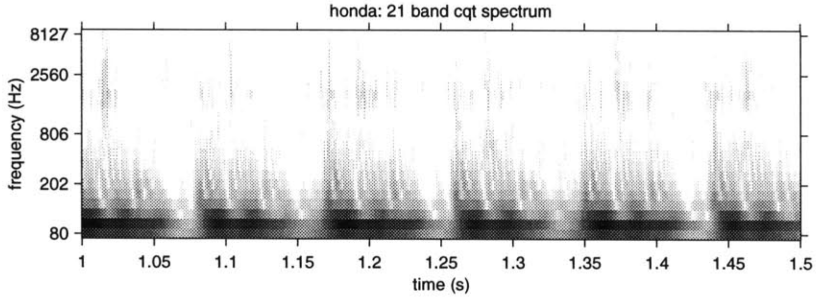


FIGURE 4.1.10 Spectrogram of "rain"

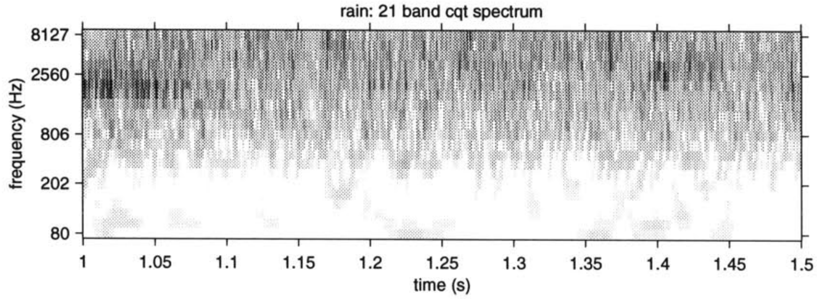


FIGURE 4.1.11 Spectrogram of "snare"

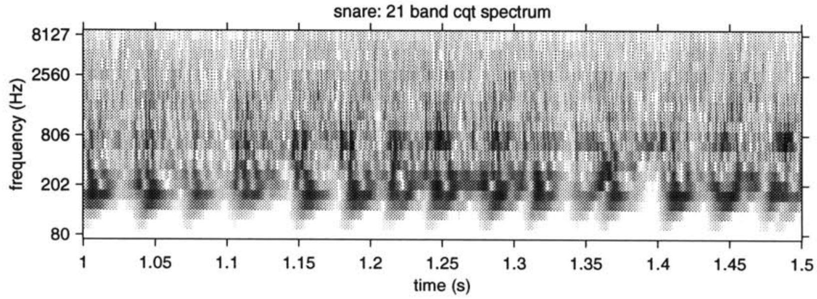


FIGURE 4.1.12 Spectrogram of "stream"

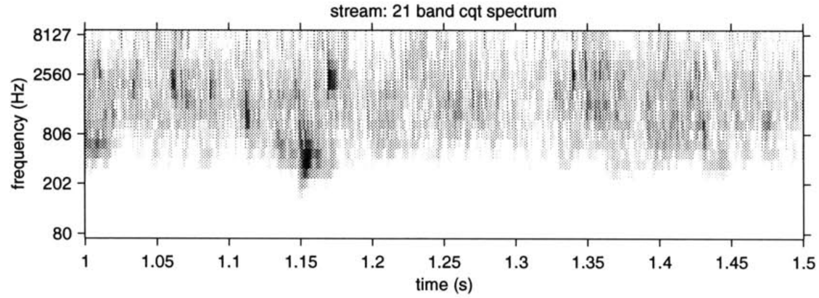


FIGURE 4.1.13 Spectrogram of "traffic"

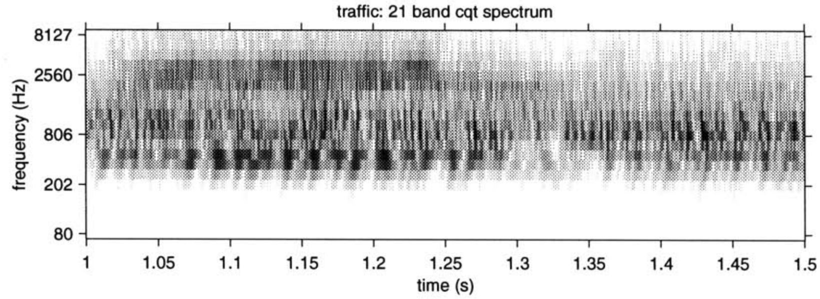


FIGURE 4.1.14 Spectrogram of "whispers"

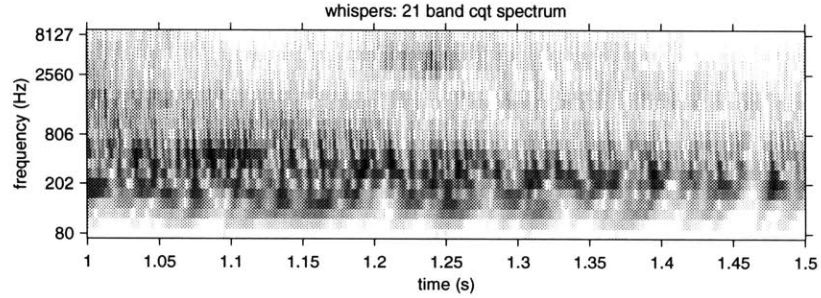
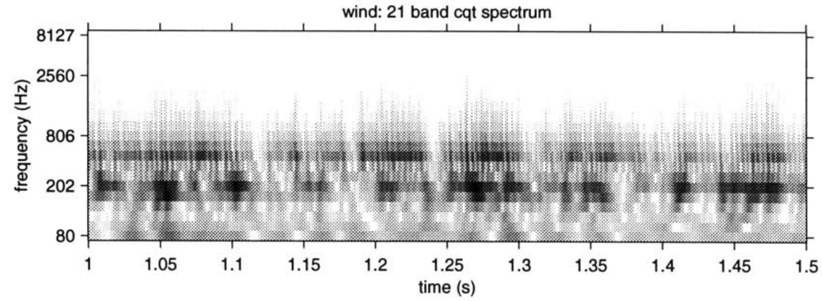


FIGURE 4.1.15 Spectrogram of "wind"



4.1.4 Modeling Transitions

Training vectors are built from the spectrogram. The same neighborhood mask is used for all spectrograms; it spans areas of earlier time and higher frequencies bands. The offsets used in the mask are shown on Table 4.1.2. Positive time differences are in the past, and positive offset in frequency-band number means a higher frequency. The masks are designed to capture both short- and long-term time transitions, as well as co-occurrences of energy in different frequency bands.

TABLE 4.1.2 Time and Frequency Offsets for the Neighborhood Mask

	time delay (samples)	**frequency band offset (number of frequency bands)**
$\hbar(t, f)$ =	(frequency band number of current atom)	
	0	2
	0	5
	8000	0
	4000	1
	2000	2
	1000	3
	500	5
	250	5
	400	0
	50	3
	10	0
	5	1
	current atom	

The neighborhood mask used has 12 positions, and with the frequency band number and the current features, it brings the number of dimensions to $N = 14$. We use three seconds of sound and take only one transition vector out of every seven possible vectors, for a total of 132914 vectors. We varied the number M of clusters from 8 to 1024.

The first feature of each transition vector is not an amplitude value read from the spectrogram, but rather the frequency of the current atom. This deviation from regular transition vector building allows one to encode the frequency into the clusters. This in turn makes it possible to encode energy transitions from different frequency bands in different clusters.

In frequency bands with little energy, transition vectors will have uniformly low values, and will tend to cluster into one huge cluster with very low variance. This means that more clusters become available for the frequency bands where the transitions have more energy. In fact it allows a dynamic allocation of clusters among frequency bands.

The inclusion of the frequency band number in the transition vectors is not without problems. Since the frequency band numbers don't have the same range of values as the energy in the spectrogram, computing distances between vectors and cluster centers in this mixed-type space is like comparing apples and oranges. The solution we bring to this problem is quite simple and proves effective: we scale the frequency band number so that its variance is similar to the variance of the other features in the vector.

4.1.5 Comparing Cluster-based Models

In Section 3.5.3, we define the dissimilarity between an unknown texture to a template texture as the square root of the sum of the square of the distance from each cluster ${}_uK_m$ in the model of the unknown to the closest cluster ${}_tK_j$ in the model for the reference:

$$\Delta(T_t, T_u) = \sqrt{\sum_{m=1}^{M} D^2({}_uK_m, {}_tK_j)} \qquad j = \min_q \{D({}_uK_m, {}_tK_q)\} \text{(EQ 4.1.1)}$$

This means that the dissimilarity measure is not symmetric, i.e.

$$\Delta(T_t, T_u) \neq \Delta(T_u, T_t) \qquad \text{(EQ 4.1.2)}$$

and is therefore not a "distance metric". This will affect the symmetry of tables and should not surprise the reader.

4.2 Results

4.2.1 Dissimilarities within the Training Set

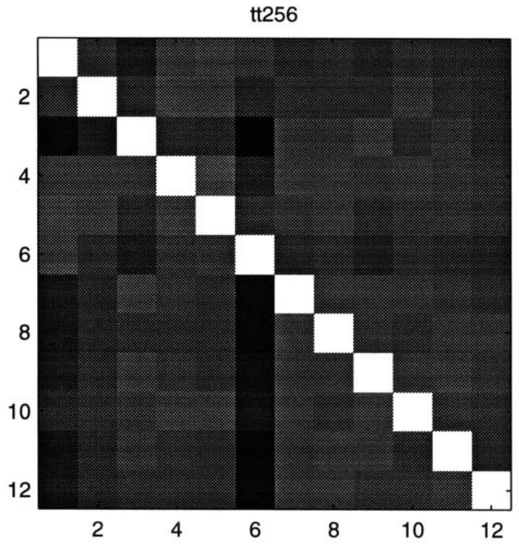
As a first step, we computed the dissimilarity between each texture in the training set. This is intended as a verification of the ability of the classifier to distinguish between different textures and *identify* identical models. The results for 256 clusters are given on Table 4.2.1.

TABLE 4.2.1 Dissimilarities between the Textures of the Training Set with 256 Clusters

	1	2	3	4	5	6	7	8	9	10	11	12	
	A/C	bub	cri	cro	hel	hon	rai	sna	str	tra	whi	win	class
air_cond		226	916	152	120	215	446	291	515	239	386	545	1
bubbles	258		343	86	91	511	244	224	233	128	265	477	2
crickets	2368	890		432	506	5400	152	155	114	294	197	487	3
crowd_06	160	156	187		67	707	131	134	148	76	173	239	4
helicopt	114	132	441	80		263	214	134	278	146	202	277	5
honda	143	342	929	240	193		462	309	585	297	384	648	6
rain	1010	412	110	197	232	2316		117	84	136	147	220	7
snare	1226	479	225	241	277	2793	114		99	169	83	133	8
stream	858	353	93	191	219	1939	73	77		135	107	169	9
traffic	522	262	163	86	120	1423	105	147	123		194	270	10
whisper	1312	544	177	312	332	3004	150	86	125	230		333	11
wind	1237	491	296	247	274	2834	143	91	200	169	133		12

We expect a perfect match (dissimilarity = 0) between identical models, and indeed we get zeros in the diagonal. The other distances vary a lot, and the table is not the best way to visualize which textures are considered similar by the machine. An grey-scale image of the dissimilarity may help a bit (Figure 4.2.1).

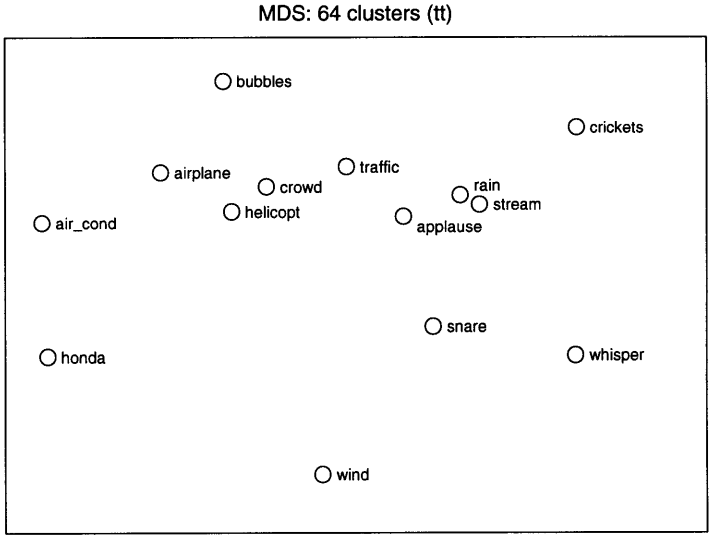
FIGURE 4.2.1 Computed Distances Between the Textures of Training Set



The dissimilarity is the greatest for most comparisons with the honda sample. Looking at its spectrum (Figure 4.1.9) we see that it is both very periodic and mostly low frequency, which makes it quite distinctive from the others. The strongest dissimilarity is between crickets (Figure 4.1.5) and honda, and in fact, both their spectra and their patterns are radically different.

MDS A more graphical way to look at the computed dissimilarities is using MDS to draw a scaled map which approximately translates dissimilarities into distances (Figure 4.2.2)

FIGURE 4.2.2 MDS of Distances Computed with 64 Clusters



4.2.2 Classification to Closest in Training Set

In this section, we compute the dissimilarities between all the textures in the test set and each texture in the training set. Tables 4.2.2 and 4.2.2 shows numerical results using 8 and 64 clusters respectively. By convention, rows refer to the "unknown" textures (T_u), and the columns refer to the training textures (T_t). The "class" column indicates to which training sample the test sample was most similar. Figure 4.2.3 display a grey-scale image of the dissimilarity for 64 and 256 clusters.

TABLE 4.2.2 Dissimilarities Measured from Test Set to Training Set for 8 Clusters

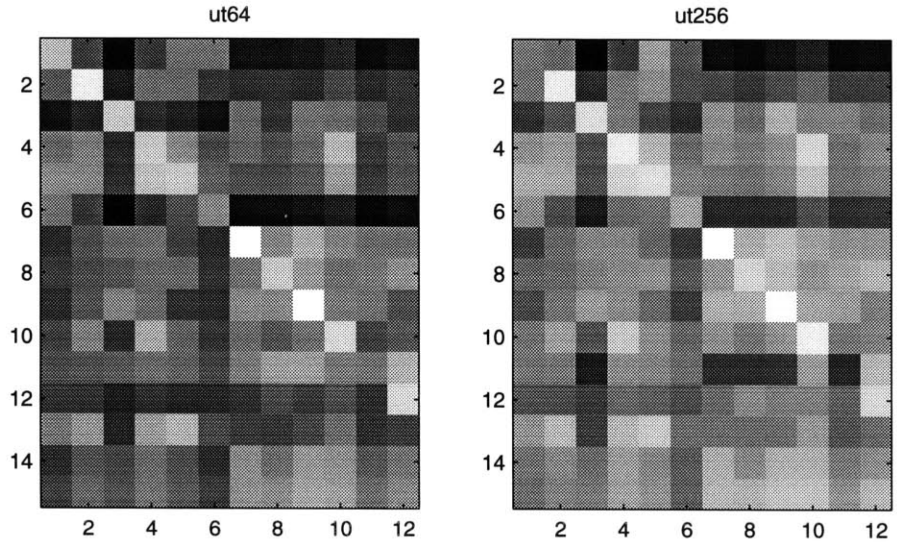
	1	2	3	4	5	6	7	8	9	10	11	12	
	A/C	bub	cri	cro	hel	hon	rai	sna	str	tra	whi	win	class
air_cond	0.3	3.0	10.6	2.2	1.4	1.4	6.5	6.1	6.1	3.0	8.8	5.2	1
bubbles	2.3	0.1	7.1	1.4	1.7	3.0	4.3	4.5	4.0	2.3	6.1	4.2	2
crickets	11.6	4.7	0.1	5.8	9.5	8.6	2.7	6.3	1.6	3.3	4.1	5.8	3
crowd_06	2.5	1.8	4.5	0.7	1.9	2.6	2.6	2.8	2.5	0.8	4.3	2.5	4
helicopt	1.6	1.7	5.4	0.9	0.7	2.0	3.2	3.1	3.0	1.2	4.6	2.7	5
honda	1.6	2.5	8.2	2.4	1.9	0.7	5.4	4.7	4.8	2.5	6.7	3.7	6
rain	5.7	2.7	1.7	2.1	5.0	4.6	0.3	2.0	1.1	1.8	2.3	3.1	7
snare	3.3	2.9	2.2	2.1	2.6	3.1	1.6	0.6	1.1	2.0	2.1	1.3	8
stream	7.4	2.7	1.2	2.6	6.2	5.5	1.8	2.5	0.1	2.1	2.7	3.8	9
traffic	3.5	1.7	4.0	1.1	3.0	3.1	2.4	2.7	2.3	0.4	3.9	2.7	10
whisper	3.4	2.5	2.4	2.2	3.0	2.7	1.9	1.1	1.2	1.9	2.0	0.9	12
wind	4.2	3.6	4.0	3.4	3.4	3.1	3.5	2.7	2.6	2.5	3.1	0.2	12
airplane	2.0	1.1	7.2	1.2	0.7	2.7	4.3	4.5	4.0	2.2	6.0	4.1	5
applause_08	3.6	1.7	2.6	1.6	3.4	2.8	1.4	1.8	1.3	1.3	2.6	2.3	10
crowd_02	4.0	2.4	2.1	2.0	3.5	3.3	1.4	1.6	1.1	1.3	2.5	2.0	9

TABLE 4.2.3 Dissimilarities Measured from Test Set to Training Set for 64 Clusters

	1	2	3	4	5	6	7	8	9	10	11	12	
	A/C	bub	cri	cro	hel	hon	rai	sna	str	tra	whi	win	class
air_cond	0.4	1.9	18.7	2.0	0.8	0.8	8.7	9.3	7.5	4.6	11.7	9.3	1
bubbles	1.4	0.2	5.3	0.9	0.9	1.8	2.6	2.8	2.2	1.5	3.5	2.8	2
crickets	8.6	3.9	0.3	2.1	4.3	8.5	0.9	2.1	0.8	1.0	1.5	2.6	3
crowd_06	1.0	0.7	2.3	0.3	0.6	1.3	1.0	1.2	1.0	0.4	1.8	1.2	4
helicopt	0.7	0.7	2.6	0.4	0.3	1.0	1.3	1.4	1.1	0.5	1.9	1.3	5
honda	0.9	1.9	20.6	2.8	1.2	0.6	9.7	10.3	8.3	5.4	12.8	10.3	6
rain	2.9	1.3	0.8	0.8	1.6	2.6	0.2	0.7	0.4	0.7	0.9	1.0	7
snare	1.7	1.4	1.1	0.9	0.9	1.8	0.8	0.3	0.5	0.9	0.8	0.6	8
stream	2.9	1.2	0.7	0.9	2.0	2.5	0.6	0.7	0.2	0.8	0.9	1.4	9
traffic	1.6	0.7	2.1	0.5	1.0	1.8	1.0	1.3	1.0	0.3	1.7	1.3	10
whisper	1.5	1.2	1.2	0.9	1.0	1.5	0.7	0.5	0.5	0.8	0.8	0.5	12
wind	2.1	2.0	4.0	2.0	2.1	2.2	2.1	1.5	1.6	1.3	2.7	0.3	12
airplane	0.9	0.5	3.4	0.5	0.4	1.4	1.6	1.8	1.5	0.9	2.4	1.8	5
applause_08	2.1	1.3	1.5	0.8	1.3	2.5	0.6	1.0	0.6	0.6	1.2	1.0	9
crowd_02	1.6	1.1	1.2	0.8	1.1	1.6	0.6	0.7	0.5	0.6	1.1	0.7	9

For signals in the testing set that are taken from the same recording as one of the templates, the dissimilarity between the test sample and its correspondent is always the lowest of all comparisons, most of the time by one or two orders of magnitude.

FIGURE 4.2.3 Dissimilarities Measured from Test Set to Training Set (64 and 256 Clusters)



4.2.3 Classification into P-classes

The last step of our machine classification experiments is to classify textures in P-classes identified at the end of chapter 2. P-classes group together the sounds that were most consistently associated by human subjects. Each P-class has a name that should be very indicative for humans. We hope that the machine will do a classification to humans. For convenience, Table 4.2.4 shows the selected P-classes.

TABLE 4.2.4 Selected P-classes

	P-class name	Group
by kind of source	water	bubbles, rain, stream
	voices	crowd, whisper
	machines	air_cond, helicopt, honda
by sound characteristics	periodic	air_cond, bubbles, helicopt, honda, snare
	random	crowd, rain, stream, traffic, whispers
	smooth noise	crickets, wind

To do machine classification into P-classes, we use the training textures as templates for each P-class. An unknown texture is compared to templates, and assigned to the P-class of the closest template.

Table 4.2.5 compares the classification done by humans and that done by our system for each of the testing textures. The models used have 8 clusters, and results are taken from Table 4.2.2.

TABLE 4.2.5 Classification by People (P) and Machine (M)

P-class:	water	voices	machines	periodic	random	smooth noise
airplane			P,M	P,M		
air_cond			P,M	P,M		
applause					P,M	
bubbles	P,M			P,M		
crickets						P.M
crowd_02	M	P			P	
crowd_06		P,M			P,M	
helicopt			P,M	P,M		
honda			P,M	P,M		
rain	P,M				P,M	
snare				P,M		
stream	P,M				P,M	
traffic					P,M	
whispers		P			P	M
wind						P,M

4.3 Discussion

For a wide range of number of clusters (8 to 128), the system is able to identify all except one of the testing samples taken from the same textures as the training samples. The classification always mismatched whispers, and listening to the testing sample for whispers revealed that there is a door slam in the testing sample which introduces a strong artifact.

When we used 256 clusters, air_cond was mismatched, and with 256 or more clusters, the classification system mismatched most samples. Doubling the number of clusters greatly increases computation time, so it is computationally cheaper to use as few clusters as possible. For example, analysis and classification using 8 clusters took about 10 minutes on a DEC Alpha, while classification using 256 clusters took over an hour.

With 8 clusters, most of the time was spent on atom extraction, which is dine in Matlab. Doing the signal filtering in the C language would speed it up. Using specialized hardware and parallel processing could possibly bring the analysis-classification cycle close to realtime.

Classification into P-classes also gives correct results in most cases. Of the 15 testing textures, three were assigned to the wrong P-class, for a success rate of 80%. This does not mean that the system can consistently do perceptual texture classification, but the results are encouraging.

The machine classification is a process which is very different from the human perception system. It should not be expected that the machine will match human perception. However, if we use a lot of textures tagged by humans as templates for the P-classes, and if the classification is robust, the machine could mimic human classification.

In a test we reduced the number of transition vectors used to construct the model for the testing signals (64 clusters, vector subsampling by 11). Although there was less data and presumably less accurate clusters, the dissimilarities remained rather stable. Considering that the modeling time is almost proportional to the number of transition vectors, we can hope to reduce computation time by selecting the lowest possible number of vectors that still achieve good classification. This will however not bring the speed close to real-time, because the analysis is currently a matter of minutes per second treated.

Chapter 5 Future Directions

Chapter Overview

Areas for further research could be focused on analysis, classification or other applications of the system. The analysis method could use a refinement of the atom selection and extraction, more careful transition vector building, and improvements in computational speed. The distance measure used in the classification could probably be improved, and other classification methods could be used; a better analysis would surely improve classification. Other applications of the system include sound texture resynthesis (as was already explored by the author [SAI95]), semantic parameter extraction, manipulation of textures along those parameters, and extrapolation between sound textures.

5.1 Better Atoms

Feature extraction is a key step in many classifying systems, including ours. The choice of sound atoms should be a subject of continued improvement in order to integrate a better model of hearing while reducing the amount of data that the probabilistic model has to work with.

Energy Groupings in Time and Frequency

On a spectrogram, the energy is not uniformly distributed, but tends to cluster. Instead of keeping all points on the time-frequency diagram, we can simplify the representation by grouping energy and keeping only the high-level parameters of the groupings. We should choose the kind of groupings which occur the most frequently.

Musical instruments and speech show tracks on their narrowband spectrograms, i.e. lines of one frequency with a continuity in time. Smith and Serra [SMI87] describe a method to extract tracks from a short-time Fourier transform (STFT) spectrogram. Ellis gets tracks from his constant-Q spectrogram [ELL92].

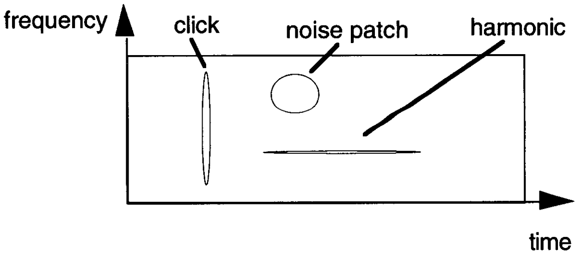
Tracks describe harmonic components, but are not well suited for clicks (broadband, short time) or noise patches (energy spread over both time and frequency). For those, a richer representation is required that allows atoms to span both time and frequency intervals. Mallat's Matching Pursuit [MAL92] is a method that can possibly extract such diverse atoms from sounds.

Harmonics, Narrowband Noise, Impulses

Grouping together energy that is adjacent in time and frequency, and parameterizing these groupings allows a great reduction of the amount of data. Figure 5.1.1 shows three possible groupings of energy adjacent in time and frequency: harmonics, clicks and noise patches. Clicks have simultaneous energy over a range of frequencies, harmonics stay on or around one frequency over a length of time, and noise patches cover an area in both time and frequency. These kinds of groupings are common to most sounds. With proper case (such as some phase conservation), they may be sufficient for a perceptual representation of all sounds.

These three groupings can be united under one "universal" grouping, where the time and frequency distribution are specified (see [MAL92]). However, having multiple parameters (amplitude, frequency or time envelope) increases the dimensionality of the problem.

FIGURE 5.1.1 Clicks, Harmonic Components and Noise Patches on a Spectrogram



5.2 Ideas for Classification

5.2.1 Multiresolution Modeling

The most promising way to improve texture modeling is to use a multiresolution scheme, where the analysis is very general at first and successively refined. A multiresolution scheme was used with success on visual textures [POP95], and there is much hope of similar improvements for sound texture modeling.

Multiresolution means that the modeling is split into a few resolution levels: the high-level characteristics are captured by a first cluster-based model, and successive refinements are encoded in each separate cluster-based model. The transition masks can have fewer points because they don't have to cover both local and global ranges; each level specializes in one range. Models with smaller masks have a lower dimensionality, which greatly reduces the computation time. There is a computational gain even if there are more levels to model, because dimensionality increases the problem size exponentially, while adding levels has a linear impact. A current multiresolution analysis for visual textures improves performance by a factor of five over the previous results [POP94].

In the time-frequency domain for sound, the variable resolution concept applies differently for the time and frequency. In the time domain, analysis can be done for long-time transitions first, and for successively shorter time. On the first analysis, the number of frequency channels can be kept low, to capture the outline of the spectrum. Successive increases in the number of bands can then refine the frequency sensitivity of the analysis. Still, the time and frequency resolution can both increase at each level of the multiresolution analysis.

5.2.2 Model Fit

There is another way to compare an unknown texture to a stored cluster-based probability model: instead of comparing clusters, the machine can form an estimate of the likelihood that a stream of unknown features is properly modeled by the cluster model [POP95].

5.2.3 Increasing the Number of Template Textures

Given that a system has a useful distance measure, increasing the size of the training set is an almost fool-proof way to improve classification accuracy, if the training samples are chosen carefully.

The currents sets of textures are too small to do exhaustive classification tests or significant comparisons with human perception. A more ideal system would have over a hundred textures, each annotated by people with one or more P-class label, so that each P-class would contain many examples (templates). Maybe then could we hope to do a robust classification of a reasonable number of textures. A robust detector of voices, laugh and applause would also be a worthy achievement.

To put the complexity of the task in focus, we should remember that in the course of their lives, humans collect thousands or millions of examples of textures. They can coin a name for any of a wide variety of sounds in a second. The real question is what information is kept in the brain about each example. This careful selection of the salient elements in the sound is what a classification system should aim at reproducing.

5.3 Sound Texture Resynthesis

A necessary condition for resynthesis is that the analysis be reversible. In our two-level texture model, this means that the resynthesis system must be able to reproduce atoms and generate strings of likely transitions.

5.3.1 Reproduction of Atoms

If we used specific atoms for analysis, like single hand claps, we could reproduce the atoms by playing a prototypical sample of the atom.

In our current system, atoms are the values of the energy envelope in frequency bands. The phase of the signal in each frequency band is however not encoded in the model, so it is not available for reproduction. The importance of phase information for accurate playback of sound textures is unclear at the moment, but we suspect that some textures would suffer – in particular those with sharp transients.

The analysis system could be modified to accommodate the phase, either in a separate model or somehow encoded in the same model. The phase can also be estimated ("faked") from a set of empirical rules observed in sound texture samples.

5.3.2 Generation of Likely Atom Transition

The analysis method lends itself naturally to resynthesis. The model is trained on an existing sound texture, encoding the probabilities of transitions between any given state and the next output value. For resynthesis, we first seed the model with an initial state, which can be trivial. Given the current state, the model provides a 1-D PMF of the next output value.

We can then generate a random value with the found distribution (random method), or use a deterministic method [POP93]. In deterministic methods, the new symbol is chosen to explain the current state best, with respect to some Bayesian criterion. Three possible criteria are maximum-likelihood (ML), maximum a posteriori probability (MAP), and least expected square error (LSE). It is not clear which of the methods will give the best result, or even if different methods will result in perceptually different outputs.

The new state of the texture is then computed, including the new value, and the procedure repeated indefinitely. This will produce a sequence of output events with the same transition properties as the training signal. We have already shown that such a resynthesized sequence is also a texture, and that it has some perceptual similarities with the original texture, while staying distinct from it and stochastic in nature [SAI94].

5.4 Modification of Sound Textures

5.4.1 Visualizing the models

Two of the main drawbacks of the probabilistic method used is that its output is large and each piece of data carries little useful information in itself. This makes it difficult to interpret, although more robust to noise. Although it is not essential for modification of sound textures, a means of visualizing the model would help get more insights on the influence of clusters on the sound parameters.

5.4.2 Controls

To modify sound textures one needs a set of control, and the choice of controls is dictated by the parameters of the sound model. Unfortunately, when the available parameters are a useful machine representation, they are not often intuitive. Conversely, intuitive parameters are often very difficult to extract and control by a machine.

A major hindrance to the very concept of control over sound textures by machine is that the current system is far from real-time – it currently takes up to an hour to resynthesize a second of sound. This makes modification of textures extremely tedious.

5.4.3 Semantic Parameters

An ideal system requires perceptually meaningful parameters to allow semantic modification of textures. This is a very hard problem, but certainly worth pursuing.

5.4.4 Texture Interpolation

One possible way to modify textures without a thorough understanding of the parameters is to interpolate between two existing textures. Many ways of interpolating between sound textures come to mind.

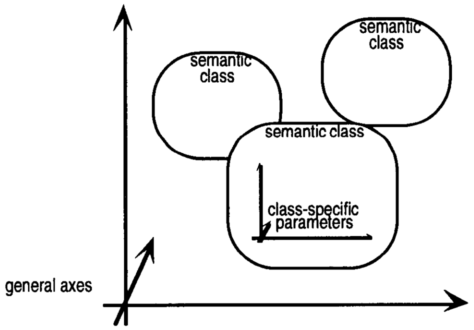
With the current system, we could interpolate the atoms, e.g. apply the frequency of the most likely clusters from one texture to the transition model of a second. We could also interpolate the models, by merging the cluster sets or using alternatingly the model from on texture and another.

In a multiresolution analysis system, it would be easy to swap the model of one or more levels to that of a different texture. This could result in very interesting textures, for example where the high level structure is perceived as belonging to a mechanical device, while the low level is reminiscent of voiced sounds. Interpolation could only take its complete meaning in a system with semantic parameters, which would allow smooth progression from one texture to another along the different semantic axis.

5.4.5 A Model of Models

The exploration described at the beginning of chapter 2 prompted us to propose a model to describe Sound Textures using a few general axes in parallel with semantic classes, completed by class-specific parameters (see Figure 5.4.1). This is a speculative model, given as a reflection and seed for further thought.

FIGURE 5.4.1 A Model of Models



At a first level we have a few general axis which all textures share, such as volume, volume contrast, periodicity, randomness and frequency content. Second we have semantic classes of sound textures, which can span some or all of the general axes; each texture can belong to one or more class. Third there is a set of parameters specific to the class used. Thus we have a model (axes-class-parameters) of models (one specific model for each class).

Machine Estimation of the Model of Models

Given that we have a way of evaluating each level of the model for all textures, the position of each texture on the general axis could be used to compare any two textures, the semantic classes could be used to classify textures, and the class-specific parameters could be used to compare textures of the same class.

Some existing techniques could be used as first approximations to extract some values of the model for an unknown texture, and possibly give usable results. At the first level, general axes, we could use second-order statistics on the intensity to estimate volume and volume contrast. Embedding can estimate the periodicity and randomness of a signal. Frequency content could be reduced to a harmonic/noise ratio (voiced/unvoiced) and a high/low frequency ratio, or a cepstrum coefficient series, although these are possibly over-simplifications of the real problem.

The second level, the definition of the semantic classes and the ability to sort unknown textures among these classes are both very difficult problems, but some of the ideas in this thesis could be used.

As for class-specific parameters, those depend entirely on the models used for each class. Some inverse physical modeling could be used for some classes.

Chapter 6 Conclusion

The work described in this thesis has two main points of focus: human perception and machine classification of sound textures. Although we do not claim to have an artificial system which simulates human perception for this class of sounds, there are parallels that can be drawn between the human and machine aspects.

First, the human perception system is complex and only partially understood. In all human perceptual systems, there is a huge reduction of the amount of data from the environment to the percept; most of the rules and mechanisms for this reduction are still mysterious and might be difficult to formalize if ever discovered. What we do know is that perceptual systems work approximately but robustly in the kinds of environments that humans are likely to face in nature.

At the other extreme, machines need precise sets of non-ambiguous instructions to function, and often lack the robustness to deal with a wide range of input data. They can, however, produce precise and consistent measurements on their preferred class of input signals.

As an example, we could compare the human eye with a spectrometer. The eye is remarkably adaptable in its ability to perceive images over a wide range of lighting conditions (robustness), but at the same time the perception of color is limited to three channels (data reduction). The spectrometer is more sensitive to the dynamic range of its input, but produces a measurement of many more than three channels, and in a more consistent way.

In our sound textures experiments, it is obvious that the human perceptual system throws away a lot of the information in the sound, but we don't know what information is thrown away. Comparison of sound textures by humans is done using very few dimensions, and the mapping from the formal description of the sound (e.g. waveform) to a much simpler percept is very complex.

Our machine classification also uses a mapping from the sound to a simpler representation. This mapping is formal, in the sense that it is described in the preceding pages, and it is very different from the perceptual mapping. In fact, the machine description keeps many more dimensions than the human perception.

The machine analysis presented works too well compared to the human system. It can say "sound A is exactly the same as sound B". Because of that, it will also say "sound C is not the same as sound A, with a difference of x units", and can give the same result consistently.

The question to ponder is then how precise do we really need comparisons of sound textures to be, and what criteria should we use for those comparisons. In the absence of other points of reference, human perception of sound textures, with all its limitations, is a good example of a practical and robust classification system.

References and Bibliography

[BRE90] Bregman, Albert S. Auditory Scene Analysis. MIT Press, 1990.

[CAS94] Casey, Michael A. "Understanding Musical Sound with Forward Models and Physical Models". ,Connection Science, Vol.6, Nos.2&3, 1994.

[ELL92] Ellis, Daniel P. W. "A perceptual representation of audio", SM thesis, MIT EECS dept., 1992.

[ELL94] Ellis, Daniel P. W., "A computer implementation of psychoacoustic grouping rules", Proc. 12th Intl. Conf. on Pattern Recognition, Jerusalem, October 1994
[grouping harmonics and other low-level sound cues]

[GAR94] William G. Gardner, personal communication, Oct. 1994.

[GAV94] William W. Gaver, "Using and creating Auditory Icons", Auditory Display, ed. Gregory Kramer, SFI Studies in the Sciences of Complexity, Proc. Vol. XVIII, Addison-Wesley, 1994.

[GIL94] Gilbert, Stephen A., and Whitman Richards, "Using Trajectory Mapping to Analyze Musical Intervals". Proceedings of the 16th Annual Conference of the Cognitive Science Society, 1994

[HAW93] Michael J. Hawley, Structure out of Sound. Ph. D. Thesis, MIT Media Lab, Sept. 1993.

[JOH82] Richard A Johnson & Dean W. Wichern, Applied Multivariate Statistical Analysis, Chapter 12: Clustering, Prentice Hall, 1982

[KRU77] Joseph B. Kruskal, Forrest W. Young & Judith B. Seery. How to Use Kyst. Instruction manual. Bell Laboratories, Murray Hill, N.J.. 1977.

[MAL93] Mallat, Stěphane, and Zhifeng Zhang, "Matching Pursuit with Time-Frequency Dictionaries", Courant Institute of Mathematical Sciences Technical Report #619, New York, Oct.1992, revised June 1993.

[MET94] Métois, Éric, "Determinism and Stochasticity for Signal Modeling", unpublished, MIT Media Lab, 1994.

[MOO92] Moore, B., An Introduction to the Psychology of Hearing, A.P. London, 1992, p.21.

[NAT94] NATO Advanced Research Workshop on Comparative Time Series Analysis, "Time series prediction : forecasting the future and understanding the past", proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis, held in Santa Fe, New Mexico, May 14-17, 1992 / editors Andreas S. Weigend, Neil A. Gershenfeld; Publication Info: Reading, MA : Addison-Wesley Pub. Co., c1994.

[POP93] Popat, Kris, and Rosalind W. Picard, “Novel cluster-based probability model for texture synthesis, classification, and compression”, MIT Media Laboratory Perceptual Computing Group Technical Report #234, 1993.

[POP94] Popat, Kris, and Rosalind W. Picard, personal communication, Oct. 1994.

[POP95] Popat, Kris, and Rosalind W. Picard,Cluster-Based Probability Model and its Application to Image and Texture Processing", Submitted to IEEE Transactions on Image Processing, 1995"

[RIC79] Richards, Whitman, “Quantifying Sensory Channels: Generalizing Colorimetry to Orientation and Texture, Touch and Tones”, Sensory Processes, Vol.3, pp.207-229, 1979.

[RIC88] Richards, Whitman, editor, “Natural Computation”, MIT Press 1988.

[RIC93] Richards, Whitman, and Jan J. Koenderink, Trajectory Mapping (TM): A New Non-Metric Scaling Technique, Occasional Paper #48, Center for Cognitive Science, MIT, 1993.

[ROS94] Rosenthal, David F., Daniel P.W. Ellis, and Alan Ruttenburg, Pushing Up Hypotheses Using Context-dependent Links”, Unpublished communication, 1994.

[SAI94] Saint-Arnaud, Nicolas. “Sound Textures Resynthesis”, unpublished class report, MIT, May 1994.

[SAI95] Nicolas Saint-Arnaud, “Sound Textures Resynthesis”, Proceedings of the Artificial Joint Conference on Artifician Intelligence (IJCAI-95), August 1995.

[SCH83] Schumacher, Woodhouse, McIntyre, “On the Oscillations of Musical Instruments”, JASA 74 (5), 1983.

[SCO83] D. W. Scott and J. R. Thompson, “Probability density estimation in higher dimensions”, in J. E. Gentel, editor, Computer Science and Statistics: Proceedings of the 15th Symposium on the Interface, pp. 173-179, Amsterdam, North-Holland, 1983.

[SIL86] B. W. Silverman, “Density Estimation for Statistics and Data Analysis”, Chapman & Hall, London, 1986.

[SMI83] Smith, Julius O., Techniques for Digital Filter Design and System Identification with Applications to the Violin. CCRMA Teport No.STAN-M-14, Stanford University, June 1983.

[SMI86] Smith, Julius O., “Efficient Simulation of Reed-Bore and Bow String Mechanisms”. Proceedings of the 1986 ICMC. Computer Music Association, 1986.

[SMI87] Smith, Julius O. and Xavier Serra (CCRMA), “Parshl: an Analysis/ Resynthesis Program for Non-Harmonic Sounds Based on a sinusoidal Representation”, Proceedings of the 1987 ICMC, Computer Music Association, 1987. p.290

[STA95] Stautner, John, personnal communication, January 1995.

[THE89] Therrien, Charles W, Decision, Estimation and Classification, Wiley, 1989.

[VAI93] P. P. Vaidyanathan, “Multirate Systems and Filter Banks”, Prentice Hall, 1993. Ch. 5.

[WAR84] William H. Warren Jr. and Robert R. Verbrugge, “Auditory Perception of Breaking and Bouncing Events: Psychophysics”, in [RIC88], reprinted from Journal of Experimental Psychology: Human Perception & Performance, 10:704-712, 1984.