

**Deriving Musical Control Features from a Real-Time
Timbre Analysis of the Clarinet**

by

Eran Baruch Egozy

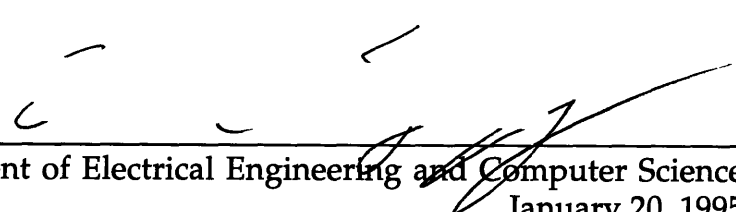
Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements of the Degrees of

Bachelor of Science in Electrical Science and Engineering
and Master of Engineering in Electrical Engineering and Computer Science

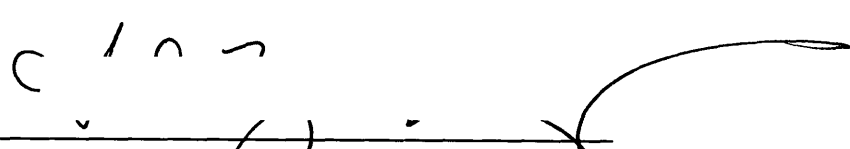
at the Massachusetts Institute of Technology
January 1995

© Massachusetts Institute of Technology, 1995. All rights reserved

Author


Department of Electrical Engineering and Computer Science
January 20, 1995

Certified by


Tod Machover, M.M.
Associate Professor of Music and Media, MIT Media Laboratory
Thesis Supervisor

Accepted by

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY


F. R. Morgenthaler
Chairman, Department Committee on Undergraduate Theses

AUG 10 1995

LIBRARIES **Barker Eng**

Deriving Musical Control Features from a Real-Time Timbre Analysis of the Clarinet

by

Eran Baruch Egozy

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements of the Degrees of

Bachelor of Science in Electrical Science and Engineering
and Master of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology
January 1995

ABSTRACT

A novel system that analyzes timbral information of a live clarinet performance for expressive musical gesture is presented. This work is motivated by the growing need to accurately describe the subtle gestures and nuances of single-line acoustic instrument performances. MIDI has successfully satisfied this need for percussion-type instruments, such as the piano. However, a method for extracting gesture of other instruments, and a MIDI-like protocol by which computers may communicate about these gestures, has been lacking. The exploration of timbre attempts to fulfill the goal of deriving such a parameter stream from the clarinet. Timbre is a particularly evasive attribute of instrumental sound, yet one that is intimately tied to musical expression.

This system uses Principle Component Analysis, a supervised pattern recognition technique, to extract musical gesture parameters based upon the timbre of an acoustic signal. These parametric descriptors can then be used as a control mechanism for interactive music systems, and as a means of deriving high level musical expression.

Thesis Supervisor: Tod Machover

Title: Associate Professor of Music and Media, MIT Media Laboratory

This work is funded in part by the Yamaha Corporation and Sega of America.

Acknowledgments

Of all the people at the Lab who helped shape this thesis, I owe the most gratitude to Eric Métois. His generous advice, pertinent ideas, and technical expertise were invaluable.

Damon Horowitz, Eric, and Dan Ellis were exceptionally helpful by providing comments and corrections after sifting through my drafts. Dan was particularly encouraging at a time when I was losing focus and needed a boost. Michael Casey gave me the first glimpse of pattern recognition and helped direct my initial ideas.

The Media Lab experience would not have been fulfilling without the friendship and talent of my “partners in media”: Ed Hammond, Eric “EMJ” Jordan, Teresa Marrin, Fumi Matsumoto, Suzanne McDermott, Eric Métois, Joe Paradiso, Pete “Phat” Rice, Josh Smith, David Waxman, Mike Wu, and John Underkoffler.

In particular, my officemates Alex Rigopulos and Damon Horowitz have been an endless source of comedy, provocative ideas, support, and puerile behavior.

I owe thanks to Michael Hawley for first showing me the lab when I was a UROP and exposing me to the world of media and music.

Thanks to Aba and Ima for giving me the opportunities to grow and for being an enormous source of support. Offer, my *favorite* brother, deserves special thanks for being a great friend and for putting up with me for so long.

This thesis would not have been possible without the generous musical training I have received throughout the past twelve years. Thanks to my clarinet teachers Louise Goni, Jonathan Cohler, and Bill Wrzecień.

Years of playing chamber music with Elaine Chew, Wilson Hsieh, Julia Ogrydziak, and Donald Yeung, among the many others, has taught me how to become a better musician and has even made MIT bearable.

Finally, I’d like to warmly thank my advisor Tod Machover for his extraordinary support, amazing ideas, and encouraging advice. The unique environment he has set up for us at the Media Lab has allowed me to develop my interests, and explore this exciting field.

Table of Contents

1. Introduction	6
1.1. Timbre and Expression.....	6
1.2. Motivation and Overview.....	8
2. Previous and Related Work.....	10
2.1. Timbre Research.....	10
2.1.1. Early Attempts	10
2.1.2. Timbre Characterization.....	11
2.1.3. Theories of Timbre and Sound Color.....	12
2.2. Performance Systems.....	13
2.3. Analysis Methods.....	15
2.3.1. Fourier Transform	15
2.3.2. Wavelet Decomposition.....	16
2.3.3. Nonlinear Models.....	17
2.4. Pattern Recognition.....	18
3. Overview of the Clarinet.....	21
3.1. Introduction.....	21
3.2. Acoustics of the Clarinet	21
3.3. Miking the Clarinet.....	25
4. The Clarinet Timbre Analyzer.....	27
4.1. System Overview	27
4.2. Principle Component Analysis.....	29
4.2.1. Description.....	29
4.2.2. Representation Issues.....	30
4.3. Signal Level Analysis.....	31
4.3.1. Volume	31
4.3.2. Pitch Detection.....	33
4.3.3. Embouchure Pressure.....	36
4.3.4. Attacks.....	40
4.3.5. Vibrato	43
4.3.6. Using the System.....	44

4.4. Towards a Higher Level of Description	46
4.4.1. A Comprehensive Analysis	47
4.4.2. Adjective Transforms	47
4.5. Demonstration	49
4.6. Evaluation	50
4.6.1. Other Approaches and Real-Time Issues	50
4.6.2. Why PCA?	51
4.6.3. Possible Improvements	51
5. Related Issues.....	53
5.1. Generality.....	53
5.1.1. Other Instruments	53
5.1.2. The Voice.....	54
5.2. An Alternative to MIDI.....	55
6. Future Work and Conclusion	57
6.1. Possible Future Directions	57
6.1.1. New Synthesis Technology	57
6.1.2. Ensembles of Instruments	57
6.1.3. A Musical Expression Language.....	58
6.1.4. Teaching Tools	58
6.2. Concluding Remarks	59
Appendix	60
A.1. Principle Component Analysis.....	60
A.2. Video of Demonstration	64
Bibliography.....	65

1. Introduction

Manipulation of timbre plays an important role in determining the expressiveness of a performer's musical gesture. Much attention has been given to the use of expressive timing and dynamic deviations in performance; however, the study of timbre and how it relates to musical expression has remained an unsolved problem.

1.1. Timbre and Expression

The term *musical timbre* loosely refers to a sound's color or textural quality. It is often described by the somewhat unsatisfying definition: that which distinguishes two different sounds of the same pitch, amplitude, and duration. Research on the timbre of instrumental sound in the early 1960's, facilitated by digital techniques, quickly led to the conclusion that characterizing instrumental timbre is a difficult and often poorly defined problem. In his earliest of such explorations, Jean-Claude Risset [Ris66] performed a spectral analysis of trumpet tones and began to appreciate the lack of any general paradigm to analyze the qualities of a musical sound with a computer. More recently, McAdams has stated that timbre is the "multidimensional wastebasket category for everything that cannot be qualified as pitch or loudness." [MB79].

Most attempts to study timbre have revolved around spectral analysis because evidence from psychoacoustics shows us that in listening to complex sounds, the ear does something similar to spectral analysis [Moo89]. However, even after an exhaustive spectral analysis of a tone with a computer, finding the relationship between certain spectral features and their salient timbral qualities is difficult. This is partially due to the complex behavior of the ear which cannot be modeled by such a simple analysis.

The absence of a satisfactory definition of timbre is largely due to its very complex relationship with the physical (i.e., measurable) aspects of sound. Pitch, duration, and volume are all perceptual phenomena that have fairly simple physical correlates. Pitch is

related to the fundamental frequency of a tone¹; volume, to the amplitude of a tone's sound pressure wave. Timbre, on the other hand, is a multidimensional property of sound. Its relation to a tone's spectral properties, sound pressure amplitude, and its time evolution characteristics are not obvious.

In some qualitative sense, timbre is intimately related to musical expression. The art of orchestration involves the development of musical expression by effective use of different sound colors and the creation of meaningful sound textures by layering instruments of different timbres. It can be a process as simple as associating different motifs of a piece with different instruments to accentuate the themes' contrast as is commonly heard in the works of Mozart and Beethoven. It can also be a more subtle juxtaposition of different timbres that create a new kind of sound. Ravel's *Bolero* calls for two winds to play in perfect fifths creating the illusion of a single instrumental line with an organ-like texture.

One of the rudiments of computer music is the creative use of synthesized timbre, either alone or linked with the timbre of traditional acoustic instruments, as a means of musical expression [Mac84]. Throughout the later part of the 20th century, and particularly in computer music, the use of timbre as the primary means of musical expression has become a much larger part of composition than in the past. With the advent of sound creation and manipulation techniques such as additive synthesis, wave shaping, and frequency modulation, computer musicians now incorporate the creation of new timbres into the art of composition and bring about a new dimension of expressiveness not possible before.

On a different scale than the use of timbre in composition, a performer will use variations in a single instrument's timbre, in addition to dynamic and temporal fluctuations, to convey expressive gestures. The body of analytical work on timbre in the

¹ The fundamental frequency need not always be present, though. In the case of a *missing fundamental*, our ear derives the frequency of an absent fundamental pitch from the higher harmonics of a complex tone [Han89].

late 1970's (such as Grey [Gre75], Grey and Moorer [GM77], and Wessel [Wes79]), studied in detail the timbres of different acoustic instruments, but did not consider how musical expression and gesture are related to timbre. Trevor Wishart explored the meaning of musical expression and shape as a whole, including its relation to timbre and musical notation [Wis85].

1.2. Motivation and Overview

The motivation for the work presented here is the need to capture and describe, in real-time, the musical gestures in the performance of a single-line musical instrument. The gestures of an expert instrumentalist represent a sophisticated and expressive stream of information. Tapping into these well developed gestures is a necessity for any interactive computer music system.

This need is fulfilled in part by the clarinet timbre analysis system described in this thesis, though the general approach is not limited to the clarinet. A real-time analysis is presented of volume, pitch, vibrato, embouchure pressure², and note-onsets (attacks) of the clarinet. Though these parameters are by no means complete, they capture the essence of clarinet playing and provide information about the clarinet timbre.

The problem of identifying timbre is posed as a classification (i.e., pattern recognition) problem. The clarinet's sound is divided into steady state (sustained sound) and non-steady state (attacks and decays) regimes and a classifier is applied to each. Instead of attempting to overtly define a palette of timbres, a pattern recognition system is trained with a large set of examples that have human-salient properties. It is then up to the system, after the training process, to identify these properties in the context of a live performance.

² Embouchure pressure is chosen because it is an easily controlled aspect of clarinet playing that directly influences the clarinet timbre. It is a *physical* gesture that a performer will make to achieve a *perceptual* difference in tone color.

Once the sound stream of a performance is identified as being of a particular timbre, this information can be used as an input source to several types of musical systems, some of which have already been developed at the Media Lab. Functioning as a front end to a Hyperinstrument (see Section 2.2), the timbre classifier can provide an instrumentalist with more control than is possible with a traditional instrument. Work by Alex Rigopulos, Damon Horowitz, and myself is focused on creating a music generation system guided by real-time parametric control [Rig94]. The timbre of an acoustic instrument could act as a control parameter to manipulate and guide the music generated by such a system. For a demonstration of the system presented here, I composed a multi-layered, multi-timbral looping sequence over which a clarinetist can improvise. The shape of the looping sequence (such as its accent structure, layer-wise volume, and density) is manipulated by the gestures of the performer to create a convincing accompanying line (see Section 4.5).

Chapter 2 presents and analyzes some previous and related work on timbral analysis, interactive music systems, and pattern recognition problems. Chapter 3 provides a background on clarinet acoustics. Chapter 4 presents and evaluates the clarinet timbre analysis system. Chapter 5 discusses applications and the generality of this system. Chapter 6 addresses future work, and provides concluding remarks.

2. Previous and Related Work

2.1. Timbre Research

2.1.1. Early Attempts

The initial work of Helmholtz [Hel54] led researchers to the naïve belief that instrumental sound could be simply defined by its steady-state spectrum. Attempts to synthesize natural tones never produced convincing results because, though Fourier analysis can give precise sound spectra information, any notion of time evolution in this analysis is lost. It took Risset’s experimentation with trumpet tones,³ using time-varying spectral analysis techniques in the late 1960’s, to realize the importance of time dependence in our perception of musical sound [Ris66].

For example, Risset realized that there are many non-linearities in the brass player’s production of a note. In the attack of the note, non-harmonic spectral components appear. Onsets of overtones in an attack occur asynchronously. During the sustained portion of a note, an increase in volume causes energy in the various harmonic bands of the note to increase disproportionately. All these dynamic characteristics are crucial to the peculiar “trumpetness” of this sound (i.e., its timbre).

Risset’s research into timbre focused on its uses for computer music — especially for use in composition. Much of the work at IRCAM, conducted by Risset, Xavier Rodet, and Jean-Baptiste Barrière (for example, [Ris89], [Ris91], [DGR93], [BPB85]) has focused on synthesis methods that are both manageable, insightful into the structure of sound, and yield an effective sonic result. A common thread among this line of thinking has been the use of analysis-synthesis techniques — that is, using synthesis as the testing criteria of successful analysis. A technique that analyzes a certain sound, say a trumpet tone, and can then resynthesize it such that the ear cannot distinguish between the real tone and

³ Risset, with Max Mathews, went on to analyze tones of other instruments as well [Ris89].

the synthetic tone is considered successful. The amount of information in the representation resulting from analysis can be reduced based on this criteria (for example, phase information in the harmonic spectra is not salient to the ear, and can be disregarded). Compositional techniques then use the representation in these analysis-synthesis methods to yield creative musical results. For example, cross-synthesis involves the creation of a hybrid sound from the analysis results of two (or more) natural sounds.

The most common synthesis techniques have been additive synthesis, frequency modulation [Cho73], linear predictive coding [Moo79], wave shaping [DP89], and physical modeling (for example, [FC91]). In each case, there seems to be a tradeoff between the quality, richness, and generality of the produced sound, and the complexity of the algorithm, and the amount of information needed to perform a successful synthesis. Risset was able to mimic some trumpet tones accurately and with a manageable number of synthesis parameters, but at the expense of making very particular synthesis rules that applied to a very limited scope of synthesis problems (i.e., only certain notes of the trumpet). Frequency modulation can yield a spectrum of rich and varied sounds, though there is no physical analog to FM synthesis and creating particular sounds is a somewhat unintuitive “black magic” task.

2.1.2. Timbre Characterization

In the late 1970s, David Wessel and John Grey began addressing the problem of defining instrumental timbre by proposing a *timbral space* representation ([Gre75], [Wes79]). A multidimensional timbral space identifies timbre through a set of coordinates which presumably correspond to the degrees of freedom of the timbre. Similar to the concepts developed by Risset, the techniques Wessel used to create the timbral space were analysis via synthesis. Synthesized tones based on acoustic instruments were played to a group of listeners who made subjective measurements about how similarly the timbres of two instruments sounded. A multidimensional scaling algorithm was used to sift through the results and produce a timbral space, where all the timbral examples used could be plotted with respect to each other.

The scaling algorithm generates a space where the perceptual similarity of different timbres corresponds directly to their Euclidean distance in that space. Oboe tones were grouped close to bassoon tones, but far away from cello tones. Furthermore, the notion of interpolating between the discrete timbre points made sense so that it was possible to talk about timbre smoothly varying across the set of axes defined by the scaling algorithm.

The scaling algorithm produced a two-dimensional space which represented 24 orchestral instruments. The axes may be interpreted as the tone's *bite* and *brightness*. Bite relates to the evolution of harmonics in the attack of a note, while brightness refers to some average center of gravity of the spectra in the sustained portion of the note. Note that these axes were used to differentiate between different orchestral instruments (a *macroscopic* view of timbre), as opposed to differentiating between the possible palette of timbres in a single instrument (a *microscopic* view).

There was some discussion of designing control systems to effectively maneuver about this timbral space as an approach to composition. Interestingly, such a control system suggests using timbre space in the opposite direction — controlling musical forms based on an instrumentalist's modulation of timbre. A performance can be analyzed and projected into the timbre space representation as a method of shaping interactive compositions. This idea is further explored in Section 2.2, and is related to the primary goal of this thesis.

2.1.3. Theories of Timbre and Sound Color

A number of theories on timbre have emerged that attempt to better define this nebulous concept. The hope is that a better theoretical understanding will let us use timbre more efficiently in creative musical processes.

Wayne Slawson describes a theory of *sound color* which, surprisingly, is somewhat divorced from aspects of time evolution in the acoustic signal, though it is still useful in this limited scope ([Sla85]). The theory is heavily based on the notion that different vowel sounds and human utterances are the basis by which to describe different sound colors. Slawson defines four dimensions by which sound color can vary: *openness*,

acuteness, laxness, and smallness. These properties are by no means orthogonal (unlike Wessel's bite and brightness), though his approach is interesting because these axes are derived from reasoning about the structure of sound spectra, as opposed to collecting data and letting a multidimensional scaling algorithm define the axes automatically. This enables Slawson to define operations on sound color, such as translation, inversion, and transposition, much in the same way that these operations can act on pitch.

Approaching the problem from an entirely different perspective, Trevor Wishart claims that the inadequacies that we seem to have in describing timbre and, more generally, gesture in music stem from the inadequacies in our flat lattice-based notation system ([Wis85]). Just as the written word can never capture the subtleties of face-to-face communication, gesture and timbral indications are completely lost in common music notation. Furthermore, he claims that timbre and gesture have taken on a secondary role in Western music (as Pierre Boulez asserts), because of our limited notation system:

The spatialisation of the time-experience which takes place when musical time is transferred to the flat surface of the score leads to the emergence of musical formalism and to a kind of musical composition which is entirely divorced from any relationship to intuitive gestural experience[Wis85].

The emergence of computer music has helped to free composers from the bounds enforced on them by a two dimensional notation system. For example, synthesis techniques have given composers a vast palette of synthetic sound textures which they can incorporate as primary elements of their compositions. Notions such as timbral tension and release, and timbral motion become meaningful.

2.2. Performance Systems

Beginning in 1986, the Hyperinstrument Group at the MIT Media Lab, led by Tod Machover, has conducted research on a new concept in instrumental performance — Hyperinstruments [Mac92]. The central idea is to let the emerging power of computers enhance the performance aspects of music making by giving the performer new levels of musical control not possible with traditional acoustic instruments.

A Hyperinstrument begins with an acoustic instrument that is equipped with various sensors to monitor instrumental performance. In the three Hyperstring projects (*Begin Again Again...*, *Song of Penance*, *Forever and Ever*), these consist of bow sensors, a wrist inflection sensor, and the analysis of the instrument's acoustic signal [Ger91]. These sensors send streams of information to a computer which performs real-time gesture analysis and interpretation of the sensor data. This analysis serves to shape the live musical result.

In *Begin Again Again...*, the cello piece, some qualities of timbre are derived from knowledge about the physical location of the bow. Bowing a cello closer to the bridge (*sul ponticello*) produces a harsher, brighter tone than bowing it closer to the fingerboard (*sul tasto*). Some aspects of performance are measured by correlating characteristics in the bow movement information with energy measurement information such as the detection of bowing style. For example, if the cellist's technique is particularly *marcato*, the computer may enhance this gesture with a burst of percussive attacks. A more *legato* bowing style may introduce an accompanying smooth sound wash which exaggerates this gesture.

With the exception of the piano and other percussive instruments, capturing the gestures of a performing musician is a difficult problem. A MIDI (Musical Instrument Digital Interface, an industry standard) keyboard suffices to provide all the musical gestures expressible on a piano (i.e., note selection, note velocity, and timing information). However, in a non-percussive acoustic instrument, such as a cello or clarinet, pitch and loudness selection only constitute the beginning of expressive possibility. Subtle gestures associated with the production of sound and the complex relation between timbre and expression is at the heart of what makes these instruments interesting, and is what drives composers like Machover to tap into that expressive power.

This thesis grapples with the same sorts of issues brought up in Hyperinstrument research. Its goal is to make a contribution to the real-time analysis of instrumental gesture in hopes of deriving a sophisticated control mechanism. However, unlike the

Hyperstring projects, this analysis is performed solely on the acoustic signal of the clarinet and is unaided by other physical sensors.

All the Hyperinstrument work has focused on giving expert musicians additional levels of control. Recently, some systems have been developed that attempt to derive musical intention from non-expert musicians. DrumBoy [Mat93], is an interactive percussion system which operates on different levels of human control. Musical ideas may be input with a high degree of control by actually specifying individual notes and rhythms. On a more accessible level for non-musicians, percussion rhythms and styles may be modified via *adjective transformer commands* such as “more mechanical,” or “increase the energy.”

The seed-based music generation system [Rig94], also developed at the Media Lab, applies the same DrumBoy ideas to melodic (i.e., non-percussion) music. Its research was motivated by the need to express musical intent to computers in a way that is removed from the details of those intentions. A parametric representation for certain types of popular music was developed which allows a user to manipulate musical ideas on the level of activity, harmonic coloration, syncopation, and melodic direction. This system is useful for both amateurs who can “navigate” through a musical space by using high level interfaces (such as joysticks), and by composers, who may manipulate musical material without being bothered by the details of note by note composition.

2.3. Analysis Methods

2.3.1. Fourier Transform

The most commonly used signal processing tool in music sound analysis is the Discrete Fourier Transform (DFT), which has a fast $O(N \log N)$ algorithmic implementation, the Fast Fourier Transform (FFT). The DFT can exactly represent any arbitrary periodic signal as a sum of complex exponentials (or sinusoids) of related frequencies. The DFT would seem ideal for spectral analysis of music, except that it assumes the periodic signal to be studied is of infinite duration (technically, the sinusoids in the DFT summation are also of infinite duration). In other words, perfect spectral decomposition is provided at the cost of a complete loss of time information. To alleviate this problem,

the Short Time Fourier Transform (STFT) is employed, which provides a tradeoff between time resolution and frequency resolution of the analysis. This tradeoff is realized by taking a small time slice of the signal (a window), whose frequencies are assumed to be time-stationary. FFTs are performed per window, which slides across the signal. A whole collection of these analyzed windows is often displayed as a spectrogram. The window length is the “knob” which favors either accurate timing information (short windows) or accurate frequency information (long windows).

Many applications have relied the FFT, mostly because it is easy to use, and has a very fast implementation. It is often used in conjunction with other methods. For example, Smith and Serra [SS90] created a spectral model of harmonic sound by describing the time-varying spectral components of the sound and also estimating the stochastic noise of the signal. Depalle, García and Rodet [DGR93] used Hidden Markov Models on top of spectral decomposition to track time-varying partials.

The FFT does have its drawbacks, however, exactly because of the time-frequency tradeoff. The time evolution of spectral components during a sharp attack of a note are essential to the characterization of that note. Using the long STFT window necessary to resolve individual harmonics will smear the temporal details of the harmonic onsets which may be important, say, for attack differentiation.

2.3.2. Wavelet Decomposition

The time-frequency tradeoff in spectral analysis is unavoidable from a theoretical point of view: the more time we have to observe a stationary signal, the more accurately we can pinpoint its frequency components. An interesting aspect of human audition is that accuracy depends on frequency. We tend to hear pitch as a logarithmic function of frequency, so that for the same perceived interval between two notes, high frequencies need to be much farther apart than low frequencies.

We can make an analysis consistent with this phenomena by a technique known as *wavelet decomposition*. One can think of wavelet decomposition as a spectral analysis method (just like the STFT), except that window length is a function of frequency.

Higher frequencies are analyzed by shorter windows so that their spectral accuracy is compromised for greater timing accuracy in much the same way that the human ear behaves.

Psychoacoustic studies have shown that human ear is subject to a *critical band* of hearing sensitivity. The critical band is a frequency range (usually of about a third of an octave) over which the ear cannot discriminate the presence of particular frequencies [Moo89]. This implies that for complex (i.e., rich in harmonics) tones, any overtones which are closer than a third of an octave will all be grouped together as one energy lump in the vicinity of that frequency. Any details, such as relative strength of harmonics within the critical band, will be merged. This phenomena has motivated the implementations of third-octave wavelet transforms which take advantage of this “deficit” in our hearing ([Ell92]).

In some particular cases, fast butterfly algorithms exist for wavelet transforms ([Hol89]) which may be suited for a real-time analysis.

2.3.3. Nonlinear Models

Both the Fourier transform and the wavelet decomposition are linear analysis techniques. They both form an orthogonal basis of frequency-type elements by which to construct a time-domain signal as a linear combination of these elements. This is not a bad assumption, since a good many sounds can be represented in a linear fashion. The popular synthesis technique of additive synthesis, for example, is directly related to linear spectral analysis. It essentially states the inverse Fourier transform (technically, the inverse sine transform) by constructing a signal as a sum of sinusoids of different amplitudes:

$$s(t) = \sum_i A_i(t) \sin(f_i(t) + \theta_i)$$

However, careful study of acoustic instruments shows that they far from linear. Even though the source/filter model (see Section 3.2) is a linear one, it is only a very crude approximation to the behavior of real sound sources. Even a simple energy source, the

reed/mouthpiece section of the clarinet, is highly non-linear: an increase in air pressure across the reed does not proportionately increase the amplitudes of the modes of oscillations of the reed [MSW85]. This is partially why the timbre of the clarinet changes with different volume.

New approaches to the study of instrumental timbre have surfaced recently that use the non-linear time series analysis of *embedology* [Ger92]. Embedology represents a signal $s(t)$ as an embedding in a multidimensional lag-space. $s(t)$ is represented in lag-space as a coordinate $\bar{z}(t)$, whose components are lagged values of $s(t)$:

$$\bar{z}(t) = (s(t), s(t - \tau), s(t - 2\tau), \dots, s(t - (d - 1)\tau))$$

This deceptively simple representation can provide information-theoretic properties of the signal $s(t)$, such as the number of degrees of freedom of the system that produced $s(t)$, and the entropy of the signal. The shape that $s(t)$ forms in lag-space, known as the *attractor*, has the information necessary to create a non-linear model of $s(t)$ ⁴. However, most sounds analyzed by embedology look nothing like linear planes. Recent work by Eric Métois ([Met95]) conducted at the MIT Media Laboratory studies these non-linear models as a way to analyze instrumental sound. He constructed a non-linear model based on one second of a violin note, and then resynthesized the sound based on that model. The model faithfully reproduced the violin note, though the synthesis process is very compute intensive. Unfortunately, a real-time analysis/synthesis model based on these methods is probably still far away.

2.4. Pattern Recognition

The fields of pattern recognition and decision theory⁵ are widely used in many of today's engineering applications, ranging from military uses in radar systems to burglar alarms to medical diagnosis systems. With the advent of digital computers, its

⁴ If it was linear, the attractor would simply reside in a hyperplane in N-space.

⁵ Pattern Recognition is a very wide area of research. The interested reader is referred to [DH73], [SB88], and [The89] for more detail.

applications have expanded considerably to areas of speech recognition systems, optical character recognition, and robotic vision systems. The basic goal of a pattern recognition system is to characterize a particular phenomena or event based on a set of measured data that are somehow correlated to the phenomena. Typically, there is some uncertainty in the measurement, so pattern recognition schemes are often statistical in nature. In classification algorithms, a subclass of pattern recognition, the purpose is to categorize an unknown event into one of several prespecified classes based on an event's data measurements.

Typically, the set of measurements or *observables* contains an unwieldy amount of data. This data can be represented as a vector in a very high dimensional space where each vector component contains a single datum. This observation vector is then transformed into a lower dimensional *features vector*. The features vector attempts to capture most of the important information present in the original measurement data while reducing the dimension necessary to present that information. Figure 2.1 shows a block diagram for a classification scheme. The difficulty in developing a good classifier lies in coming up with a good *feature extraction procedure*. Finding an appropriate representation of features is at the heart of classification algorithm design.

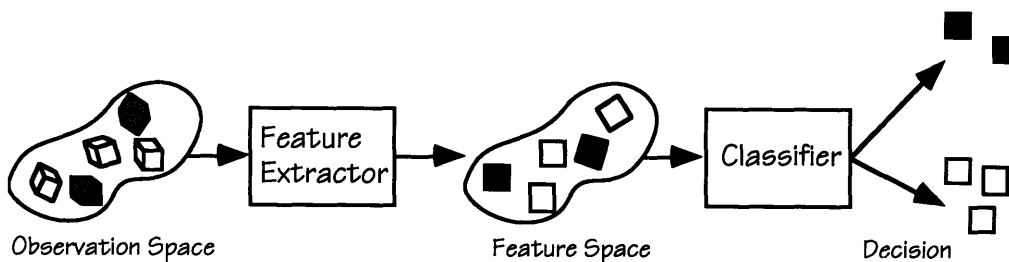


Figure 2.1. A block diagram for a classification system. Observations to be classified are typically described by a high dimensional vector. A feature extractor captures the important information from the observation space and projects it into a lower dimensional space. A classifier makes the final decision.

Classifications where the set of possible classes is known in advance are cases of *supervised pattern recognition*. Here, the methodology involves training a classifier with example data which we know belong to a particular class. Then, unknown data presented to the trained classification system will be categorized into one of the *a priori* learned classes. When the types of classes and/or the number of classes are not known *a*

priori, the classifier uses *unsupervised learning* or *clustering* algorithms to guess at how the data distributes into a set of classes.

3. Overview of the Clarinet

3.1. Introduction

During the end of the 17th century, Mozart first popularized the clarinet by introducing a pair of clarinets into all of his later symphonies. Towards the end of his life, he wrote two of his most acclaimed works for the clarinet: the *Concerto* (K. 622), and the *Clarinet Quintet* (K. 581). Composers since Mozart have provided the instrument with a repertory that in quality and variety is equaled by no other wind instrument.

In jazz, the clarinet was heavily used during the Big Band era, popularized by jazz greats such as Benny Goodman and Woody Herman. It is an integral part of Klezmer music and is often featured in Dixieland. It is known for a wide pitch range spanning nearly four octaves, a large dynamic range, and a substantial palate of different tone colors.

For the purposes of this thesis, the clarinet is useful because it is essentially a monophonic instrument, with a wide range of player manipulated expressive gestures. It lends itself well to the study of expression in single line instruments. Its range is comfortable for a real-time signal processing application where a sufficiently low sampling rate can be used without losing too much information about the signal. Additionally, the author plays the clarinet — a significant factor since experimentation with the clarinet and the testing of the various algorithms presented here were copious and unrestricted.

3.2. Acoustics of the Clarinet

In general terms, the production of sound in any musical instrument begins by applying energy to a sound source⁶. The sound source couples into the instrument's resonating body which transforms the sound somehow and radiates it into the room. The particular

⁶ This section is only meant to be a brief overview. See [Ben76] for a more complete discussion.

acoustical properties of the room further modify the sound which then enters our ears. The traditional mathematical model for an acoustic instrument is the source/filter model (Figure 3.1). In stringed instruments, energy is applied to the string by the bow (usually represented by a slip/stick model) which couples into the instrument by the bridge and sound post. The instrument body is described by a linear filter which amplifies certain frequencies and suppresses others. The particular shape of this filter is known as the *formant* structure of the resonating body.

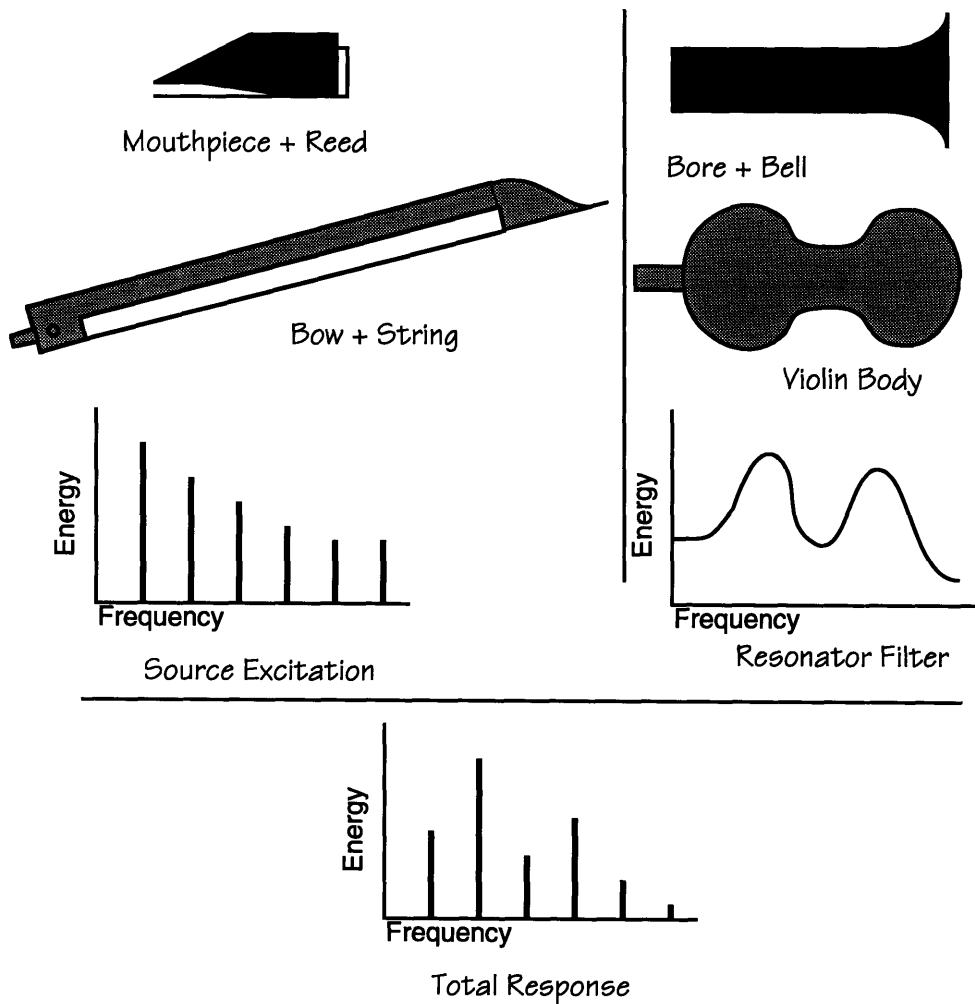


Figure 3.1. The source/filter model. The clarinet's reed and mouthpiece and the violin's bow and string are the energy sources. They excite the instrument bodies with a spectrum that evenly decays with higher frequencies. The formant structure of the instruments' bodies act as filters on the energy source and shape its spectra.

A wind instrument's energy source is the reed (in the case of the bassoon, oboe, and English horn, a double reed). Air being blown past the reed and into the bore of the

instrument causes air pressure waves which couple directly into the up and down motion of the reed. The instrument's body can be described by a filter which modifies the frequencies of the reed's oscillations. Fingering a note on a woodwind establishes an effective bore length. This length is directly related to the allowable modes of oscillation of the reed because of the boundary conditions imposed at each end of the bore. As a result, a certain fingering allows for the production of a discrete set of frequencies known as the harmonic progression (or sometimes overtones) of a fundamental pitch. The formant structure of the instrument's bore then modifies the amplitudes of each of these harmonic frequencies which results in a characteristic sound for a particular instrument of the woodwind family.

For example, on a saxophone, fingering a (sounding) G₂ will allow the reed to oscillate at 196 Hz, 392 Hz, 588 Hz and on up, with the frequencies increasing by a constant value for each mode of oscillation. A regularly blown note (assuming no lip pitch bend) will sound like the note G₂, and will be composed of a series of overtones above the fundamental frequency. However, if the saxophone is overblown (or the register key is depressed) with this fingering, the reed will find a stable mode of oscillation with a fundamental frequency of 392 Hz, causing the sounding note to be a G one octave higher than the first register note. Similarly, it is possible to twice overblow on a G fingering and hear the D a twelfth above the first register G.

The clarinet is a bit unusual as a member of the woodwind family, because it is the only instrument whose overtone series consists only of the odd harmonics (fundamental frequency, third overtone, fifth overtone, etc.) [Bac73]. In the clarinet, the same fingering for a particular note in the first register will produce a note sounding a twelfth above it in the second register (one octave plus a fifth). As in the case of the saxophone, the flute, oboe, and bassoon all leap a single octave from the first register to the second.

The timbre of the clarinet (that distinguishes it from other instruments) is largely a function of its formant structure. However, unlike string instruments whose body remains the same shape, the "shape" of the clarinet bore is constantly being modified by different fingerings. Playing different notes on the clarinet changes its effective bore

length, but also alters the formant structure of the bore itself. Discussing the timbre of the clarinet becomes a difficult issue because if one argues that timbre is directly related to the formant structure of the bore, every note would tend to have a different timbre. Of course, when we hear a clarinet, we identify the sound produced as a clarinet timbre irrespective of what note is being played at the time. A closer look at the particular timbral characteristics of the clarinet is necessary.

The lowest register of the clarinet or the *Chalumeau* register (which ranges from written E₂⁷ to written F#₃) has by far the richest sound color. Many of the early composers took advantage of this sound by writing long lyrical passages which show the earthy tone quality of this register. A spectrogram of any note in the lowest octave of the clarinet will reveal almost no energy in any of the even numbered overtones. A signal composed of strictly odd harmonics will resemble a square wave and has a characteristically deep and open sound, as opposed to the more nasal sound exhibited by an oboe. The individual notes in the low register will have slight timbral variation which depends on the more minute details of the coupling between embouchure, reed, and bore which is not addressed by the simple source/filter model described above.

The middle or throat register of the clarinet (which ranges from written G₃ to written B^b₃) is much less rich, and tends to be more airy and unfocused. The higher registers of the instrument (ranging from written B₃ to C₆) can be bright and sometimes even piercing. Above the first register, all the frequencies in a pitch's overtone series are accounted for, producing a generally brighter sound.

Aside from the actual sound production mechanism in the clarinet is the important role of room acoustics and sound radiation. The sound radiation from the clarinet happens at three different locations. The bell of the instrument transmits many of the higher harmonics. The openings in the finger holes transmit more of the lower harmonics,

⁷ The clarinet, like the saxophone, is a transposing instrument. Playing a pitch indicated in a score will actually produce a pitch two semitones lower on the B^b clarinet. For example, playing a *written* C will produce a *sounding* B^b (hence the name).

particularly when they are next to a closed tone hole. The mouthpiece and reed area contribute some characteristic breath noise and high frequency non-tonal sound from the tongue hitting the reed. As a result, the positioning of the clarinet with respect to a listener's ears (or with respect to a microphone) determines the type of sound that will be perceived.

It is interesting to note that the clarinetist usually hears something quite different from what an audience member hears. This is mostly due to the clarinet's constant positioning with respect to the performer's ears. The performer has a harder time hearing the higher frequency parts of the tone because the bell is furthest away. This is especially noticeable during some orchestral performances where the composer calls for the clarinetist to raise the bell of the instrument (as in many of Mahler's symphonies). The audience hears a much brighter, louder tone than it would normally, whereas the clarinetist perceives little timbral shift.

3.3. Miking the Clarinet

When considering how a timbre analysis system should work, an important issue rests on the phenomena described above and poses the question: what is the best way to mike the clarinet? If the system is meant to interpret the clarinet sound as an audience member would, then the microphone should be placed away from the instrument, in a position similar to where a listener would be. However, if we desire a reliable gesture control and interpretation engine, the microphone should be fixed with respect to the instrument so that slight physical movements will not appreciably vary the sound at the input stage.

The clarinet is a painfully difficult instrument to mike properly [BL85]. The desired miking scenario for a studio recording of the clarinet would have one microphone a few meters away from the instrument, so that an overall balance between the different sound sources is achieved. However, for the purposes of computer analysis, the clarinet must be close-miked to avoid lots of surrounding noise and to reduce room reverberation.

The final setup (after much experimentation) uses a clip-on instrument microphone which is fixed to instrument and captures most of the sound from the tone holes. Some high frequency components from the bell are not as loud, though they are still picked up. An unfortunate side effect of this setup is that key click noise is somewhat amplified in the microphone because it is in direct contact with the clarinet.

4. The Clarinet Timbre Analyzer

4.1. System Overview

The purpose of the clarinet timbre analysis system is to extract meaningful musical gesture information from a real-time clarinet performance based solely on an analysis of its acoustic signal. *Musical gesture* is something of a catch-all phrase that refers to the language by which a musician can communicate to an audience of listeners or to other musicians. In the scope of this thesis, musical gesture is essentially a collection of information, parameters, and events varying over time, that describe how the clarinetist is playing. These parameters are at an intermediate level of description. They are higher than a “voltage over time” signal-level description, but cannot describe musical emotion, phrasing, intent, or any genre specific musical qualities. The level of information is designed to be appropriate for inter-computer communication, comparable to that of MIDI, but encompassing more than the limited note on/note off capabilities of MIDI. The information will delve below MIDI-type description into nuances of different possible clarinet attacks (i.e., more than just one bit for a “note-on”), and a little above MIDI to describe longer term trends in a musical performance (see Figure 4.1).

In the scope of this thesis, it is impossible to perfectly extract every musical gesture produceable on the clarinet. Just the task of enumerating all such gestures is a lengthy process, though some of the common ones are mentioned throughout this chapter. A derivation of the following gestures is presented: volume, pitch, embouchure pressure, attack type, and vibrato.⁸

⁸ Vibrato is actually a two dimensional parameter consisting of vibrato *speed* and vibrato *amplitude*.

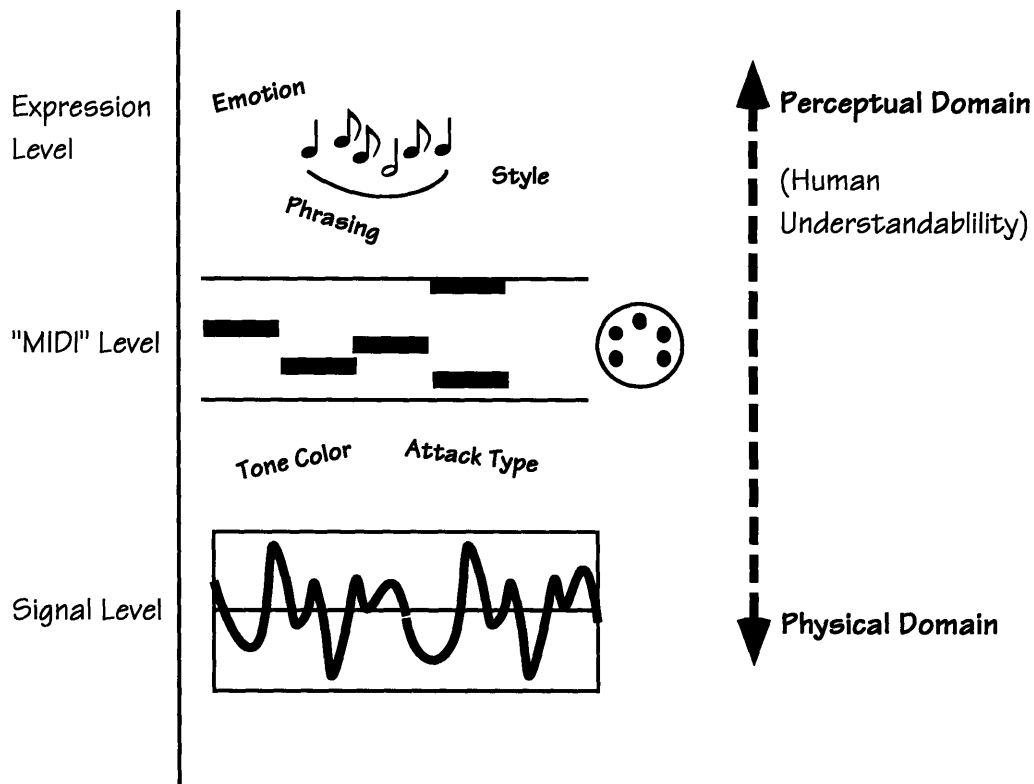


Figure 4.1. A hierarchy of the different levels of musical information.

Even though these five parameters are a subset of all possible gestures, they still capture the essential “ingredients” of clarinet playing. In particular, embouchure pressure, attack type, and arguably, vibrato, constitute ways in which a clarinetist can control timbre.

The goal of this system is not a perfect reconstruction of the clarinet sound. Rather, the criteria for a successful analysis system is a player’s ability to consistently control some other music generation system.

Figure 4.2 shows a schematic for the basic setup of the system. A Beyerdynamic monophonic dynamic instrument microphone is mounted on the bell of the instrument, with the mike head pointing at the tone holes. The mike-level signal is fed to a Symetrix microphone pre-amp, which is then fed to a Silicon Graphics Indigo Machine. The Indigo samples the acoustic signal at 11.025 kHz (after first running an antialiasing filter on the signal), and all analysis is performed in real-time by manipulation of the sampled data as it comes in to the machine. After the analysis, the Indigo sends MIDI commands to communicate with other music applications. The MIDI stream encodes gesture

parameters (described in Section 4.3) of the live clarinet performance as control-change messages.

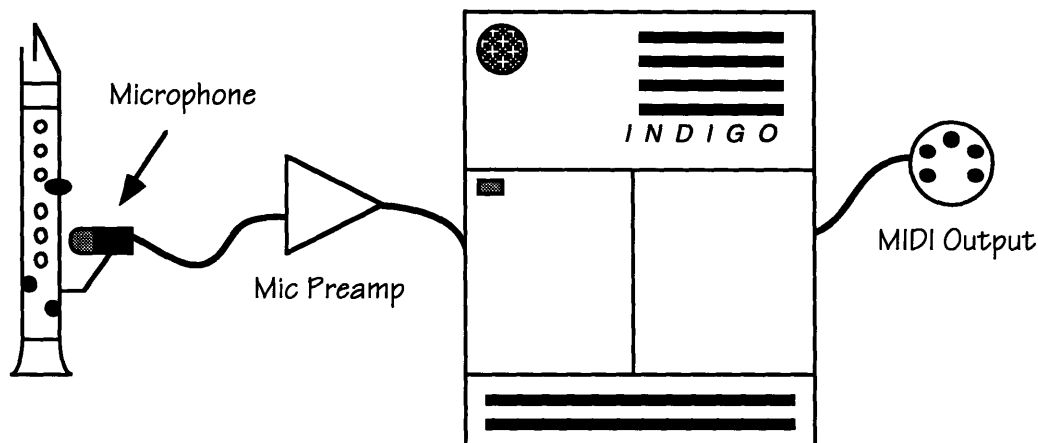


Figure 4.2.
A schematic of the basic setup of the clarinet timbre analysis system.

The clarinet timbre analysis program is coded entirely in C on a Silicon Graphics Indigo R4000 Workstation. Some mathematical algorithms (such as the Jacobian method for finding the eigenvalue of a matrix) are borrowed from Numerical Recipes [Pre92].

4.2. Principle Component Analysis

An essential component to solving the problem of real-time gesture recognition of clarinet performance is the classification scheme that was introduced in Section 2.4. Specifically, the measurement of timbral change in sustained notes, and the discrimination between different types of clarinet attacks is achieved by the use of *Principle Component Analysis* (PCA), a supervised learning technique that brings out the differences between sets of features that most discriminate between a set of classes. The algorithm used here is similar to Turk and Pentland's method for distinguishing between different facial images ([TP91]). A detailed derivation of PCA is provided in Appendix A.1. A brief description follows.

4.2.1. Description

The goal of Principle Component Analysis is to facilitate the comparison of an unknown observation with a set of known training observations. Each training observation is

associated with one of several classes. Each class represents a salient property that we are interested in measuring. For examples, one class might be “soft attack,” while another may be “hard attack.” The PCA would then classify an unknown observation as being most similar to one of the *a priori* learned classes.

In the training process, the entire collection of training observations must be transformed into a *feature space*. In this space, each observation is a single high dimensional *feature vector* that contains important data from the original observation. What constitutes “important data” is not relevant to the PCA algorithm per se, but is crucial to a successful classification (see below).

A series of mathematical operations known collectively as *eigenvalue decomposition* transform the set of training vectors into a lower dimensional space, where the statistical differences between these vectors is encapsulated. Redundant information has been filtered out.

In the classification process, an unknown observation is transformed into this lower dimensional space and is measured against the training set. A decision is made based the distance between the observation sample and the average of the training vectors in each class.

4.2.2. Representation Issues

The PCA classifier is elegant because it is somewhat removed from the peculiarities of particular pattern recognition problems. The algorithm can be used to differentiate between facial images, pieces of chocolate, or instrumental timbre. However, the “catch” in PCA is the representation of the data. More precisely, the exact procedure of transforming observation data into a set of features will determine the success of this classification.

PCA forms a basis by which to represent a series of features as a linear combination of elements in the basis. If the set of features corresponds to salient properties of the class in some terribly non-linear way, PCA may not work very well. If, for example, the

features consist of a set of energies of some audio signal over time, it matters if the energies are represented as straight energy levels (linear voltage) or in decibels. A linear variation in a straight voltage feature representation will yield a linear scaling of all the feature vectors, while a decibel energy representation will probably just offset the feature vectors by an additive constant (i.e., a single principle component). Therefore a decibel representation of energy is preferable because the PCA will be less sensitive to general loudness such as the microphone amplification.

4.3. Signal Level Analysis

The parameters derived from the clarinet's acoustic signal are described in detail below. A summarizing block diagram is show in Figure 4.3. Most of the mathematical algorithms are standard, and can be found in any signal processing texts (see for example, [OS89]). Pitch-synchronous energy detection was motivated by some early descriptions of pitch-synchronous spectral analysis [Ris91]. PCA is not a new classification method, but its application to timbre is novel. Autocorrelation pitch-trackers are commonly known. The specific implementation here is adapted from the violin pitch tracker developed by Eric Métois. Using a high-pass filter for the vibrato detector is not an especially brilliant discovery, though I have not seen this idea before.

4.3.1. Volume

One of the most accessible expressive qualities of any monophonic instrumental line is volume. After pitch and rhythm, it is the next important feature when a written notation attempts to indicate some kind of expressive intent. On a macroscopic level, dynamics of a long phrase are indicative of direction of motion and intensification (or release). Dynamics help to clarify the apex of a phrase. On a microscopic level, changes in volume function as accents. Accents help shape a line rhythmically by stressing important temporal points. They can convey the jaggedness or smoothness of a musical phrase, and can support or contradict the inherent pulse that we feel when we listen to a musical phrase.

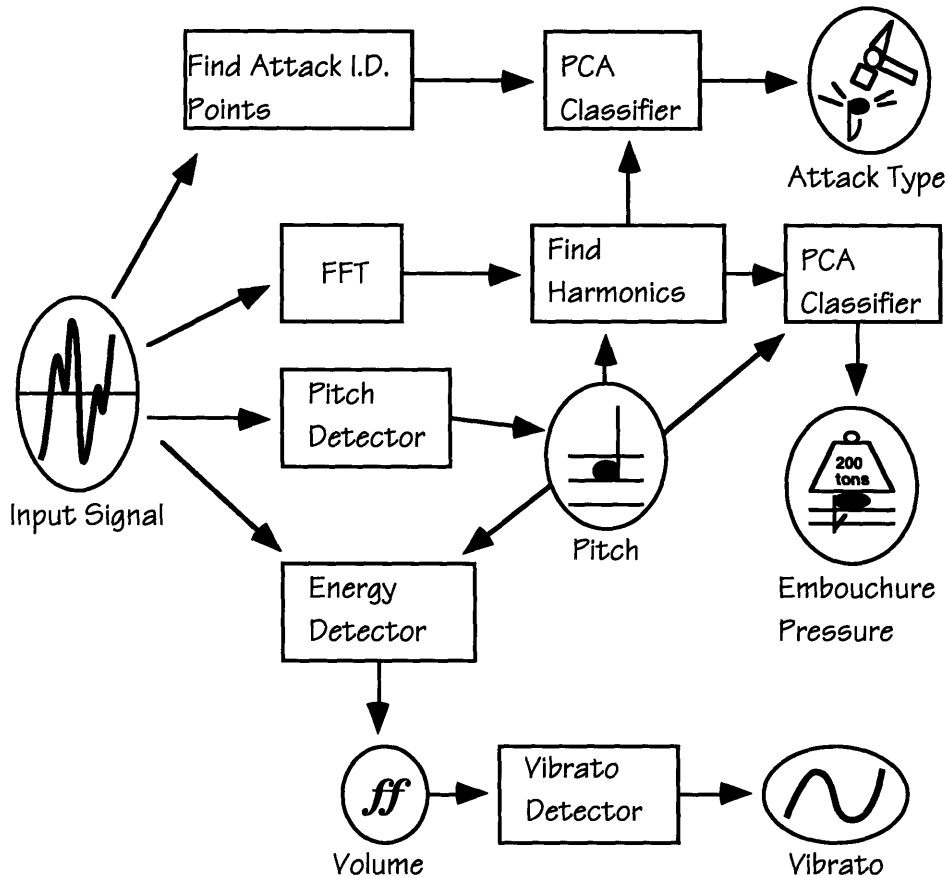


Figure 4.3. A block diagram of the modules in the clarinet timbre analysis system. Module dependencies are depicted by arrows.

Studies of audition have shown that loudness perception of pure tones is a complicated function of both energy and frequency [Moo89]. However, the loudness or volume of a complex tone can be approximated as a logarithmic function of its energy (i.e., the *intensity*):

$$I = 10 \log \left(\frac{E}{E_{\min}} \right)$$

E_{\min} is defined as the minimum threshold energy level of hearing.

The energy of an acoustic signal is easy to measure. In this system, it is simply the expected value of the square of a set of samples inside a window of length w . For an audio signal $s(t)$, sampled to produce a discrete time sequence $s[n]$, the average energy over a window of length w is:

$$E = \frac{1}{w} \sum_{n=T}^{T+w-1} s[n]^2$$

During any windowing operation in signal processing, there is an implicit assumption of stationary. When we window a time-varying non-stationary signal, we are assuming that within the small time span of the window, the signal's properties do not change much, and in fact, calculating information about a signal inside a window is equivalent to making that calculation assuming an infinite signal of a periodicity equal to the window length.

In the simple case of measuring the energy of a (mostly) periodic signal, such as the acoustic sound of the clarinet, an error is expected if the window length w is different than the periodicity of the signal. For example, measuring the expectation of the energy of a sine tone with a period of p by using a window of length $p + \Delta p$ will cause an error which oscillates with frequency $\frac{1}{\Delta p}$. This error can be eliminated by the use of pitch-synchronous energy detection. The operation is simple. The window length w is picked to be an integer multiple of the period of the signal. This periodicity information is provided by the pitch detector (see section 4.3.2).

4.3.2. Pitch Detection

The pitch detector used in this system is by no means meant to be revolutionary, though it turned out to be a useful tool for the measurement of other signal properties (such as energy detection, mentioned above). The pitch detector used here is a time-based cross-correlation filter⁹. In its essence, it attempts to maximize the correlation of the signal with a shifted version of itself where the variable of optimization is the shift length (the signal is assumed to be zero-mean):

⁹ This pitch detector is heavily based on the methods used for the violin pitch detector developed by Eric Métois for Tod Machover's *Forever and Ever*.

$$\rho = \frac{\sum_{n=T}^{T+w-1} s[n]s[n+\tau]}{\sum_{n=T}^{T+w-1} s[n]^2}$$

For maximum correlation ($\rho = 1$), τ is an integer approximation to an integer multiple of the period. It is first necessary to know a lower limit on the expected pitch, because the pitch detection algorithm needs a bound on w , the window length. The B-flat clarinet's lowest note is a written E2, sounding D2, which has a fundamental frequency of 146.8 Hz. The SGI samples at 11.025 kHz, so this note has a periodicity of approximately 75 samples. w is chosen to be 75 samples. It is not essential for w to be equal to or greater than the largest expected period, but ρ tends to be more accurate with larger w . This way, we know that the correlation is taken on at least a full period of the incoming signal.

Computing ρ is an expensive operation. Having to compute ρ for all values of τ between the minimum and maximum periods becomes unwieldy. To reduce the size of the computations, it is first necessary to identify likely candidates for τ based on the overall shape of the audio signal. A peak detector is passed across the signal which identifies local maxima. Candidates for τ are simply differences between these local maxima, with the added restriction that τ is between the minimum and maximum periods, and that differences are only taken between maxima of approximately the same value (see Figure 4.4). For each candidate, ρ_τ is computed and ranked from highest value to lowest value. It does not suffice to simply choose the period with the largest correlation coefficient because frequencies higher than twice the lowest possible frequency (i.e., higher than D3), will have an ambiguous period. Those frequencies will show high correlation for integer multiples of their associated period (a signal periodic in τ is also periodic in 2τ). Therefore, it is necessary to detect these redundancy situations, and choose the lowest possible τ which does not have an integer divisor. A simple heuristic suffices to achieve this, and seems to work reasonably well.

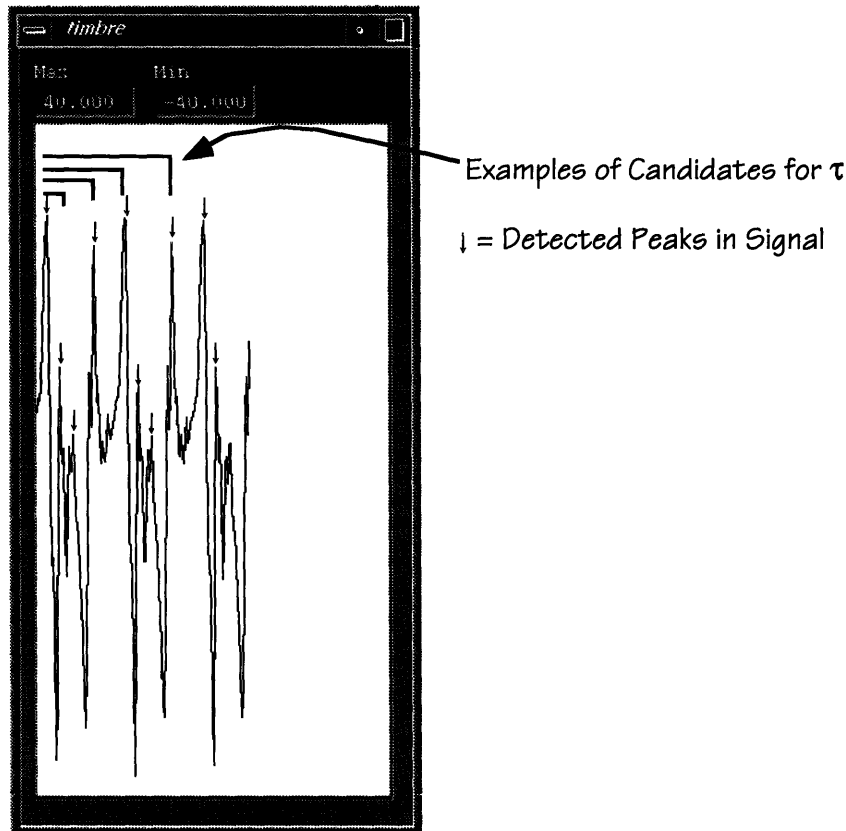


Figure 4.4.
An example of the candidates picked by the pitch detector

A problem with the pitch detector as it stands so far is that τ can only be accurate to an integer period. For high frequencies, this can be a severe problem since integer pitch resolution starts to acquire errors on the order of semitones. To cure this problem, a linear interpolation between two adjacent values of the integer period (i.e., τ and $\tau + 1$) is used to achieve higher resolution.

Let s_n be a vector which contains the first w samples of the signal $s[n]$:

$$s_n^T = [s[n] \ s[n+1] \ \dots \ s[n+w]].$$

s_0 contains the reference samples, s_τ contains samples of the signal one integer period later, and $s_{\tau+1}$ is a shifted version of s_τ by one sample. All these vectors point in approximately the same direction. If the period of the signal was in fact an integer period, s_0 and s_τ would point in exactly the same direction (ignoring signal noise for now). However, let us assume that the actual period of the signal is $\tau + \alpha$, where α is a

number between 0 and 1. s_0 now points somewhere in between s_τ and $s_{\tau+1}$ and can be approximated as follows:

$$s_0 = (1 - \alpha)s_\tau + \alpha s_{\tau+1}.$$

This is simply a linear interpolation between s_τ and $s_{\tau+1}$. In the limiting cases, it is obvious that $s_0 = s_\tau$ for $\alpha = 0$ and $s_0 = s_{\tau+1}$ for $\alpha = 1$. Finding α , the “mix” ratio, is the result of a simple projection:

$$\alpha = \frac{d_0^T d_1}{d_0^T d_0}$$

where $d_0 = s_\tau - s_{\tau+1}$ and $d_1 = s_0 - s_{\tau+1}$.

4.3.3. Embouchure Pressure

For the moment, let us focus our attention on a particular aspect of the clarinet timbre: the steady-state sound color (here, *sound color* is used in the same sense that Slawson defines sound color; see Section 2.1.3). After all transients from the attack of a note have passed, and the clarinetist is blowing a steady stream of air into the mouthpiece without fluctuating anything about the embouchure, air pressure, or diaphragm, the sound color reaches an equilibrium where it does not vary appreciably. Given this situation where we can describe a steady-state sound color, we are at liberty to talk about how sound color varies during different moments of a clarinet performance.

The obvious differences in the sound color of a clarinet occur between different registers of the instrument. A steady tone, produced at similar dynamic levels, will sound deep and dark in the first register, brighter in the second register, and quite shrill in the third register. The notes in a particular register will have slightly different sound colors based on the peculiarities of the clarinet tone hole placements and fingering techniques. Generally, notes near the top of the first register (the *throat register*) sound airier and have a less focused tone. Some close intervals have a surprisingly large discrepancy in sound color such as the low (written) G2 to G#2. These differences can be maker

dependent (such as the characteristic differences between Selmer and Buffet clarinets), clarinet dependent, and player dependent.

Next is the variation in tone color that is coupled with the loudness of a note. Notes played fortissimo will, in general, be richer in higher harmonics than notes played pianissimo. This is due to the non-linear properties of the clarinet reed and mouthpiece structure ([BK88]).

Finally, players can vary the tone color of their sound by changing embouchure, diaphragm tension, and finding alternate fingerings for similarly pitched notes. Often, in classical styles of playing, clarinetists will modify these aspects of their playing technique to balance the tone coloration change imposed by the instrument itself in an attempt to achieve an even sounding tone color. In jazz and some non-Western genres, it seems quite the opposite is true. At any rate, clarinetists have some degree of control over the sound color produced by their instruments.

For the clarinet timbre analysis system, one of the player-controlled timbre modification techniques, embouchure pressure, is the candidate for this computer analysis. Embouchure pressure is easily controlled by proficient clarinetists and has a direct influence on the clarinet's tone color. However, because of the complexities and dependencies of sound color production, it would be next to impossible to find a heuristic that could measure a player's embouchure pressure based solely on a trivial analysis of the acoustic signal. For example, a general measure of brightness (which is related to embouchure pressure) can be derived by finding the "center of mass" of the spectra, or sometimes by finding the ratio of the first to the second harmonics. These measurements, however, are much too correlated with the many other variables that make up a clarinet's sound color. Therefore, finding the contribution to brightness from embouchure by this heuristic is fruitless.

Instead, a Principle Components Analysis is used. The features chosen for the classifier are simply the energies associated with the first 10 harmonics of the clarinet tone. These are measured by taking a 256 point FFT, and using the pitch information (whose derivation is described in Section 4.3.2) as a pointer to where the peaks in the Fourier

transform should be. For example, a pitch of 300 Hz, given a 11025 Hz sampling rate and a 256 point FFT indicates that the fundamental frequency lies somewhere around bin 7 of the FFT. The second harmonic has energy somewhere around bin 14 and so on (see Figure 4.5). Because of the frequency spread that occurs in an FFT when the period of the measured signal is not an integer multiple of the window length, energy is summed for a few bins around where the harmonic is expected. For high pitches, where the sampling frequency limits the detection of energy in the upper harmonics (because of the Nyquist sampling theorem), these features are assumed equal to zero.

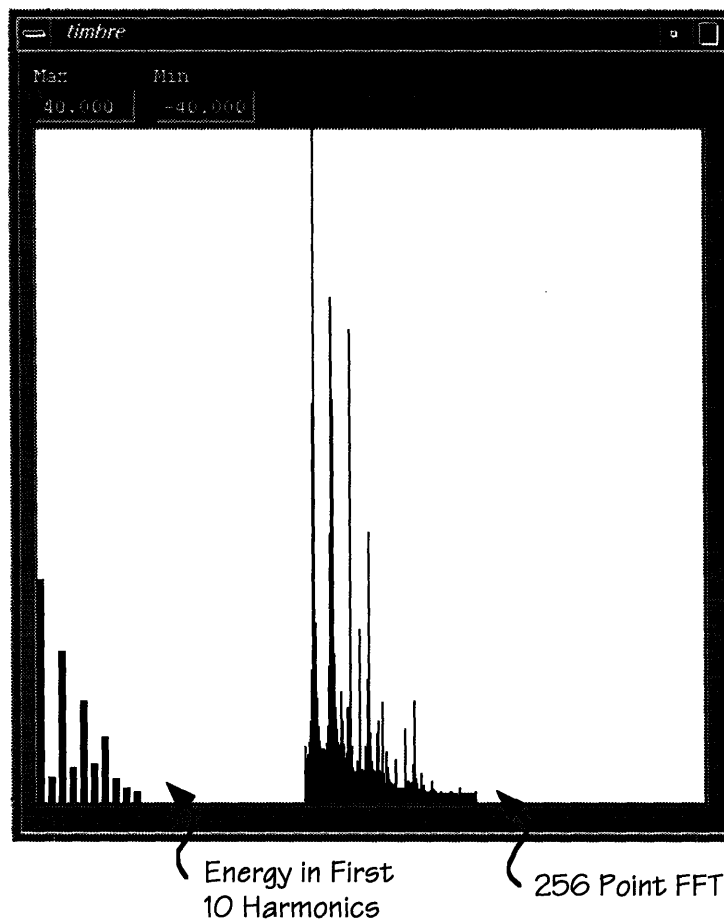


Figure 4.5. Converting the FFT into a feature vector. The 128 point vector from the 256 point FFT is reduced to a 10 dimensional vector. The detected pitch is used as an index into the FFT. The energies of each of the FFT peaks become the new feature vector.

Our goal is the estimation of a parameter that varies with a player's embouchure pressure. Because the relation of embouchure pressure to the 10 features is slightly different for each note of the clarinet, and certainly different for different registers in the

clarinet, it constitutes a non-linear relationship to note value. A classifier that is trained on different examples of embouchure pressure spanning the whole pitch range of the clarinet will fail. To alleviate this problem, many instances of the PCA classifier are used, one per note¹⁰. For each note of the clarinet, the player trains the system with about 20 examples of loose embouchure pressure and about 20 examples of tight embouchure pressure, each example played at a different volume. The eigenvalue decomposition is performed and two distinct clusters form, each belonging to one of the two trained classes. It is usually sufficient to use only the top 2 eigenvectors as the basis for these features ($M = 2$).

During the classification stage (i.e., after training), the pitch of the note is detected, the correct classifier is selected, and the features of the note are projected onto the space. The Mahalanobis distance (see Appendix A.1) is calculated from the new point to each of the class means, and a classification can be made. Furthermore, by subtracting the two distances, a continuous parameter of embouchure pressure is extracted (see Figure 4.6).

The advantage of using this method is that any player can train the system to detect his or her particular brand of embouchure pressure. The type of instrument and playing style are all accounted for by the classification scheme. The disadvantage is that training time can often be long and somewhat tedious.

¹⁰ This is perhaps a bit excessive, though. In reality, a single classifier can probably work well on a small cluster of notes that have similar spectral characteristics. This can reduce training time.

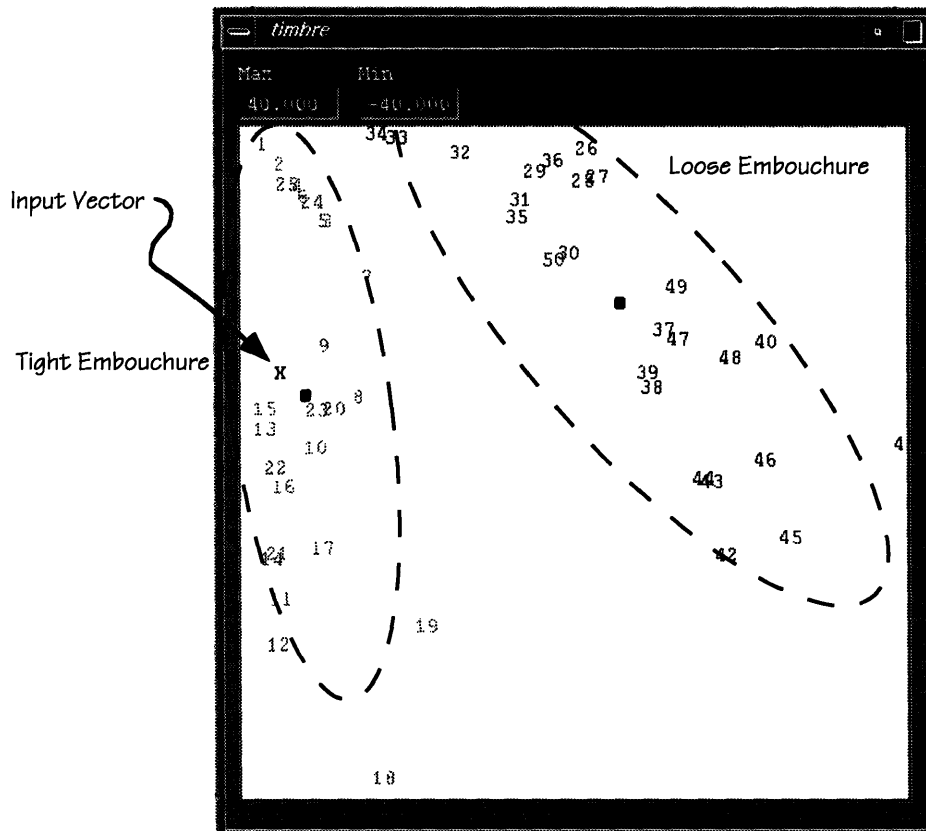


Figure 4.6. A classification of two different embouchure pressures. The plotted numbers correspond to the training set. The dark square in the center of each class is the class average. Covariances are indicated by the ellipses. The input vector is plotted against the training vectors in real-time. In this case, the input is classified as a tight embouchure.

4.3.4. Attacks

The attack portion of a note has been somewhat trivialized by its superficial treatment in the MIDI specification. Note decays have no specification at all. In certain instruments, this may be justified. A pianist has little control over how a note may begin. The note is struck with a certain speed, at which point the only artistic decision to be made is when to release the note. Such is not the case with the clarinet. In the clarinet, there are times when a note's onset is as clear as hitting a note on the piano, though this is seldom the case. Notes may be articulated with the tongue, or with a sharp burst of air pressure. In stop-tonguing, a reed comes to full rest before vibrating again at a new note, whereas in most regular tonguing, the reed's vibration is never fully squelched. In some cases, a note may "rise from nothing," with no discernible attack whatsoever. The distinction between note decays and note onsets is sometimes blurred; it is not always

clear if a rapid drop and rise in volume constitutes a new note attack or simply a quick change of dynamics. Jazz musicians use a full spectrum of different attack types, ranging from ghosting certain notes, to slap tonguing, to flutter tonguing.

The clarinet analysis system classifies attacks using a PCA, similar to the one used for embouchure pressure, though the input features are different. The important aspect of identifying different attack types in the clarinet is the choice of features which captures both the temporal and spectral evolution of the clarinet note as it is being started. However, before a set of features can be extracted, attacks must be located by finding *attack identification points* in the sound waveform.

An attack identification point is a property of the waveform that is generally characteristic of a new attack and is easily computed (since finding these points is a continuous process). There are several situations which constitute attacks in this system:

- Fast attack from silence. In this situation, the energy of the signal begins at its minimum noise level. The energy of the waveform must rise to be greater than a threshold value I_{fa} , and the derivative of the waveform must be greater than a threshold value dI_{fa} . The derivative is approximated by a linear regression fit over the past n energy samples (n is typically between 3 and 6).
- Slow attack from silence. If, by the time I_{fa} is reached, the derivative is less than dI_{fa} , the attack is considered a slow attack and receives special treatment (see below).
- Attack during sustain. When a note is reattacked normally (not stop-tongued), the energy level usually drops, but not below I_{fa} . A rearticulation attack is flagged when the derivative becomes sharply negative (below $-dI_{ra}$) and then rises positive (above dI_{ra}). dI_{ra} is typically a lower threshold than dI_{fa} .

Once an attack is identified, the following properties of the attack are marked:

- Attack beginning: The lowest energy point of the attack.

- Attack peak: The highest energy point of the attack.
- Attack ending: m energy samples after the attack (m is typically between 5 and 15).

Figure 4.7 illustrates these attack identification points. The choice of all the values that can be adjusted (i.e., thresholds, sample lengths) is somewhat user-dependent and also system-setup dependent.¹¹

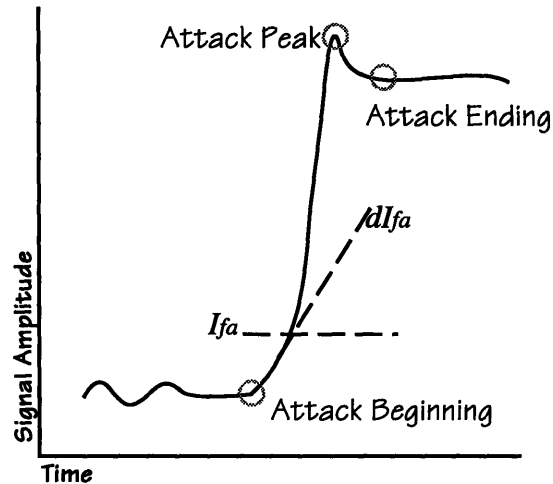


Figure 4.7. Attack Identification Points. These points are used as landmarks when identifying an attack. The feature vector for an attack is derived from the location of these points in the waveform.

The features used for the PCA are the following, arranged in a single feature vector:

- Two samples, one at the attack beginning, and one at the attack peak, of the energies in each of the first 10 harmonics of the note onset. These are found exactly in the same manner as in the embouchure pressure PCA.
- An energy contour of m samples between the attack peak and the attack ending.
- The attack speed, measured as a linear regression fit of the points between the attack beginning and the attack peak.

¹¹ For example, variables such as room noise, gain properties of the microphone pre-amp, and exact microphone placement.

The system is trained with examples of attacks played at different notes and at different velocities. Similar to the analysis of embouchure pressure, there are several instances of classifiers, each identified with a small range of pitches.

The clarinet analysis system was trained to distinguish between four types of attacks: soft (no tongue) attack, stop-tongue attack, continuous attack (the reed keeps vibrating), and slap tongue (see Figure 4.8). Slow attacks are not classified because they do not conform nicely to the attack identification points. A slap tongue is characterized by a large increase in higher harmonics during the evolution of the attack and has a twangy sound. Continuous attacks show almost no change in harmonic content.

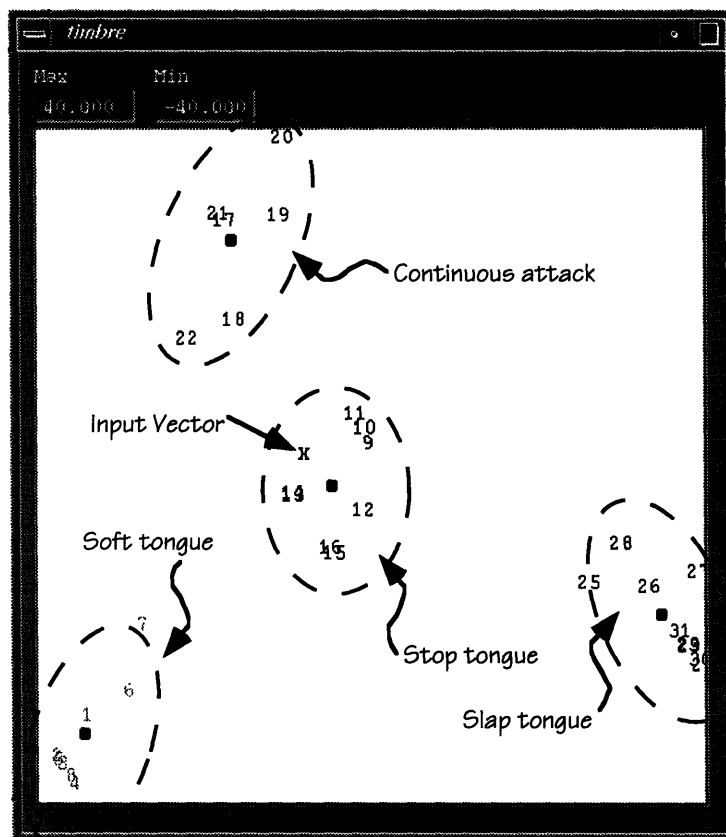


Figure 4.8. A classification of four different types of attacks. Similarly to the embouchure pressure classification, this plot depicts a training set of different attacks. The input vector in this case is classified as a stop-tongue.

4.3.5. Vibrato

Vibrato refers to the slow modulation of pitch and/or amplitude of a sustained note. It is a common means of expression in stringed instruments and most winds. String

instruments use vibrato to enhance the liveliness of an otherwise duller tone. In classical styles of clarinet playing, vibrato is used sparingly (as opposed to flute and oboe), as a means of emphasis. In addition, most string instrument vibrato is pitch vibrato, while wind vibrato is often a mix of both pitch and amplitude modulation. In the clarinet (at least in the author's style of playing), amplitude vibrato is most common and is the candidate for system analysis.

In amplitude modulated vibrato, the energy signal fluctuates almost like a perfect sinusoid¹². The amplitude and frequency of this sinusoid are measured to yield a quantitative sense of the intensity of the vibrato. First, a frequency range of the vibrato is calculated (which is entirely player dependent, but tends to range between 1 Hz and 5 Hz). Then, the energy is filtered by a second order IIR bandpass filter with cutoff frequencies at the bounds of the player's vibrato frequencies. Maxima and minima detectors are alternately passed over the signal which identify the amplitude and frequency of the vibrato.

The bandpass filter is useful because it decreases the vibrato detector's sensitivity to noise and long term changes in dynamics. In addition, to avoid a false registration of vibrato during attacks and decays, the vibrato detector is turned off when attack identification points are discovered (see Section 4.3.4).

4.3.6. Using the System

The software package that groups all of these functions together has a graphical user interface that both facilitated the development of the analysis algorithms, and allows a user to inspect the working behavior of the system with sound input. The GUI (see Figure 4.9) has a control panel consisting of buttons, pop-up menus, sliders, text dialog boxes, and a graphics output display for graphically representing two dimensional plots such

¹² This is the measured result of my vibrato. I assume that other instrumental vibrato is similar.

as the energy waveform, pitch contours, and the bases generated by the PCA. Figures 4.4 through 4.6 show actual examples of these plots.

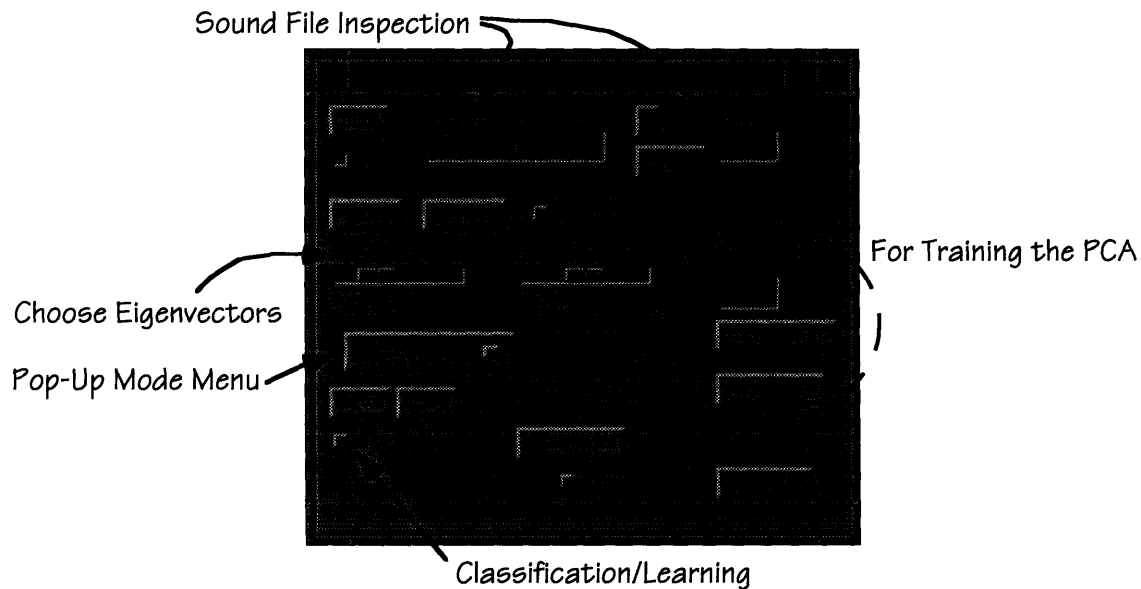


Figure 4.9. The Graphical User Interface.

The software package has the option of running in real-time, reading sound bytes from the audio port, or in non-real-time, by reading bytes from a sound file¹³.

In the sound-color training process, the user plays a series of tones of a particular sound color, or in a particular playing manner (such as with a given embouchure pressure), and labels the members of this training set accordingly. First a series of tones produced with a tight embouchure is presented to the system. The idea is to vary everything about these tones except for embouchure tightness. Tones are played at different volumes, and at varying pitches. Then, a second series of loose embouchure tones is played in a similarly manner. After training, the PCA is automatically computed and classification can begin.

During the classification process, a plot of the training data is displayed in the new basis (the user can choose which two eigenvectors should constitute the basis), and the new, unresolved sample is plotted against the training set (see for example, Figure 4.6). This plot is continuously being updated with a new tone sample about 70 times per

¹³ This was especially useful for debugging and tuning the analysis methods.

second. The user can see a representation of the input tone color meandering through this two dimensional space in real-time. In this way, her or she can confirm the training process and may opt to modify the training set in some way (say, by adding more training vectors or removing any training mistakes). In addition, it is not necessary to limit this classifier to just two classes of embouchure pressure. Instrumentalist who feel that they can accurately reproduce additional sound color dependent gestures may train the system on those as well.

The note-onset classifier works in a similar fashion. The user trains the system with different types of attacks, played at different volumes and with different pitches. Groups of the same attack type are labeled as such, and the PCA is computed. In the classification phase, a user may inspect the success of the PCA by playing different types of attacks. For each attack played, the graphical display will be updated with a two dimensional plot of the training set and the location of the new attack.

At the end of the training process, the training data can be saved to disk and loaded at a future session. In addition to the PCA plot, the user may also display a scrolling graph of other system parameters such as the signal waveform, the pitch, vibrato measurements, and the attack identification points (discussed in Section 4.3.4).

During normal operation, a constant parameter stream is delivered through the MIDI output port. Energy information, tone color information, and vibrato information is constantly being updated. Attack onset information and pitch information are only sent when necessary.

4.4. Towards a Higher Level of Description

Analysis of the gesture parameters discussed above constitutes the first step in the sizable goal of describing the full range of nuances of a clarinetist's performance. They do not pretend to span the whole range of possible clarinet gestures, but rather give a good sampling of the areas that one would want to look at when describing a single-line instrumental performance. At this juncture, some issues arise that have to do with the "grandeur purpose" of this type of musical gesture analysis.

4.4.1. A Comprehensive Analysis

A comprehensive description of a clarinetist's gestures would provide a great amount of detail of everything that was going on in the clarinetist's performance. From an information theory point of view, the set of parameters gathered would have as many degrees of freedom as the system being measured (i.e., the performer). From the analysis by synthesis point of view described by Risset, a comprehensive gestural description would allow a synthesis engine of suitable complexity to mimic the clarinet performance with errors imperceptible to the human ear.

However, such a detailed analysis may not be what we want from this system. For one thing, testing such a criteria would be a very difficult task. It would require not only an uncanny set of analysis tools, but also a very sophisticated synthesis engine which could synthesize the clarinet to perfection, and furthermore, be exquisitely matched to the parameter description language of the analysis tool. A more realistic, and perhaps more interesting goal of the clarinet analysis system, would be to provide higher level descriptions of the performance. These descriptors would say something about a performance's expressive nature at a level closer to how we, as listeners, consider expression in music.

4.4.2. Adjective Transforms

The Hyperinstrument group at the Media Lab developed an interactive percussion system (DrumBoy), which addresses the problem of generating interesting drum patterns on different levels of user control and user description. At the lowest level, a user can specify every single drum hit, much as he or she would do with a sequencer. At the highest level, *adjective transforms* are employed to modify existing drum patterns in genre-independent ways. A particular drum pattern can be made more mechanical, more floating, or more complex (to name a few), by a simple strike of a key.

The thesis of Drum-Boy is that through these transformers, it becomes possible to manipulate a complex interactive system by abstracting away details and simplifying controls. This idea can be applied to the clarinet analysis system (though in the opposite

direction). A set of adjective descriptors can be defined based on a time-evolving combination of the low level parameters described in Section 4.3. These descriptors would hide the details of the low level parameters and provide more meaningful information about instrumental performance.

When we listen to an instrumental performance, we often associate certain descriptive or emotional states with the shape of the musical line that we hear. A performer will often imagine certain scenarios or attempt to arouse an emotional response to better convey an expressive gesture. Even though attempting to label or describe the scope of these responses and emotions is beyond the scope of this thesis and delves into realms of cognitive models and Artificial Intelligence, there exists a level of expression description in music that may serve our purpose.

Composers generally notate music by indicating pitch, rhythmic, phrasing, and dynamic information. At times, they will want to give expressive instructions to musicians, but in order to keep notational compactness, use only a handful of expressive adjectives. For example, some of these are: *dolce* (sweetly), *marcato* (sharply accented), *sotto voce* (with a soft undertone), *cantabile* (singing), and *maestoso* (in a dignified manner). Trained musicians instinctively change their playing style at these indications. It is at this level that higher level descriptors may be constructed from the low level gesture parameters.

For example, playing *dolce* on the clarinet implies soft dynamics, smooth and soft attacks, occasional vibrato, and a fairly bright tone (achieved by a tighter embouchure). *Sotto voce*, on the other hand, uses a looser embouchure for a slightly breathier, duller tone, and hardly any vibrato. *Marcato* is characterized by precise, sharp attacks played loudly, evenly, and with a bright tone. Defining these expression adjectives, therefore, is a process of combining certain gesture parameters and considering their evolution over time. For example, “smoothness” of a volume can be measured by high-pass filtering the energy signal and looking for a threshold. However, analyzing the time evolution of some parameters is not always a simple matter. If we wish to measure the rhythmic precision of attacks, we require a tool which collects attack events and performs a rhythmic analysis of them. This alone can be the subject of a thesis (see for example [Ros92]).

To come up with anything but very simple adjective-type descriptors which simply combine low level gesture parameters requires musical knowledge and sophisticated representations of musical expression that are in early stages of development. Today, this still remain an elusive problem.

4.5. Demonstration

I composed a simple multi-layered, multi-timbral looping sequence to demonstrate the concepts presented above. The demonstration is written in Hyperlisp ([Chu91]), a MIDI-enhanced version of Macintosh Common Lisp. The central idea is to let a performer improvise over an ostinato line which changes its character and shape in response to the analyzed performance parameters. Each layer can potentially have a different volume, accent structure, and note duration. Several of the layers are hockets — two 16th note layers, each interspersed with rests, that interlock together to form a complete line. Some of the layers may be time-shifted with respect to one another to form a close canon or echo effect. The timbre of some lines can be modified by mixing in different patches available from the synthesizer. In addition, harmonic aspects of the layers can change (such as the dissonance of the line).

I first trained the clarinet timbre analysis system with different gesture examples (as described in Section 4.3.6), and then connected it to the Hyperlisp demonstration system. The mapping between the analyzed parameters and the looping sequence can essentially take on any relationship, though I created a few modes which either supported or contradicted the performer's gestures.

In one mode, I tried having the system respond to a clarinetist playing *dolce*. Under *non-dolce* conditions, the ostinato plays very short notes. Only one of the hocket sequences is playing, and no layers are time-shifted. Harmonically, the sequence sounds "open." With an increase in vibrato, the sequence durations increase. A brighter tone alters the timbre of the sequence and introduces thicker harmony. Playing softer causes a time-shifting and brings in both hocket layers. The overall result is a texture whose smoothness and richness increases with a sweeter clarinet playing style.

A video demonstration of the clarinet timbre analysis system is available from the Media Laboratory. See Appendix A.2.

4.6. Evaluation

4.6.1. Other Approaches and Real-Time Issues

From its onset, the implementation goal of this system has been that it run in real-time on a modern workstation. Today's workstations have a good balance between performance and the comfort of their work environment. Still, the want of a real-time system ruled out a few approaches to the analysis problem.

One such approach is an analysis based on physical modeling. A mathematical model of the clarinet is constructed which faithfully reproduces the clarinet sound. The reed and mouthpiece can be modeled with a non-linear energy source and the bore, as a waveguide with T networks for finger holes [VKL93]. If the mouthpiece model is sufficiently complex, it can account for air velocity and embouchure pressure. Once a good model is constructed, it must be *inverted* — the input of an acoustic signal will output values for the model parameters. Inverting a physical model is not a trivial task. Both running the model and inverting it are extremely costly operations. Such a scheme, though appealing¹⁴, cannot work in real-time.

To run in real-time the PCA must get feature vectors from the input stream quickly. In the embouchure pressure classification, these features are the strength of a tone's harmonics. I began looking for these harmonics by building a non-linear synthesis model whose parameters consisted of a fundamental frequency and amplitudes of harmonics. Thus, the model could function both as a pitch tracker and as a spectrum analyzer. I inverted this model with an extended Kalman Filter [BH92]. Unfortunately, even this scheme failed to run in real-time.

¹⁴ The appeal is that a physical modeling paradigm comes from first principles. It attacks the problem at its core. The method used here in some sense circumvents the source of the problem (i.e., the physical behavior of the clarinet) and just deals with its perceptual outcome.

4.6.2. Why PCA?

The PCA approach is appealing in several ways. First, PCA easily runs in real-time. All of its compute-intensive operations (i.e., the training process) happen during the learning stage. During run-time, classifications occur with imperceptible delay.

Secondly, PCA frees the user from having to overtly define which properties of the clarinet signal the system should look for during classification. This makes the system user-independent because the PCA takes into account the slight differences between different users' playing styles. Each user can train the system with his or her favorite attack types and tone types.

Thirdly, using a PCA draws a clear boundary between the classification algorithm (i.e., all the math for eigenvalue decomposition, etc.) and the feature extraction process. This is reflected in the system's implementation — the classification scheme is abstracted as a simple input/output "black box." Features are simply injected into this "box" which then produces a classification. Thus, creating an instance of a classifier is simply a matter of constructing an appropriate feature vector. The timbre analysis system has several signal processing tools (see Figure 4.3) which can be easily manipulated to build a feature set. Building the embouchure pressure and attack analyzers was facilitated by this programming paradigm.

4.6.3. Possible Improvements

Considering all the pitfalls of trying to extract a reliable timbre stream from the clarinet, this system performed quite well. There are, however, a number of improvements that could be made in future versions.

Two problems surfaced by using an external instrument microphone. First, the system was somewhat sensitive to loud external noise. For example, the volume of the demonstration sequence could not be turned up too loud, since the sound it generated would interfere with the signal analysis. At comfortable listening levels, it behaved normally. But, to completely alleviate this problem, a contact microphone could be

attached to the clarinet's bore. This microphone would only couple into the vibrations of the instrument body and would not amplify any ambient room noise.

Second, the timbral analysis worked most reliably in pitch regions where the mike was close to open tone holes. Tones produced in parts of bore far away from the microphone were more susceptible to distortion from room reverberation. In a future version, several microphones mounted on different locations of the clarinet would yield a more consistent analysis. Of course, external noise would still be a problem in this approach.

Note decay is an important aspect of the clarinet which, unfortunately, was not fully implemented. Doing so would only require another instance of a PCA, and the definition of a suitable feature vector. Identification points can be constructed for decays, just as they were for attacks. Other clarinetistic gestures, such as growling and multiphonic effects, should be measured in future versions as well.

5. Related Issues

5.1. Generality

5.1.1. Other Instruments

Even though this system was tuned specifically for the clarinet, there is nothing about it which limits these types of analyses to the clarinet alone. For example, the pitch detector expects to find pitches within the possible range of notes on the clarinet (D2 through B5), though this range may be set in the algorithm. A flute pitch tracker would obviously require a different expected range. Instruments with a vastly different range (say a contrabassoon or a piccolo) would certainly require a different length pitch analysis window, and possibly a different sampling rate.

Similarly, the current vibrato measurement takes only amplitude vibrato into account. An oboist who wishes to analyze pitch vibrato as well would need to apply the vibrato detector to the pitch contour as well as the energy contour.

The Principle Component Analysis (for both tone color and attack detection) subsystem requires a set of input features derived from the sound stream. These were optimized with the clarinet in mind, but the system was designed with some modularity in mind. Because the PCA is independent of any specific classification problem, it can easily operate on any set of feature vectors. If needed, a different set of feature vectors can be optimization for other instrument.

An important limitation of the system, however, is the requirement that the sound source be monophonic. This analysis does not apply to the piano or to ensembles of monophonic instruments (except if a different instance of the analysis system is applied to each member of the group, with no interference from any other member). String instruments would have to avoid playing double-stops.

5.1.2. The Voice

The study of voice synthesis and analysis is a vast topic in and of itself. In most cases, the voice is studied within the context of speech analysis and coding, speech synthesis, and speaker identification [OSh87]. Some recent work has looked into the expressiveness of speech in terms of low level voice synthesis parameters such as pitch inflection, and timing information [Cah90]. Since the voice is often used in a musical context, and can be thought of as a single-line monophonic instrument, it becomes a candidate for the analysis methods and issues discussed here.

The human voice is perhaps the most expressive musical instrument we have. Even if one disregards all meaning delivered through language, the communicative powers of the singing voice can relay much more expression than traditional instruments. In some of the more contemporary classical music literature (like John Deak's), stringed instruments are instructed to imitate aspects of human voice such as laughing or crying.

Using the human voice as a control stream to shape a generative musical system is a relatively unexplored area in computer music. Perhaps the tools to perform an adequate analysis of the singing voice have been absent. The analysis methods described in this thesis, though far from being complete, may constitute the beginnings of a way to tap into this very expressive musical stream.

To apply the analysis system to the human voice, some significant modifications would need to take place. The system would need to have a representation for fricatives and be able to characterize them in some way (perhaps by using a pattern recognition approach). Vowel distinction, which is essentially a discrimination in the sound color of pitched utterances can be achieved by the PCA, though its input feature vector should probably incorporate the formant of the voice. Pitch tracking can essentially stay the same, though the notion of distinct notes within a scale makes less sense since sung notes often merge from one to the next without a definite boundary as is the case with wind instruments.

In any case, the idea of a continuous parameter stream which provides information about the music-gestural qualities of the human voice (as opposed to speech qualities) is still valid, and higher level descriptors may be gleaned from these lower level parameters.

5.2. An Alternative to MIDI

One of the implications of gesture parameter extraction from a clarinet performance is the need for a sophisticated communication interface between computers, performers, and synthesizers. The limitations of MIDI become apparent when attempting to describe performance data that did not originate from a keyboard instrument. Even though protocols can be built on top of MIDI by using unassigned controller values to represent specialized parameters (such as tone color), MIDI has severe limitations in bandwidth and scope [Moo88]. Furthermore, because the MIDI standard has been so embedded into the mainstream electronic music community, it has probably stifled the growth of interesting electronic music applications because of these limitations.

For example, a note on a clarinet does not necessarily have a definite attack point. It may slowly increase in volume from nothing. MIDI can send volume change messages, but then the attack velocity information becomes meaningless (it normally corresponds to the loudness of the note being played). For stringed instruments, the situation is worse. A violinist can change notes by quickly “pitch bending” from one note to the next (i.e., by sliding the finger on the fingerboard). Representing such a gesture in MIDI is uncomfortable because, even though pitch bend is available, it is always associated with a particular note (given by a “note-on”) which the violinist may have slid away from. MIDI completely fails as a language by which to represent the singing voice.

A protocol replacing MIDI has been recently developed in a collaboration between Zeta Music and CNMAT at Berkeley called the *ZIFI Music Parameter Description Language (MPDL)* [MWW94]. MPDL remedies many of the problems facing MIDI users, especially in the face of the emerging need to accurately describe non-keyboard musical events. It is an ideal language to represent performance details such as the ones presented here, taking into account note attack and decay descriptors, timbral modifications, parameter

modulation (such as vibrato), and other issues not addressed here such as sound spatialization. On a broader level, MPDL supports hierarchical grouping where individual notes may be grouped into instruments which may in turn be grouped into families of instruments. MPDL commands may be issued at any one of these levels, thereby affecting anything from minute details in a single instrument to parameters which influence groups of instruments.

6. Future Work and Conclusion

The work described here is at times very specific — on an implementation level, it deals only with the interpretation of the acoustic properties of the clarinet. However, on a broader scope, I hope it motivates the need for further investigation of expression-rich musical streams. I will only touch on some of these possibilities here.

6.1. Possible Future Directions

6.1.1. New Synthesis Technology

The Yamaha Corporation recently released a new waveguide synthesizer, the VL1, which is fundamentally different from most synthesizers we have seen to date. It uses pseudo-physical modeling algorithms as opposed to traditional synthesis methods (FM, sampling, wave table). To control the production of sound, the synthesizer must be provided with a number of parameters that affect the model's behavior. For example, these synthesized instruments respond to breath pressure, embouchure pressure, vibrato, tonguing, breath noise, and growl parameters. Controlling all these parameters, however, is somewhat tedious, and certainly feels a bit awkward with the controllers provided (foot pedals, mod. wheels).¹⁵ The information stream derived from the clarinet timbre analysis system (perhaps in conjunction with MPDL) could function as part of a controlling mechanism for this new synthesis technology.

6.1.2. Ensembles of Instruments

All Hyperinstruments to date have been concerned with gesture interpretation of a single performer¹⁶. In the future, more emphasis will be given to paradigms of multi-performer interactive experiences where the gestures measured from each member of an ensemble

¹⁵ The breath controller is somewhat more helpful.

¹⁶ Plans are currently underway for developing a Hyperstring quartet in collaboration with the Kronos Quartet.

are used to guide the total musical outcome. In a chamber music scenario, musicians will be challenged by a previously unheard of way of shaping a composition — active communicative influence between members of the group, where gestures from one player may directly control another players instrument. This situation points to an urgent need for careful timbral measurements of instruments and introduces a new topic of research in understanding how to effectively combine such measurements from multiple musical streams.

6.1.3. A Musical Expression Language

The exploration of higher level musical expression parameters will require methods by which systems may communicate about these parameters. MIDI and MPDL service the interface needs of machines and synthesizers on a note by note level but have no mechanism for describing musical events at the level of phrasing or intent. Recent work [Rig94] begins to address some of these issues. With the advent of systems that extract details and nuance from an instrumental performance (obviously consuming a larger bandwidth than MIDI) comes the need to step back from these streams of parameters and describe events at a more human-understandable level. A formalized language of musical expression based in a serious cognitive, knowledge-intensive interpretation system, could revolutionize the way in which our machines communicate with each other and with us about music.

6.1.4. Teaching Tools

During the course of my lessons, my clarinet teacher, William Wrzecien, will frequently reiterate some of the same simple instructions over and over during occasional follies in my playing. Many of these comments have to do with some fairly low-level aspects of playing practice such as consistency of attacks, clarity of tone, and intonation. With no offense intended towards Mr. Wrzecien, on occasion I wish for a computerized teaching tool that could indicate and correct my performance blunders during practice. The realization of such a tool is not that far away. Researchers at the Media Lab are currently developing a teaching system for the piano which listens to a student's performance through MIDI and gives feedback about interpretation and expression. A

similar system can be constructed for other acoustic instruments given the timbral analysis techniques presented here. A series of teaching systems can be constructed, each tuned for a specific instrument, which will improve the practice of amateur performers.

6.2. Concluding Remarks

The art of interfacing acoustic instruments to computers is at a turning point. Computers are becoming fast enough to perform sophisticated analysis in real-time and the electronic music community is becoming frustrated with the limitations imposed by the protocols of today. We understand more about the timbre of instruments than we did around 20 years ago and, from a physical modeling point of view, can do a better job of mimicking the nuances of these instruments. Being both a computer scientist and a musician, I am excited to be in the midst of these developments. They will, without a doubt, shape the way we think about and experience the interplay of computer and musician.

Appendix

A.1. Principle Component Analysis

Principle Component Analysis begins with a training set of feature vectors. Each vector is associated *a priori* with a class which represents a salient property. In general, there is no limit to the number of allowable classes, though the classifier should be trained with a reasonably sized sample of instances from each class to ensure statistical significance. The exact sample size depends on the variance of the sample data. A larger class variance requires more samples. A set of N features is grouped into a single vector of dimension N (the *feature vector*). It is a constraint of the analysis method that all feature vectors, regardless of what class they are associated with, must be of the same dimension.

We start out with a set of M training feature vectors $\Gamma_i, \{i = 1 \dots M\}$, arranged as column vectors, where each vector is associated with one of P *a priori* defined classes $R_k, \{k = 1 \dots P\}$. We now need to find the covariance between every feature vector in our training set. Ψ is the sample mean of all the feature vectors:

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i$$

The covariance matrix K is $N \times N$ and describes the dependent and independent variations between each feature dimension across the whole training set:

$$K = \frac{M}{M-1} \sum_{n=1}^M \sum_{m=1}^M (\Gamma_n - \Psi)(\Gamma_m - \Psi)^T$$

If we redefine our features to be zero mean so that each vector is $\Phi_i = \Gamma_i - \Psi$, and group the collection of feature vectors in the matrix $A = [\Phi_1 \Phi_2 \dots \Phi_M]$, the covariance matrix is simply:

$$K = \frac{M}{M-1} AA^T$$

Assuming that the training set is statistically distributed as Gaussian and that all the sample vectors are jointly Gaussian, the probability distribution for the features of the different classes is defined by the multivariate Normal distribution for an N -dimensional random vector Γ .

$$p_{\Gamma}(\Gamma) = \frac{1}{(2\pi)^{N/2} |K|^{1/2}} e^{[-\frac{1}{2}(\Gamma-\Psi)K^{-1}(\Gamma-\Psi)]}$$

Normally, the covariance matrix K is highly correlated. Finding the eigenvectors of K will lead to a set of vectors u which are all normal and uncorrelated. These vectors form an orthogonal basis which can describe the original feature vectors by weighted linear combinations of these eigenvectors. However, if M is less than N , only an M th dimensional subspace of these eigenvectors will be necessary to fully span the set of training feature vectors. If N is dramatically larger than M , as is the case with eigenfaces (cite[]), a mathematical simplification can be used which only solves for the significant M eigenvectors of K by finding the eigenvectors of the $M \times M$ matrix $A^T A$.

At any rate, if all the eigenvectors are arranged in the matrix $V = [u_1 u_2 \dots u_N]$, a new matrix $\Lambda = VKV^T$ is the diagonal *eigenvalue matrix* of K . Λ has the important property of being an uncorrelated covariance matrix for the variable $\Gamma' = V^T \Gamma$.

This new variable is nothing more than Γ after a linear rotation of coordinates. Again, assuming that Γ' is Gaussian, its probability distribution can be written as above, except that in the new coordinate system, all its variables are independent. Thus, they may be written as a product of N single dimensional distribution functions:

$$p_{\Gamma'}(\Gamma') = \prod_{i=1}^N \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left[-\frac{1}{2} \frac{(\Gamma'_i - m_{\Gamma'_i})^2}{\lambda_i}\right]$$

where λ_i are the diagonal elements of Λ .

In the new basis V , any feature vector, whether part of the original training set or not, can be represented as a unique linear combination of the vectors in V . In other words, for every $\Phi = \Gamma - \Psi$,

$$\Phi = \sum_{j=1}^N \omega_j \mathbf{u}_j$$

Finding the appropriate weights ω_j involves nothing more than projecting Φ on the corresponding eigenvectors in V :

$$\omega_j = \mathbf{u}_j^T \Phi$$

From the total set of eigenvectors in V , we now choose a subset of M' most significant eigenvectors as a basis by which to represent any feature vector. In most cases, this subset consists of the M' vectors whose corresponding eigenvalues are the largest. In most practical problems, M' is significantly lower than either M or N , which means that most of the information contained in the N dimensional feature vector has been boiled down to only M' dimensions. It is in this reduced basis that a decision is made about how a new unknown observation compares with the set of training observations.

Each N dimensional feature vector Γ is represented as a new vector $\Omega^T = [\omega_1 \ \omega_2 \ \dots \ \omega_{M'}]$. Before attempting to classify an unknown observation, statistics are gathered about the *a priori* defined classes. Presumably, each of the P classes has a reasonable number of example observations that would ensure statistics with a high degree of confidence. The M' weights are computed for each feature vector per class as described above to yield vectors Ω_q^p where p ranges from 1 to P (the total number of classes), and q (ranging from 1 to Q^p) is an index into a particular feature vector in a class p . The first and second order statistics are found for each of the P classes:

$$\bar{\Omega}^p = \frac{1}{Q^p} \sum_{q=1}^{Q^p} \Omega_q^p$$

and

$$K^p = \frac{1}{Q^p - 1} W^p W^{pT}$$

where $W^p = [(\Omega_1^p - \bar{\Omega}^p) \ (\Omega_2^p - \bar{\Omega}^p) \ \dots \ (\Omega_{Q^p}^p - \bar{\Omega}^p)]$.

$\bar{\Omega}^p$ is the class average vector and K^p is the class covariance matrix.

When an unknown observation is presented to the classifier, we first transform its feature vector Γ , into the M -space representation to get a vector $\Omega^?$, and then find the distance (i.e., error measurement) between $\Omega^?$ and each of the class averages $\bar{\Omega}^p$. The observation is labeled as belonging to class which has the lowest corresponding error measurement. The distance criteria can be a simple Euclidean distance:

$$\varepsilon_p = (\Omega^? - \bar{\Omega}^p)^T (\Omega^? - \bar{\Omega}^p),$$

but a better measurement, which takes into account the variance along each dimension is the Mahalanobis distance:

$$d_p = (\Omega^? - \bar{\Omega}^p)^T K^{p-1} (\Omega^? - \bar{\Omega}^p).$$

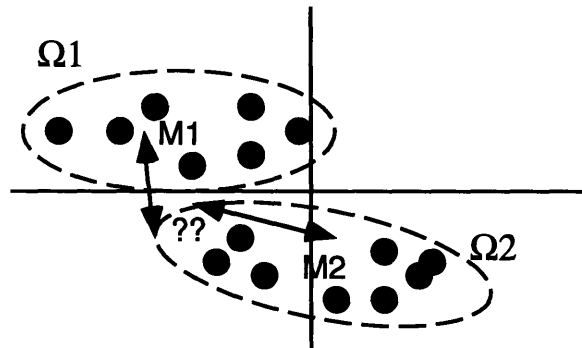


Figure A.1. A projection of an unknown observation vector ?? on the eigenspace. The Euclidean distance between the input vector and the class average of Ω_1 (M1) is smaller than between ?? and M2. However, ?? is really a member of Ω_2 . The Mahalanobis distance takes the covariance matrices into account (depicted by the ellipses) and classifies ?? as part of Ω_2 .

Computation for this classification scheme is quite reasonable. Most of the heavy math can be done off-line. The major steps (i.e., computing the eigenvectors of the total covariance matrix K and inverting the class covariance matrices K^p) are done once as part of the training process. In real-time, the classification involves a few multiplication to find $\Omega^?$, and then a few more to find d_p .

A.2. Video of Demonstration

A video demonstration of the clarinet timbre analysis system, including the composed sequence described in Section 4.5 is available from:

Hyperinstrument Group
Room E15-496
MIT Media Laboratory
20 Ames St.
Cambridge, MA 02139
(617) 253-0392
Contact: Suzanne McDermott
alma@media.mit.edu

Bibliography

- [Bac74] Backus, J. "Input impedance curves for the reed woodwind instruments" *Journal of the Acoustical Society of America*. (56), 1266-1280, 1974.
- [Ben76] Benade, A. H. *Fundamentals of musical acoustics*. London: Oxford University Press, 1976. Also available from New York: Dover Publications, Inc., 1990.
- [BH92] Brown, R. G., and Hwang, P. Y.C. *Introduction to Random Signals and Applied Kalman Filtering*: New York, NY: John Wiley & Sons, Inc., 1992.
- [BK88] Benade, A. H., and Kouzoupis, S. N. "The clarinet spectrum: Theory and experiment" *Journal of the Acoustical Society of America*. (83), 292-304, 1988.
- [BL85] Benade, A. H., and Larson, C. O. "Requirements and techniques for measuring the musical spectrum of the clarinet" *Journal of the Acoustical Society of America*. (78), 1475-1498, 1985.
- [BPB85] Barrière, J.-B., Potard Y., and Baisnée, P. F. "Models of continuity between synthesis and processing for the elaboration and control of timbre structure." *Proceedings of the 1985 International Computer Music Conference*: San Francisco, 193-198, 1985.
- [Cah90] Cahn, J. E. "The generation of affect in synthesized speech" *Journal of The American Voice I/O Society* (8), 1-19, 1990.
- [Cho73] Chowning, J. "The synthesis of complex audio spectra by means of frequency modulation." *Journal of the Audio Engineering Society* (21): 526-534, 1973.
- [Chu91] Chung, J. *Hyperlisp Reference Manual*, available from the MIT Media Laboratory, 1991.
- [DGR93] Depalle, Ph., Garcia, G., and Rodet, X. "Analysis of Sound for Additive Synthesis: Tracking of Partial Using Hidden Markov Models." *Proceedings of the 1993 International Computer Music Conference* : 94-97, Tokyo, Sept. 10-15, 1993.
- [DH73] Duda, R. O., and Hart, P. E. *Pattern Recognition and Scene Analysis*, New York, NY: John Wiley & Sons, Inc., 1973
- [DP89] De Poli, G., and Piccialli, A. "Waveform transformation techniques for sound synthesis." *Proceedings of the International Congress on Acoustics*, Belgrade, 1989.
- [Ell92] Ellis, D. P. W. *A Perceptual Representation of Audio*, Master's Thesis. Department of Electrical Engineering and Computer Science, MIT, 1992.
- [FC91] Florens, J.-L., and Cadoz, C. "The Physical Model: Modeling and Simulating the Instrumental Universe" in De Poli, G., Piccialli, A., and Roads, C. *Representations of Musical Signals* : Cambridge, Massachusetts: MIT Press, 1991.
- [Ger91] Gershenfeld, N. Sensors for real-time cello analysis and interpretation, *Proceedings of the 1991 International Computer Music Conference* : Montréal, 1991.

- [Ger92] Gershenfeld, N. "Information in Dynamics", Workshop on Physics and Computation, PhysComp: Dallas, Texas, October 2-4, 1992.
- [Gre75] Grey, J. M. *An Exploration of Musical Timbre*. Ph.D. Thesis, Stanford University, 1975. Distributed as Dept. of Music Report No. Stan-M-2.
- [Han89] Handel, S. *Listening*: Cambridge, Massachusetts: MIT Press, 1989.
- [Hel54] Helmholtz, H. L. F. "On the Sensations of Tone as a Physiological Basis for the Theory of Music" A. J. Ellis, trans. Dover, New York, 1954.
- [Hol89] Holschneider, M., et. al. "A real-time algorithm for signal analysis with the help of the wavelet transform." In J. Combes, A. Grossmann, and Tchamitchian, P., *Wavelets, Time-Frequency Methods and Phase Space*. New York: Springer-Verlag, 286-297, 1989.
- [Mac84] Machover, T. "Computer Music With and Without Instruments" *Contemporary Music Review*, 1(1):203-230, 1984.
- [Mac91] Machover, T. *Begin Again Again...* Musical Score. Milan: Ricordi Editions, 1991.
- [Mac92] Machover, T. *Hyperinstruments: A Progress Report*, available from the Media Laboratory, Massachusetts Institute of Technology, 1992.
- [Mat93] Matsumoto, F. *Using Simple Controls to Manipulate Complex Objects: Application to the Drum-Boy Interactive Percussion System*, Master's Thesis. MIT Media Laboratory, 1993.
- [MB79] McAdams, S. and Bregman, A. "Hearing Musical Streams" *Computer Music Journal*, 3(4):26-43, 1979.
- [Met95] Métois, E. *Determinism Versus Stochasticity*, Unpublished. Available from MIT Media Laboratory, 1995.
- [MG77] Moorer, J. A. and Grey, J. M. "Lexicon of Analyzed Tones" *Computer Music Journal*, 1(3):12-29, 1977.
- [Moo79] Moorer, J. A. "The use of linear prediction of speech in computer music application." *Journal of the Audio Engineering Society* (27) 134-140, 1979.
- [Moo88] Moore, F. R. "The Dysfunctions of MIDI." *Computer Music Journal* 12(1):19-28
- [Moo89] Moore, B. C. J. *An Introduction to the Psychology of Hearing* : London, Academic Press Ltd., 1989.
- [MSW85] McIntyre, M. E., Schumacher, R. T., and Woodhouse, J. "On the oscillations of musical instruments" *Journal of the Acoustical Society of America*. 78, 1475-1498, 1985.
- [MWW94] McMillen, K., Wessel, D. L., and Wright, M. "The ZUPI Music Parameter Description Language" *Computer Music Journal* 18:(4), 1994.
- [OS89] Oppenheim, A. V. and Shafer, R. W. *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1989.

- [OSh87] O'Shaughnessy, D. *Speech Communication* Reading, MA: Addison-Wesley Publishing Co., 1987.
- [Pre92] Press, W. H. et. al. *Numerical Recipes in C: The Art of Scientific Computing*: Cambridge: Cambridge University Press, 1992.
- [Rig94] Rigopoulos, A. *Growing Music from Seeds: Parametric Generation and Control of Seed-Based Music for Interactive Composition and Performance*, Master's Thesis, MIT Media Laboratory, 1994.
- [Ris66] Risset, J.-C. "Computer Study of Trumpet Tones." Murry Hill, N.J.: Bell Telephone Laboratories, 1966.
- [Ris89] Risset, J.-C. "Additive synthesis of inharmonic tones" In Mathews, M. and Pierce, J. R. *Current Direction in Computer Music Research*. Cambridge, Massachusetts: MIT Press, 159-164, 1989.
- [Ris91] Risset, J.-C. "Timbre Analysis by Synthesis: Representations, Imitations, and Variants for Musical Composition" in De Poli, G., Piccialli, A., and Roads, C. *Representations of Musical Signals* : Cambridge, Massachusetts: MIT Press, 1991.
- [Ros92] Rosenthal, D. *Machine Rhythm: Computer Emulation of Human Rhythm Perception*, Ph.D. Thesis, MIT Media Laboratory, 1994
- [SB88] Shanmugan, K. S., and Breipohl, A.M. *Random Signals: Detection, Estimation and Data Analysis*, New York, NY: John Wiley & Sons, Inc., 1988
- [Sla85] Slawson, W. *Sound Color*, Berkeley, CA, University of California Press, 1985.
- [Sma86] Smalley, D. "Spectro-morphology and Structuring Processes." *The Language of Electroacoustic Music*: Edited by Simon Emmerson: MacMillian, London, (4) 62-92, 1986.
- [SS90] Serra, X. and Smith, J. O. "Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic plus Stochastic Decomposition" *Computer Music Journal*, 14(4):12-24, 1990.
- [The89] Therrien, C. W. *Decision, Estimation, and Classification* : New York, NY: John Wiley & Sons, Inc., 1989.
- [TP91] Turk, M., and Pentland, A. "Eigenfaces for Recognition." *Journal of Cognitive Neuroscience*: 3(1):71-86, 1991.
- [Wes79] Wessel, D. L. "Timbre Space as a Musical Control Structure." *Computer Music Journal*, 3(2):45-52, 1979.
- [Wis85] Wishart, T. *On Sonic Art* :York, U.K., Imagineering Press, 1985.
- [Wis86] Wishart, T. "Sound Symbols and Landscapes." *The Language of Electroacoustic Music*: Edited by Simon Emmerson: MacMillian, London, (3) 44-60, 1986.