

**Computational Modeling and Analysis
of Multi-timbral Musical Instrument Mixtures**

A Thesis

Submitted to the Faculty

of

Drexel University

by

Jeffrey Scott

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

October 2014

Contents

List of Figures.....	iii
List of Tables.....	vi
Acknowledgements.....	viii
Abstract	x
1. Introduction	1
1.1 Motivation.....	1
1.2 Contributions	2
1.3 Organization	3
2. Background.....	5
2.1 Multi-Track Mixing.....	5
2.1.1 Psychoacoustics	7
2.1.2 Common Mixing Techniques and Practices	13
2.1.3 Current Software Audio Mixing Tools	15
2.2 Automated Mixing	17
2.2.1 Evaluating Mixing Assumptions.....	18
2.2.2 Cross-Adaptive Methods	20
2.2.3 Relating Perceptual Terms to Audio Effects	23
2.3 Perception and Modeling of Timbre.....	24
2.3.1 Timbre Perception	25
2.3.2 Modeling Global Timbre	31
2.3.3 Dynamic Timbre Modeling.....	37
2.4 Perceptual Feature Evaluation	42
2.4.1 Data Collection.....	43
2.4.2 Experiments and Results.....	43
3. Methods, Models and Features.....	47

3.1	Multiple Linear Regression	47
3.2	Linear Dynamical Systems	47
3.3	Dynamic Texture Mixtures	48
3.4	Principal Component Analysis	50
3.5	Non-Negative Matrix Factorization	51
3.6	Probabilistic Latent Component Analysis	52
3.6.1	Convolutional Formulation	53
3.7	Datasets	55
3.7.1	Rockband Dataset	55
3.7.2	Multiple Genre Dataset	56
4.	Instrument Based Processing	58
4.1	Stereo Panning	59
4.2	Relative Levels	60
4.3	Equalization	61
4.4	Drum Type Classification	62
4.5	Listening Evaluation	63
5.	Representing Dynamic Timbre	66
5.1	Modeling Instruments as Dynamic Textures	67
5.1.1	Parameter Estimation	68
5.1.2	Model Reduction and Synthesis	69
5.1.3	Modeling Timbre Variation	70
5.1.4	Joint Analysis	70
5.1.5	Altering Timbre	72
5.1.6	Results	73
6.	Supervised Learning of Instrument Mixtures	75
6.1	Weight Estimation	75
6.2	Modeling	78
6.2.1	Multiple Linear Regression	79
6.2.2	Linear Dynamical System	79
6.2.3	Results	79
6.3	Improved Architecture	83
6.3.1	Feature Analysis	85

6.3.2	Results	86
7.	Perceptual Evaluation of Features	92
7.1	Feature Extraction	92
7.1.1	Auditory Model	92
7.2	Basis Decomposition of Spectral Representations.....	93
7.3	Listening Evaluation of Timbre Reconstructions.....	95
7.3.1	Results	97
8.	Conclusions and Future Directions	100
8.0.2	Future Research	101
A.	Calculation of Mel Frequency Cepstral Coefficients	103
B.	EM Algorithm.....	105
C.	Album Effect on Feature Data	106
	Bibliography.....	109

List of Figures

2.1	Diagram picturing the basic process of multi-channel audio mixing.	6
2.2	An illustration of frequency masking.	8
2.3	Original spectra and modified spectrum of a guitar track after equalization to reduce masking effects.	9
2.4	Critical band filterbank with octave based center frequencies.	10
2.5	Log frequency spectrogram and critical band filterbank outputs of the song <i>No Phone</i> by Cake.....	11
2.6	Equal loudness contours (ISO:226 curves) [35].	12
2.7	Block diagram showing the processing involved in computing the loudness measurement in the International Telecommunication Union BS.1770-3 standard.	12
2.8	Stage 1 filter of the K-weighting filter for loudness estimation. This section approximates the acoustic absorption of the head.....	13
2.9	Stage 2 filter of the K-weighting filter for loudness estimation.	13

2.10	Iterative process of mixing. (Courtesy of Cyrille Tallandier)	14
2.11	The often complex interfaces in four major DAWs – (a) Steinberg Cubase (b) Apple Logic (c) Ableton Live (d) Avid ProTools.	16
2.12	Signal flow and architecture of a cross-adaptive mixing model.	21
2.13	Modeling procedure to relate words that describe sound and timbre to audio signal processing parameters.	23
2.14	Attack, decay, sustain and release segments of the ADSR envelope.	26
2.15	Temporal and spectral envelopes for bass guitar (a-b), cello (c-d), kick drum (e-f) and snare drum (g-h)	27
2.16	An example timbre space resulting from multidimensional scaling.	29
2.17	The change in the spectrum and overall shape of the spectrum (red line) for 5 seconds of audio.	32
2.18	Unlabeled data set (a), one Gaussian (b) and two Gaussians (c).	33
2.19	Unlabeled data set (a) and iterative k -means process (b)-(f).	35
2.20	System diagram of the spectro-temporal envelope extraction process.	38
2.21	System overview for spectro-temporal envelope extraction.	39
3.1	Diagram of a linear dynamical system modeling a noisy process.	48
3.2	Graphical model for (a) dynamic texture (b) dynamic texture mixture model.	49
3.3	A depiction of basis decomposition using convolutive PLCA. Two-dimensional kernel distributions are learned with corresponding activations (impulse distributions). These two components are convolved and multiplied by the latent weights (z) to produce the reconstruction of the original distribution.	54
3.4	Diagram of dataset preprocessing for each song in the RockBand dataset.	57
4.1	Processing chain to calculate the active areas of an instrument track.	59
4.2	Contours of gain attenuation for various γ .	62
4.3	Listening test results showing the number of ratings for each clip pair.	64
5.1	Spectral and temporal envelopes for snare drum (a-b) and a white noise burst with modified temporal and spectral envelope (c-d).	67
5.2	Average SNR and standard deviation computed for 21 piano tones against the model dimension n .	68
5.3	Average SNR and standard deviation computed for 21 piano tones by varying the number of Hankel observations with $n = 40$.	69

5.4	Top: Log-magnitude spectrogram for a piano tone produced with a “hard” articulation. Bottom: Re-synthesized piano tone generated from the output of the estimated LDS model.	71
5.5	C and \tilde{C} for hard velocity and re-weighted to be a lower velocity.	72
5.6	Top: Spectrogram of piano note B3 played with hard velocity. Middle: The same note with a re-weighted observation matrix to change the velocity. Bottom: The original sample of piano note B3 played with soft velocity.	73
6.1	Extracted weights for bass guitar using NNLS, Kalman smoothing and normalization. ...	76
6.2	Histogram of linear mixing coefficients.	77
6.3	Supervised machine learning of gain coefficients using LDS and MLR.	78
6.4	Results for weighting coefficient prediction using multiple linear regression (MLR). The estimated ground truth weights are shown in gray and the predicted coefficients are depicted in red.	81
6.5	Results for weighting coefficient prediction using a linear dynamical system (LDS). The estimated ground truth weights are shown in gray and the predicted coefficients are depicted in red.	82
6.6	System diagram detailing the ‘One Vs. All’ method for mixing coefficient prediction.....	83
6.7	MSE versus the number of stacked features used in training an LDS for each track. Note that the scale of each sub-plot varies. The minimum is indicated for each track.....	87
6.8	Comparison of ground truth (black) values with AT (gray) and OVA (orange) models for ‘More Than A Feeling’ by Boston.....	88
6.9	Comparison of ground truth (black) values with AT (gray) and OVA (orange) models for ‘Hammerhead’ by The Offspring.	89
6.10	Comparison of ground truth (black) values with OVA model using the single best feature (gray) and using the best combination of features (orange) for ‘More Than A Feeling’ by Boston.....	90
6.11	Comparison of ground truth (black) values with OVA model using the single best feature (gray) and using the best combination of features (orange) for ‘Hammerhead’ by The Offspring.	91
7.1	Log frequency spectrogram and critical band filterbank outputs of the song <i>No Phone</i> by Cake.	93
7.2	Mel and constant- Q filterbanks depicted in log frequency scale.....	94

7.3	The spectrogram of an instrument mixture is perceptually filtered and a set of bases functions are computed using PCA, NMF and PLCA. The resulting bases from the mixture are used to reconstruct the individual instrument files.	95
7.4	An example question from the perceptual survey asking participants to identify the instrument present in the audio reconstruction.....	97
7.5	Listening test results showing the number of correctly identified instruments based on reconstruction type.	98
7.6	Listening test results showing the number of respondents expressing inability to determine instrument class for each reconstruction type.	99
A.1	The mel frequency scale (a) and mel filterbank (b).....	103
A.2	MFCC calculation.	104
C.1	MFCCs for 30 seconds of audio for several songs per album.	107

List of Tables

2.1	Common digital audio workstations and effect plug-ins.	15
2.2	Common features extracted for timbre analysis.	31
2.3	Instrument recognition results for polyphonic and monophonic audio samples. [8].	42
2.4	Normalized difference error between the valence/arousal ratings for the reconstructions versus the originals.	44
2.5	Percentage of paired comparisons that yielded the desired perceptual result for mode and tempo.	45
4.1	Mixing parameter values for individual drum tracks.	60
4.2	Features used in drum type classification.	62
4.3	Listening test participant familiarity with audio mixing and production.	64
6.1	Average mean squared error across all songs between ground truth weights and predicted weights for MLR and LDS.	80

6.2	Results for LOOCV on the database. The MSE for each track across all songs is shown for the All Tracks method and the One Versus All approach. The Best Features column is the result from sequential feature selection.	84
6.3	Spectral and time domain features used in mixing coefficient prediction task.	85
6.4	Mean squared error for all features and individual instruments. Features for each instrument are listed in order of best performance to worst performance. The best combination of features for each instrument is in boldface.	86
7.1	Number and type of instruments used in the reconstruction listening experiment.	96
7.2	Number and type of instruments used in the reconstruction listening experiment.	97

Dedication

For Nana

Acknowledgements

I would like to thank Youngmoo Kim for his support and guidance, for welcoming me in the lab and showing me just how cool music research can be. I am extremely grateful to Joshua Reiss for his detailed feedback and for traveling so far to attend the dissertation proposal and defense. My sincerest thanks go out to John Walsh, Yon Visell and Cyrille Taillandier for being on my committee and providing their expertise.

There are many great lab mates who helped out on projects large and small. I would first like to thank Erik Schmidt and Matthew Prockup for making all of those late nights in the lab fun and for always being available to bounce ideas off of. The lab was a great place to learn and grow and I appreciate everyone's help and input over the years, specifically from Ray Migneco, David Rosen, Brandon Morton, Jeff Gregorio, David Grunberg, Brian Dolhansky, Alyssa Batula, Mike Caro, Patrick Richardson and Travis Doll.

I would also like to recognize two wonderful professors from my undergraduate education at Ramapo College, Phil Anderson and Daniela Buna. Their enthusiasm and passion for science was infectious.

None of this would have been possible without the unending love and support of my grandmother who was always willing to listen to stories about graduate school. My brother Colin has always been available to provide help of any kind at the drop of a hat, for that I am grateful. Thanks go out to my Mom for instilling in me resilience and perseverance, qualities essential to make it through graduate school. I appreciate the friendship of Adam Nash, Sal Galati and David Rosen. Having a musical outlet throughout this period was essential. Thank you to Jess Cosca for being so patient and understanding with me throughout the whirlwind leading up to the completion of this thesis. I am so happy that we have found each other.

Abstract

Computational Modeling and Analysis
of Multi-timbral Musical Instrument Mixtures

Jeffrey Scott

Advisor: Youngmoo E Kim

In the audio domain, the disciplines of signal processing, machine learning, psychoacoustics, information theory and library science have merged into the field of Music Information Retrieval (Music-IR). Music-IR researchers attempt to extract high level information from music like pitch, meter, genre, rhythm and timbre directly from audio signals as well as semantic meta-data over a wide variety of sources. This information is then used to organize and process data for large scale retrieval and novel interfaces.

For creating musical content, access to hardware and software tools for producing music has become commonplace in the digital landscape. While the means to produce music have become widely available, significant time must be invested to attain professional results. Mixing multi-channel audio requires techniques and training far beyond the knowledge of the average music software user. As a result, there is significant growth and development in intelligent signal processing for audio, an emergent field combining audio signal processing and machine learning for producing music.

This work focuses on methods for modeling and analyzing multi-timbral musical instrument mixtures and performing automated processing techniques to improve audio quality based on quantitative and qualitative measures. The main contributions of the work involve training models to predict mixing parameters for multi-channel audio sources and developing new methods to model the component interactions of individual timbres to an overall mixture. Linear dynamical systems (LDS) are shown to be capable of learning the relative contributions of individual instruments to recreate a commercial recording based on acoustic features extracted directly from audio. Variations in the model topology are explored to make it applicable to a more diverse range of input sources and improve performance.

An exploration of relevant features for modeling timbre and identifying instruments is performed. Using various basis decomposition techniques, audio examples are reconstructed and analyzed in a perceptual listening test to evaluate their ability to capture salient aspects of timbre. These tests show that a 2-D decomposition is able to capture much more perceptually relevant information with

regard to the temporal evolution of the frequency spectrum of a set of audio examples. The results indicate that joint modeling of frequencies and their evolution is essential for capturing higher level concepts in audio that we desire to leverage in automated systems.

1. Introduction

1.1 Motivation

Technology has had a tremendous impact on the way music is created, performed and enjoyed in the past century. The advent of recorded music allowed audiences to enjoy performances without leaving the comfort of their home. As the methods and equipment for capturing and processing audio advanced, the process of recording became an art-form in itself and the recording engineer and producer became just as essential as the musician, composer or conductor. Currently, producers and engineers are highly sought after artists in their own right and the concept of using the “recording studio as an instrument” has become commonplace. As music production, performance, recording, listening and distribution becomes ever more dependent on technology, researchers and professionals are beginning to rethink the entire pipeline from idea to recording, to the listener’s ear.

Computing technology is being leveraged to process and understand the world we live in on a high level of abstraction and the realm of music and audio is no exception [6, 23]. The turn of the century has ushered in a surge of advancement in digital technologies, specifically in the sphere of media processing, indexing, organization and retrieval. The vast amount of content created daily and easily uploaded to the internet has generated a need for powerful automated tools to help providers and consumers make sense of what is out there. Entire new modalities of interaction with tools for content consumption and creation are now possible through the sustained efforts of interdisciplinary researchers, entrepreneurs and professionals. In the audio world, researchers from a wide variety of fields including signal processing, machine learning, psychoacoustics, information theory and library science have combined their efforts to analyze the way we process music on a physical and psychological level. Music Information Retrieval (Music-IR) researchers attempt to extract high level information such as pitch, meter, genre, rhythm and timbre directly from audio signals as well as semantic meta-data over a wide variety of sources. This information is then used to organize and process data for large scale retrieval and novel interfaces.

Digital audio production tools have also significantly impacted the way we consume, produce and interact with music on a daily basis. Consumers have the ability to create quality recordings in a home studio with a relatively limited amount of equipment and mobile devices provide easy to use platforms for performance, composition and remixing. In the professional audio sphere,

although there is a wide variety of digital audio workstations (DAW) and plug-in suites available, the level of expertise required to operate them proficiently necessarily inhibits many newcomers from obtaining reasonable results even with a significant amount of effort. This has led to an exploration in the audio signal processing community for methods of automatically analyzing audio and improving the perceived quality. Several significant difficulties arise when attempting this task. The qualitative difference between the preference of individuals, the wide range of timbre, dynamics and instrumentation and the multitude of production techniques available present ample hurdles to overcome.

This thesis explores methods, models and representations for working with audio from a standpoint of music production and creation primarily involving multi-channel (separated) audio sources. The three key areas of investigation are the following:

1. Inferring high level perceptual information from audio tracks
2. Analyzing the relationships between audio tracks
3. Developing salient feature representations of tracks

The experiments presented herein address one or more of these topics through a variety of methods. The contributions in these areas are outlined below.

1.2 Contributions

This thesis approaches the problem of multi-track audio processing from both an analysis and synthesis perspective. I investigate methods to automatically process audio using time varying models, discuss acoustic feature salience for mixing models and present a framework for estimating timbre contributions of individual instruments in a mixture. The contributions are ordered by the authors opinion of significance to the field.

1. In Chapter 6 supervised machine learning is used to approximate mixing tasks from data. Parameters relating to control values that mixing engineers use to process audio are extracted from a multi-track corpus. A framework is proposed to model the relation between acoustic features extracted directly from audio and the application of these control values [82]. It leverages a representation of the time-varying characteristics of audio using linear dynamical systems (LDS). That approach is improved by reducing the constraints on the model and generalizing it to a larger number of instruments. Additionally, we explore an extended feature

set within this framework and analyze the performance of each individual feature as well as combinations of features. The features are chosen to contain information about the total energy of the signal, energy within various frequency bands, spectral shape and dynamic spectral evolution [81].

2. Having shown that the LDS approach is able to model specific characteristics of the audio, Chapter 5 investigates its ability to synthesize notes and reproduce timbres. A corpus of instrument tones is represented using linear dynamical systems and then re-synthesized, showing the capability of the model to capture and alter perceptual characteristics [78].
3. Chapter 7 discusses a set of experiments designed to evaluate different features for timbre and instrument identification. Individual instrument examples are reconstructed from features. The lossy nature of this reconstruction is investigated to determine whether salient aspects of the audio signal that humans use to perceive individual timbres are retained. The results show that 2-D representations, those that consider the temporal evolution of the spectrum are much more perceptually relevant in a computational framework.

1.3 Organization

In Chapter 2, I discuss the relevant background information and previous work relating to the experiments in this thesis. It encompasses a range of subjects due to the interdisciplinary nature of the work, opening with a summary of the perception of audio including the psychoacoustic principles of masking, loudness and timbre. Multiple approaches of timbre modeling are discussed including of global models and dynamic timbre models. A summary of audio engineering principles and practices are presented followed by recent work on automated mixing techniques and relationships between audio perception and mathematical modeling.

Chapter 3 outlines the mathematical formulations and models used throughout the subsequent chapters. The datasets used in the thesis are also discussed.

In Chapter 4, an experiment to determine the efficacy of an approach to multi-track mixing based on information about the instrument type in a multi-track session is performed.

The material in Chapter 5 presents experiments to synthesize audio and manipulate it using linear dynamical systems as well as represent the temporal evolution of timbre. Methods for representing audio mixtures and analyzing the contributions of components to the mixture are discussed and an evaluation of the models to capture salient aspects of timbre is completed using listening tests.

Chapter 6 evaluates supervised techniques for processing multi-channel audio and Chapter 7 analyzes commonly used features in the community for representing musical instrument timbre.

Chapter 8 summarizes the findings presented herein and recommends future directions for researchers based upon the results of this work. Audio examples and related materials may be found online¹

¹<http://music.ece.drexel.edu/research/AutoMix>

2. Background

This chapter provides an overview of research in the areas of intelligent audio processing, timbre perception/modeling and feature design that are relevant to the developments presented in later chapters. First, I will familiarize the reader with major concepts and practices in music production from a technical standpoint. Those with a prior knowledge of mixing engineering and perception of sound may want to skip ahead to Section 2.2 for an overview of research related to automatic mixing. The subsequent section discusses the literature of timbre perception as well as computational modeling of timbre. The final section of the chapter highlights previous research on evaluating perceptual information in features based on listening tests.

2.1 Multi-Track Mixing

This section presents common concepts and practices employed in mixing audio. The mixing engineer uses the tools at their disposal to modify a signal with respect to the time, frequency and spatial domains. The time domain representation of the signal is the waveform captured by a microphone or otherwise synthesized electronically. Common time domain processing operations include dynamic range compression, noise gating, amplification and attenuation. Frequency domain operations are generally accomplished using a Fourier representation and either the complex or magnitude spectra is used depending on whether the goal is processing or analysis, respectively. The spatial domain refers to the stereo field and depth of field. The stereo field is the perceived direction that a sound is coming from and depth of field refers to the perceived closeness (distance) to the listener. Nearly all tools available to a mixing engineer will modify one or more aspect of the signal in frequency, time or space.

The mixing procedure consists of processing at multiple scales with respect to the input tracks. The engineer will apply processing to each individual track as well as sub-groups (i.e. drums, vocals, guitars) and to a lesser extend the mixture as a whole. This workflow is depicted in Figure 2.1.

One important aspect of mixing is defining the goals and objectives that the array of processing techniques employed by the engineer will accomplish. On a global level, there are no concrete qualitative measurements or features that will guarantee a good mix or even an acceptable mix. The quality of a mix-down is dependent upon the instrumentation and arrangement of the song as

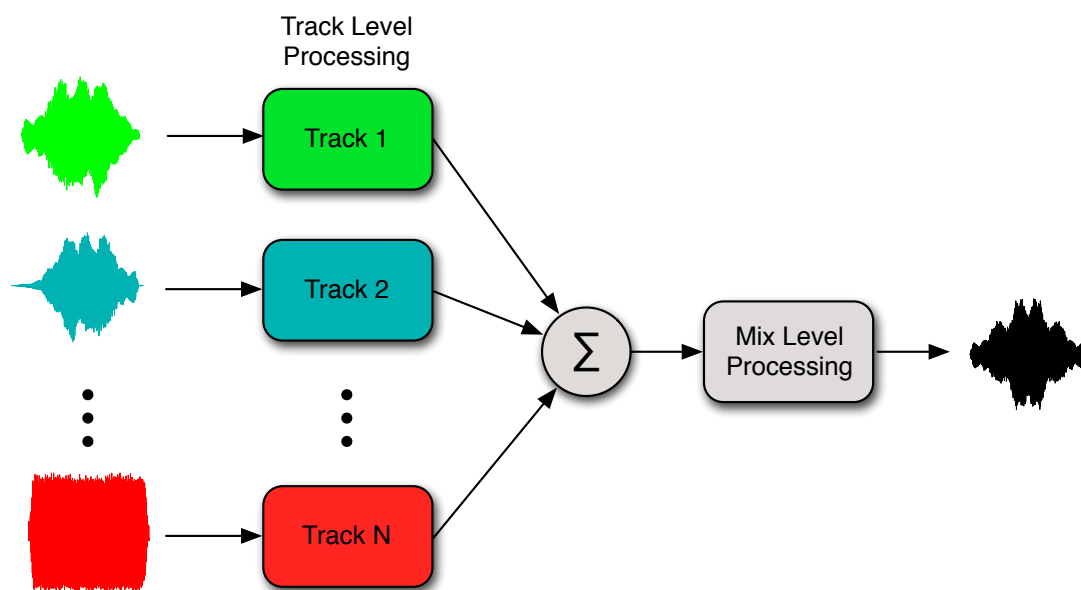


Figure 2.1: Diagram picturing the basic process of multi-channel audio mixing.

well as the audio fidelity of the source material. A poor performance of a bad song that was not recorded properly cannot be transformed into a hit through mixing alone. However, there do exist general guidelines one can follow and pitfalls one should avoid to achieve a better sounding mix.

Two of the main objectives of the mix are developing a balance in the spatial, time and frequency domains and ensuring clarity and definition of the instruments. Processing performed to reach one goal may be complementary with another goal or could have an adverse affect on other sonic objectives. For example, increasing the low-mid range frequency content of a piano may give it more body and warmth but when evaluated in the context of the other instruments in the mixture can create a muddy sound. It is often the case that processing an instrument sounds good in context of the mixture but detracts from the quality of the recording when listening to the instrument by itself.

While the desired spectral balance can often depend upon the genre of the music being mixed, a general rule is to avoid significant excess or deficiency in specific frequency ranges. Too much low frequency content below approximately 250 Hz will create a boomy or muddy sound. Conversely a lack of energy in the low register will result in a thin or weak sound.

Clarity and definition are related to spectral balance but also apply to the spatial domain. In order to hear individual instruments clearly when they are played simultaneously it is important for

there to be distribution across the stereo field. If many instruments are panned to the same position, the clarity of each instrument will be reduced. Definition is also related to depth of field which can be manipulated primarily through applying reverb. Adding reverb effectively pushes an instrument further away from the listener. While increasing the perceived distance of a sound from the listener will create space in the mix, it will also decrease the clarity and definition of the source.

A good summary of the various techniques and practices employed in mixing engineering may be found in [37, 83]. Many of the assumptions about mixing audio and methods for applying processing are discussed in [64]. In this work, Pestana explores various commonly used techniques and uses listening evaluation and self-report from professional engineers in an attempt to quantify the decisions of engineers.

Before delving into some of the common techniques and tools for mixing we must first discuss how humans perceive audio and music. Several aspects of psychoacoustics are essential to the mixing engineer's decision making process and as a result, determine their choice of signal processing techniques to use. In addition, there is significant literature about modeling the auditory process computationally which can be leveraged in developing automation techniques for multi-channel mixing.

2.1.1 Psychoacoustics

The methodical study of human perception of sound is known as psychoacoustics. Psychoacoustic principals result from the physical constraints of sound propagation, the conversion of the sound to electrical potential in the human auditory system and the cognitive processing of sound in the brain. Many sub-topics exist within psychoacoustics including sound source localization, binaural processing, pitch perception, timbre, masking and loudness. Here we focus on masking, loudness and timbre as they relate to monophonic and polyphonic audio.

Masking

Auditory masking refers to the phenomenon of certain sounds being imperceptible in the presence of other sounds. This occurs due to the physical mechanism for translating the mechanical energy absorbed in the middle ear into electrical signals to be passed through the auditory nerve [16]. If two sinusoids occur within the *critical bandwidth* then masking will occur. Consider the 1000Hz sinusoid played at 70 dBSPL in Figure 2.2. To be perceived, another sinusoid would have to be played with amplitude larger than that of the masking threshold depicted. A low amplitude sinusoid will not be

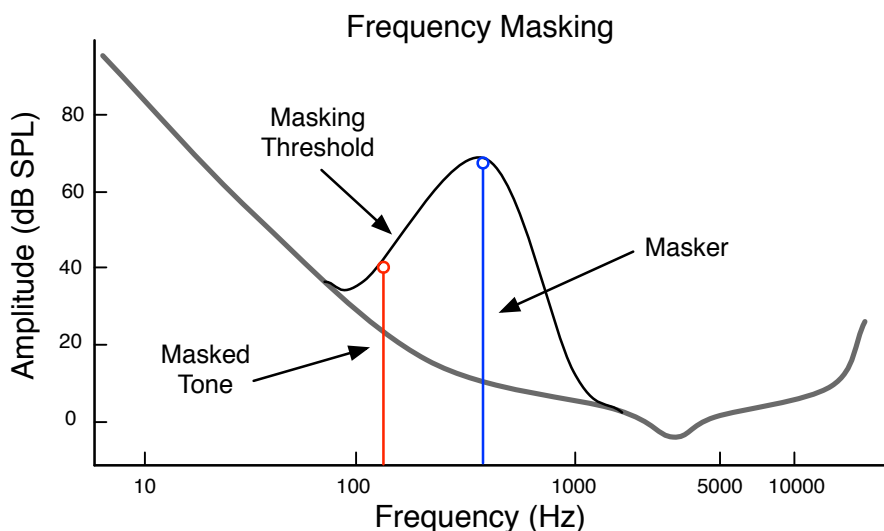


Figure 2.2: An illustration of frequency masking.

perceived if there exists a higher amplitude sinusoid of similar frequency at the same time. As the sinusoids become separated in frequency, the masking effect is reduced and the masked sound will become audible. Alternatively, the amplitude of the quieter sinusoid could be increased above the masking threshold, allowing a listener to perceive both sinusoids.

Frequency masking is a significant consideration in multi-track mixtures. As the instrumentation of a song becomes denser (i.e. more instruments), masking plays an ever increasing role and instruments that share the same frequency range will fight for intelligibility in the mix. Frequency masking can be either *complete*, where one sound is rendered inaudible by the presence of another, or *partial* where the perceived loudness of one sound is affected by the concurrent sound.

The engineer has two tools to deal with problems created by masking: equalization (EQ) and panning. To reduce the effects of masking and increase clarity and definition in both instruments, a filter is applied to ‘carve out’ a frequency range in one instrument to make room for the other. Often, one instrument is chosen as the desired instrument to be heard and a target frequency range of overlap is determined. In Figure 2.3, a vibraphone has significant energy in the 400-1000 Hz range that overlaps with the guitar track. The guitar is filtered with a band-stop filter and the resultant spectrum is shown. This has the effect of making the vibraphone more prominent in the mixture. In addition, filtering the signal also makes the guitar more defined since there is greater frequency separation between the two instruments.

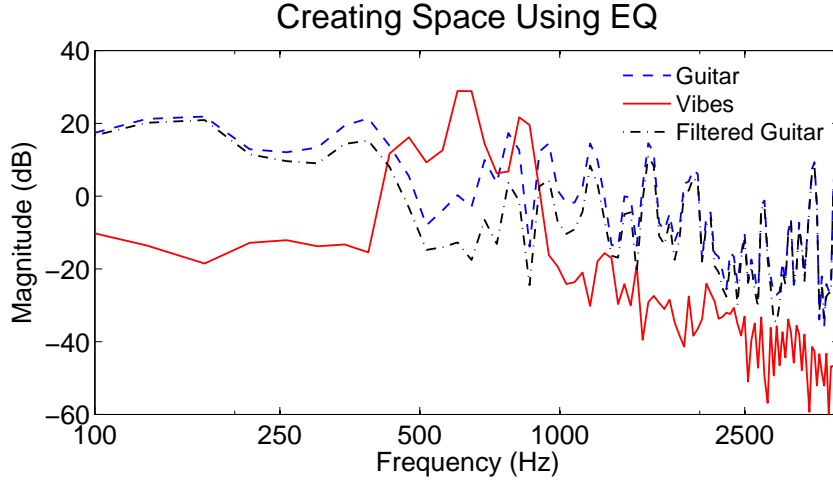


Figure 2.3: Original spectra and modified spectrum of a guitar track after equalization to reduce masking effects.

Methods for modeling masking generally rely on heuristic models derived from the critical bandwidth mentioned above [7]. The critical bandwidth of an auditory filter roughly defines the range in which another sound will cause masking and effect the perception of the other sound. The bandwidth is defined in terms of the center frequency of the filter and increases as the center frequency increases. This relationship approximates the observation that humans have more ability to differentiate frequencies that are close together at the lower end of the frequency spectrum. The Equivalent Rectangular Bandwidth (ERB) is often used to specify the relationship between the center frequency and the critical bandwidth,

$$ERB = 24.7(4.37f + 1), \quad (2.1)$$

where f is the center frequency in kHz [52].

A critical band filterbank is shown in Figure 2.4. Each filter channel has approximately equal energy and the channels are spaced logarithmically over the range of human hearing. One common method of implementing a critical band filterbank is to use a gammatone filterbank with the center frequencies distributed though the frequency domain in proportion to their bandwidth [56]. A comparison of the spectrogram (513 dimensions) and the output from a 10-band critical band filterbank is shown in Figure 2.5.

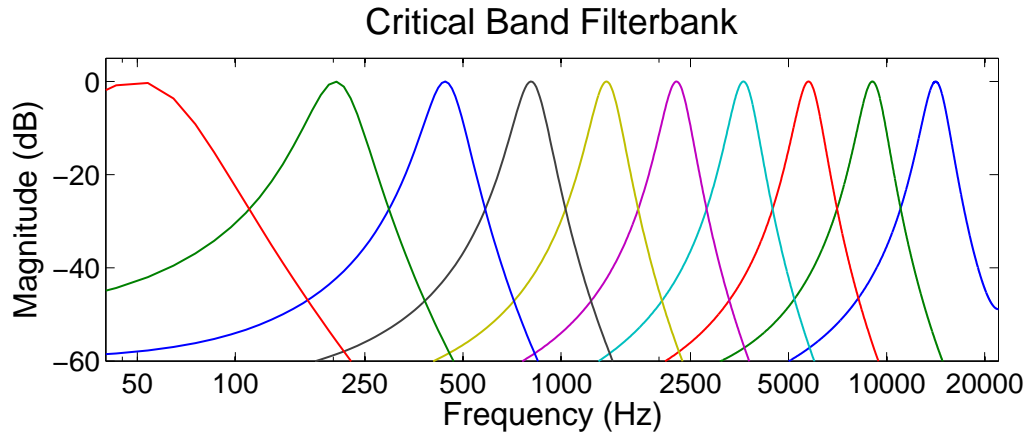


Figure 2.4: Critical band filterbank with octave based center frequencies.

Loudness

There are many tools and methods to monitor level and amplitude of a signal but loudness is an inherently perceptual measure. Two important concepts central to a discussion about loudness are frequency dependence and the just noticeable difference (JND). Frequency dependence forms the core of all loudness models. As the frequency of a sound is modulated but kept at constant amplitude, our perception of the loudness of the sound will change. Therefore, what we commonly refer to as volume or loudness is an inherently perceptual quantity and cannot be analytically defined as a relation to amplitude or some other physical measure such as RMS energy. Experiments have shown that human sensitivity to loudness and frequency change is greatest in the mid-range frequencies and is reduced in both the very low and high audible ranges [16].

Figure 2.6 shows the equal loudness contours originally developed by Fletcher and Munson through a series of perceptual experiments and modified in the figure by the International Standards Organization [24, 35]. Each point along one of the curves represents equal perceptual loudness (phons) for a pure tone (sinusoid). At quiet volumes, low frequencies require significantly greater amplitude than frequencies in the middle register to be perceived at all. The low frequencies (below 100Hz) exhibit much less susceptibility to changes in sound pressure level (SPL) with regard to perceived loudness. A 50 Hz sinusoid needs to be almost 55 dB SPL to sound as loud as a 1kHz sinusoid at 10 dB SPL, a tremendous increase. It is also worth noting that the equal-loudness contours change shape as overall volume increases. The curve for 90 phons is much flatter than the curve at 10 phons specifically in the low frequencies.

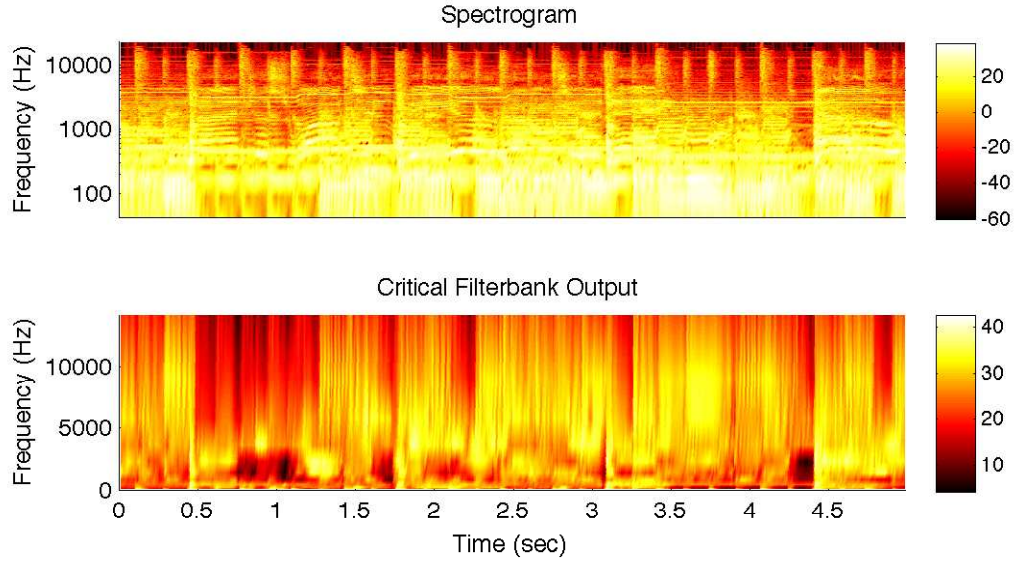


Figure 2.5: Log frequency spectrogram and critical band filterbank outputs of the song *No Phone* by Cake.

There are several standards for modeling perceptual loudness as opposed to simple amplitude or intensity monitoring as is done in volume unit (VU) meters. The International Standards Organization (ISO) Normal equal-loudness-level contours specify sound pressure levels for pure tones (sinusoids) similar to the Fletcher-Munson curves. The International Telecommunication Union developed specifications for measuring loudness in [36]. This standard implements a four stage process to model loudness consisting of frequency weighting using a two-stage filtering process, mean square calculation, channel weighted summation and multi-threshold gating. This specification is designed for use in broadcast and monitoring complex sounds and does not apply to pure tones as is the case of the ISO standard. Figure 2.7 shows the signal flow involved in computing the loudness measurement. The K-weighting filter specifies a two-stage filtering operation. The first stage accounts for the acoustic effects of the head and is based upon a rigid body spherical approximation. This is a second order IIR filter with frequency response shown in Figure 2.8. The result is a 4 dB hi-shelving filter with a transition band starting around 1 kHz. Stage two of the K-weighting filter is also a second order IIR filter. In this case, it is a high pass filter with the passband starting around 200Hz. Notice that these resemble the inverse of the basic shape of the Fletcher-Munson curves in Figure 2.6. Let us consider a signal y the result of passing an original signal, x through the K-weighting

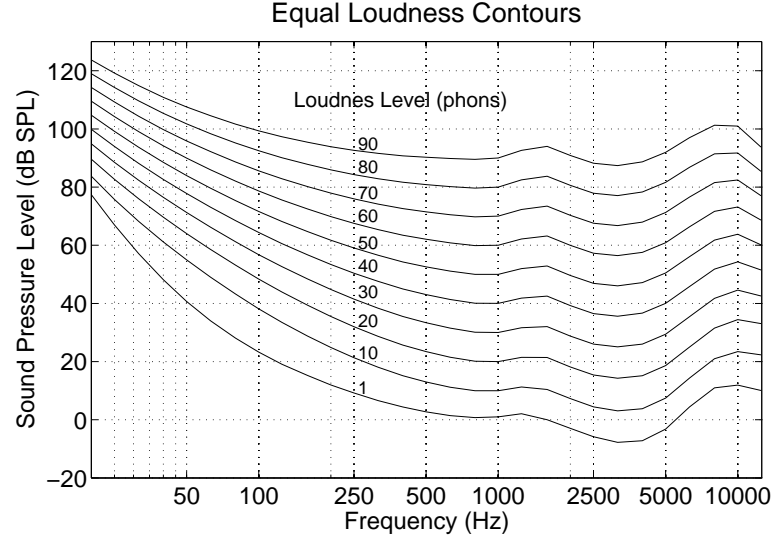


Figure 2.6: Equal loudness contours (ISO:226 curves) [35].

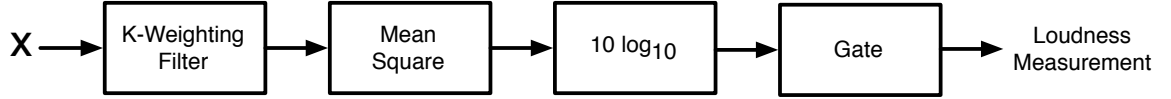


Figure 2.7: Block diagram showing the processing involved in computing the loudness measurement in the International Telecommunication Union BS.1770-3 standard.

filter. Then the power, p of the signal is computed as

$$p = \sum_{n=0}^T y[n]^2, \quad (2.2)$$

and the loudness is given by

$$L_k = -0.691 + 10 \log_{10}(p). \quad (2.3)$$

Loudness models are often applied prior to extracting features from audio. They have been used as part of front end feature extraction models for a variety of Music-IR tasks. They are particularly relevant for tasks where we are trying to emulate what a listener hears rather than perform brute force computation to find patterns. The next section relates these psychoacoustic principles to techniques and practices used to mix audio sources.

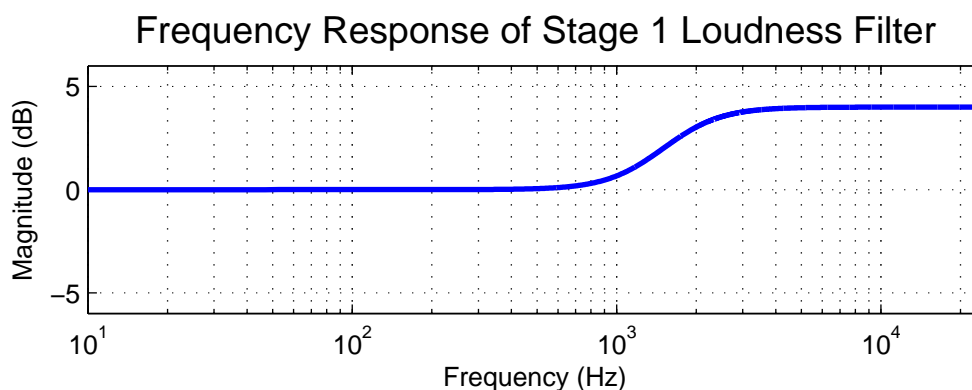


Figure 2.8: Stage 1 filter of the K-weighting filter for loudness estimation. This section approximates the acoustic absorption of the head.

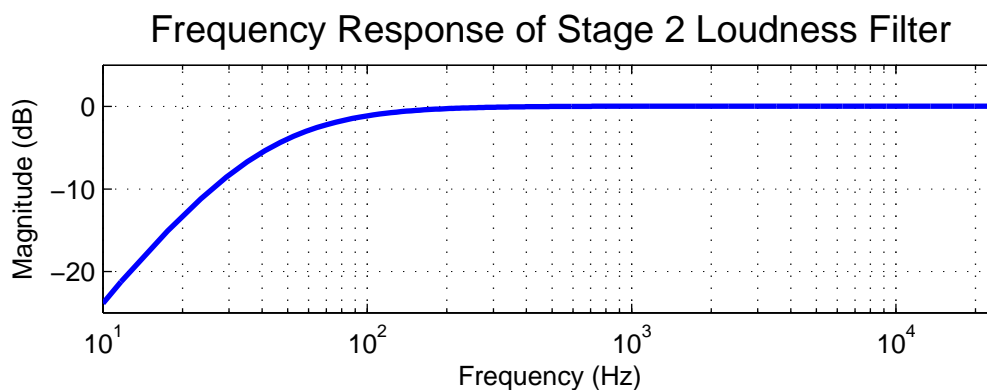


Figure 2.9: Stage 2 filter of the K-weighting filter for loudness estimation.

2.1.2 Common Mixing Techniques and Practices

Multi-track production often involves a constant evaluation and re-evaluation of the mix. Changes made to one instrument or group of instruments will necessarily impact the perception of the other instruments in the mix. A coarse to fine approach is often employed where large changes are made first and then smaller and smaller changes are applied with each iteration. A basic grouping of the processing categories is as follows:

- Levels - The gain (boost/attenuation) applied to each track
- Panning - The perceived position of the source in the stereo field
- Equalization - Filtering applied to boost or cut specific frequency ranges for desired effect
- Dynamics - Non-linear processing to control/normalize the changes in the energy of a track

- Effects - Modulation, delay, reverb, etc.

Since each change made will affect the objectives of spatial and frequency balance as well as instrument definition and clarity, it is common to return to balancing the levels after making other processing decisions. For example, panning a synthesizer to the left in order to create space and prevent overlap with the vocals may cause the synthesizer or vocals to be too loud compared to the rest of the accompaniment and need to be attenuated. Compressing the vocal line to normalize the volume may cause it to become too soft in the mix and require a compensating level boost. This process is summarized in Figure 2.10.

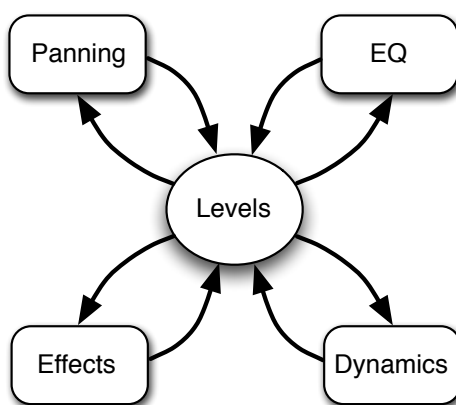


Figure 2.10: Iterative process of mixing. (Courtesy of Cyrille Tallandier)

There are different approaches to the order in which the instruments are mixed in addition to the order processing is applied. Two main approaches are the **serial** and **parallel** orders. The serial approach involves focusing on more important layers first such as the lead vocals or drums while there is more space in the mix and adding layers in order of importance. One caveat of this method is the potential lack of space for the instruments added in at the end of the process.

The parallel approach starts with all instruments audible and adjusts levels to get a rough mix. Once this is attained, the engineer will follow a process similar to that outlined in Figure 2.10. An advantage to this method is that the mix as a whole is constantly being evaluated. One difficulty that may arise (especially in sessions with many tracks) is an inability to focus on individual elements in the mix. These two different approaches may inform decisions about how to computationally model the mixing process. One of the causes for recent interest in this area of research is the difficult

and time-consuming nature of using the existing audio tools available on the market. The following section outlines some of those difficulties and shows how there is increasing demand for automated tools to assist in music production.

2.1.3 Current Software Audio Mixing Tools

There are a plethora of tools for multi-track audio production on the market, far too many to make a comprehensive list of any use. However, these can be divided into two main classes, digital audio workstations (DAWs) and audio effect plug-ins. DAWs are standalone programs for full-service editing and manipulation of multi-track audio and symbolic data (MIDI/OSC/etc.). They provide the capability to sum tracks, perform processing and route audio both internally and externally. Most DAWs come with a standard set of basic audio effect plug-ins and allow for third party plug-ins to be easily integrated into the processing framework. A brief overview of available tools is shown in Table 2.1.

DAWs	Plugins	
Logic	Equalizer	Phaser
ProTools	Compressor	Flanger
Cubase	Limiter	Delay
Studio One	Gate	Chorus
Digital Performer	Stereo Spread	Distortion
Reaper		
Live		

Table 2.1: Common digital audio workstations and effect plug-ins.

One of the primary reasons for developing intelligent software for music analysis and processing for multi-channel audio is the inherent complexity of the task. The tools for producing quality audio possess a vast amount of options and are rather daunting for a new user to familiarize themselves with. Screenshots of four major DAWs are shown in Figure 2.11. Each has a similar facade where the audio tracks are presented in a horizontal display with controls for each track on a sidebar. Some of the functionality can be hidden or obscured for improved workflow and some desired functionality may be buried within several sub-menus of a confusing hierarchy.



Figure 2.11: The often complex interfaces in four major DAWs – (a) Steinberg Cubase (b) Apple Logic (c) Ableton Live (d) Avid ProTools.

Intelligent Plugins

Whereas most plug-ins provide increased audio fidelity or a more intuitive interface or emulation of a ‘legendary’ piece of audio hardware, there is a recent trend to produce software that can *listen* to the signal and make decisions based on a higher level construct rather than amplitude or thresholding.

Melodyne, released by Celemony in 2000, is one of the first such tools. While the original version provided tools to alter pitch, manipulate formants and alter vibrato, the more recent release is able to separate individual pitches in polyphonic audio and manipulate them separately with minimal effect on the remaining signal.

The **Vocal/Bass Rider** plug-ins from Waves allow a user to set a target loudness for a track in relation to the other tracks in the instrument mixture. This effect is similar to a dynamic range compressor except that it does not just normalize the loudness with respect to the track it is modifying, it considers the loudness of the overall mixture.

Trackspacer by Waves attempts to create space in a mix for a target track. The tool analyzes

the frequency content in on track and automatically filters out similar frequency content in another track. The user still must identify the tracks that overlap in frequency but the tool simplifies the process of correcting this issue.

Unfilter is a plug-in by Zynaptiq that attempts to remove unwanted filtering due to a recording environment or processing chain. The software attempts to learn a filter that will compensate for the change in the spectral envelope of the original audio.

Ozone from Izotope takes spectral ‘snapshots’ of a reference recording. This spectral profile is used to create an equalization filter that tries to match the envelope of the source material to the reference recording.

2.2 Automated Mixing

Intelligent automated combination of multi-channel audio is a relatively new endeavor in the research community. Adaptive digital audio effects describes an architecture that analyzes input tracks to appropriately determine the parameters used to control the signal processing chain of a mixture. At the core level, the concept is not new, evidenced by the implementation of common effects such as compressors and limiters. A compressor applies a non-linear gain based on the RMS energy in a signal. The parameters allow a user to decide the degree of compression as well as the response time and threshold for activation. Determining the proper amount and type of compression to use is a fairly advanced skill for a mixing engineer and is often misunderstood and misapplied by novice engineers. One goal of intelligent music processing systems is to develop models to automatically make estimates of effect parameters based on a set of input tracks, psychoacoustic modeling, audio engineering best practices and machine learning.

Early work dealt with live situations primarily focused on speaking engagements with multiple microphones [20]. The goal in this scenario was to selectively deactivate microphones when they were receiving no input, thereby reducing feedback as well as comb filtering effects due to the multi-microphone setup.

More recent work in the live setting focuses on creating a more balanced and artifact free mixture. Determining and correcting comb filtering effects when there are multiple sources present with multiple microphones is explored in [15]. Methods for adjusting the gains for both the performer monitor mixes as well as the front of house (audience) mix are explored in [62, 90]. These methods rely on an equal loudness assumption that attempts to normalize the perceptual loudness of the

sources in the mixture to ensure that all layers can be heard by the listener. A detailed explanation of several methods for automatically modifying panning, equalization, levels and time offset correction can be found in [60].

2.2.1 Evaluating Mixing Assumptions

Recent work has focused on exploring and validating the generalization of techniques that mixing engineers use in the context of producing a track for release [64, 65]. In [64], Pestana generates a series of 88 assumptions of how mixing decisions are made and explores and validates them using a variety of strategies. This is the most thorough execution of exploring mixing assumptions to date. The assumptions span the space of possible signal processing operations and their subsequent effect on spatial and frequency balance. Many of the assumptions have to do with the relative levels of instruments and their role in the mix, the effect of panning, equalization and compression. A few example assumptions are stated below:

- All signals should be presented with equal loudness.
- No element should be able to mask any of the frequency content of the vocals.
- The main track is always panned centrally
- Low-end frequencies should be centrally panned
- Hard panning should be avoided
- Equalization use should always be minimized
- There is an optimal amount of compression in terms of dB and it depends on sound source features

The assumptions are separated into categories (loudness, panning, equalization, temporal processing, dynamic range control) and evaluated based on the quantitative and qualitative measures below

1. Measuring parameters from mixing sessions of successful songs
2. Having successful sound engineers perform specifically tailored mixing exercises
3. Measuring features from completed successful mixes

4. Performing subjective listening tests on experienced subjects
5. Analyzing through quantitative surveys the habits of successful mixing engineers
6. Performing exploratory interviews with successful mixing engineers
7. Using literature review

The primary conclusion for loudness is that all instruments should not be equally loud. There is an order of importance, with the vocals always being the primary element in the mix. Additionally, no other element should mask the frequency range of the vocal tracks. It was also found that the order of precedence changes over time as the arrangement of a song progresses [64].

Pestana found that panning processes exhibited the strongest conclusions. Low-frequency content should be centered as well as the main element in the mix (vocal/melody). In sessions with high track counts, most of the other elements will be panned off-center to some degree. Exceptions occur for sparse arrangements, but this general rule was shown quantitatively through comparing RMS energy of left and right channels. Two key assumptions that were disproved are that wide panning (full left/right) should be avoided and that the degree of panning should be proportional to the amount of high frequency content in the signal.

Some of the most interesting results arise from the equalization assumptions. The common assumption of applying a high-pass filter when there is no low frequency content was shown to be infrequently performed. Assumptions about using subtractive equalization more than additive equalization as well as generalizations about engineers making minimal use of equalization were also shown to be false. Engineers stated that there is no target spectral profile (envelope) however it was found that there is significant similarity especially when grouping songs by decade and genre [66]. This was shown by comparing relative spectral shape independent of absolute magnitude across a corpus of popular tracks from 1950-2010.

Temporal processing involves a higher number of parameters than panning loudness and becomes more difficult to analyze as it is difficult to control for all parameters. Pestana found that there is little correlation between tempo and reverberation time yet delay time is frequently quantized with the tempo. The level of the reverb signal was found to be preferred around 9 Loudness Units (LU) relative to the level of the dry signal. Other components of reverberation application were inconclusive, specifically the use of pre/post high- and low-pass filters.

Finally, the assumptions about dynamic range compression sought to determine what situations warranted use of compression as well as the desired settings of the parameters (primarily

attack/threshold/release). Control of low-frequency content and erratic changes in loudness were the two main technical reasons for applying compression. Surprisingly, the control of low-frequency content was more prominent in both the mixing exercises and subjective listening tests. Stronger correlations were found between instrument type rather than acoustic features in the signal.

The work presented in [64] provides a great base for implementation of systems that can apply the above concepts in an automated fashion. This signals a significant shift as intelligent tools for the creation of and manipulation of audio for production purposes is slated to become a reality.

2.2.2 Cross-Adaptive Methods

The most essential concept of modern approaches to automating multi-channel instrument mixing is to consider the signal characteristics in terms of how they relate to the other tracks in the ensemble. The simplest form of this concept is side-chain processing. In side-chain processing, features from one track (energy/loudness) are used to control the processing applied to that same track or a different target track. One very common use of side-chaining is to *duck* the bass to the kick drum in a rock or dance mix. Section 2.1.1 will show that the low frequency content in the kick drum and bass causes masking and results in reduced clarity of each instrument. Due to the transient nature of the kick drum, the bass signal is lowered in volume when the kick drum is played. A compressor is applied to the bass signal, using the analysis of the kick drum to control the effect. The end result is that the bass volume is reduced during the attack of the kick drum then rises back to its initial level. This reduces the masking affect the bass has on the kick drum and allows it to *cut through* and be more prevalent in the mixture.

Cross-adaptive processing of multi-track mixtures extrapolates the side-chain concept to the mixture as a whole. In this architecture, features such as energy, loudness and spectral content are computed on each input source and compared to both other individual sources and the mixture as a whole. This is very similar to the process the mixing engineer employs as described in Section 2.1.2.

The basic architecture of a cross-adaptive mixing system is depicted in Figure 2.12. Each track is analyzed individually, producing a desired feature set that is informative for a target goal. If the goal is to determine gain levels for each track, features such as RMS energy (multiple time scales) and the frequency spectrum will be passed to the cross adaptive analysis block. Here, the features will be compared across tracks using psychoacoustic principles of loudness and masking as well as encoded information about general audio engineering practices. A system for live mixing is constrained by real-time computation concerns but there is no reason an offline system cannot perform multiple

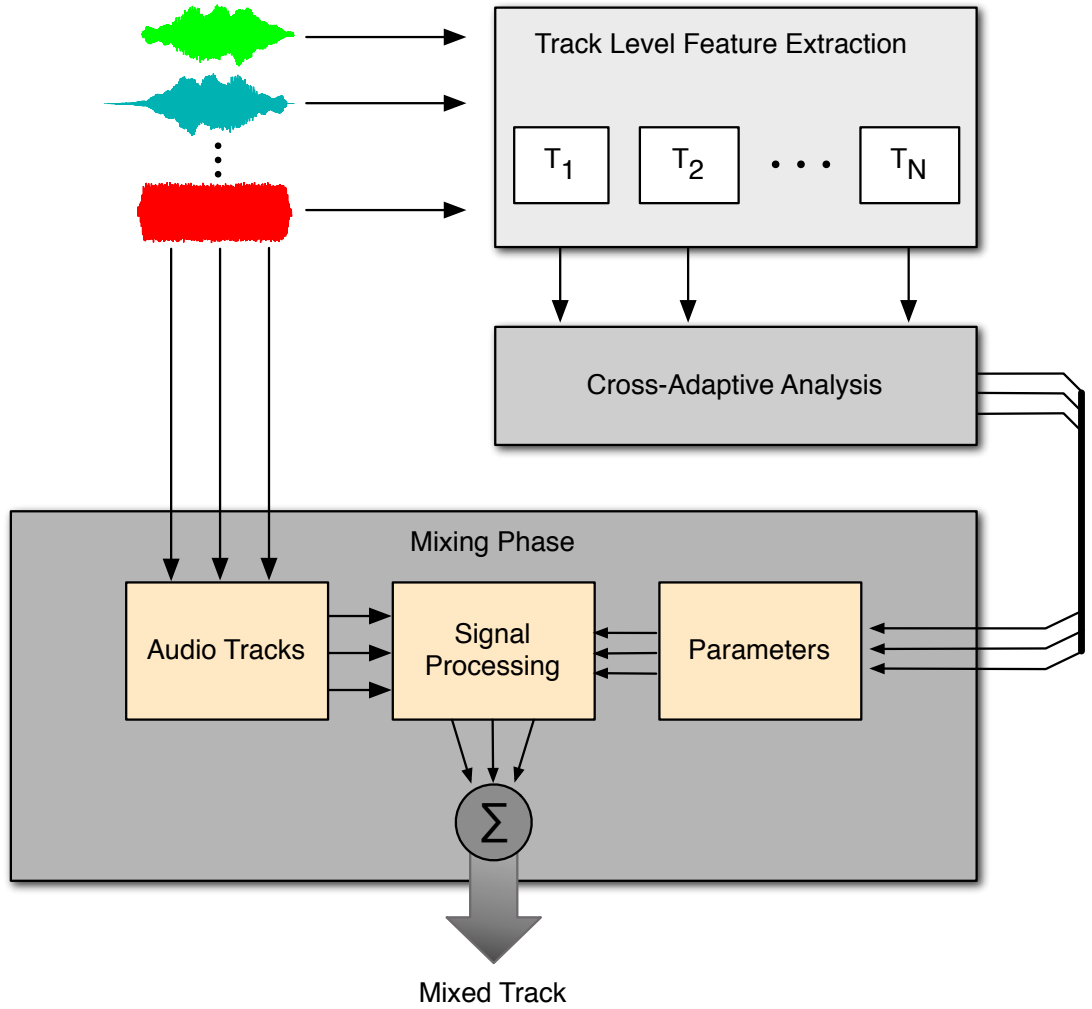


Figure 2.12: Signal flow and architecture of a cross-adaptive mixing model.

passes using the mixing model, hoping to converge on a resulting mixture that no longer requires processing based on the constraints of the model.

Reiss *et al.* have done extensive work in developing real-time mixing systems for levels, equalization, dynamic range compression and panning [3, 46, 48, 49, 58, 59, 61, 62, 63, 72, 95]. In addition to developing the cross-adaptive approach in Figure 2.12 they have conducted structured listening tests to evaluate the performance of their systems. In comparisons between unmixed audio, manually mixed audio and automatically mixed audio, their methods reliably outperform the unmixed audio and consistently approach or even surpass the mixes created by trained engineers.

The general formulation for applying effects in a multi-channel audio scenario is as follows

$$mix_l[n] = \sum_{m=0}^{M-1} \sum_{k=0}^{K-1} c_{k,m,l}[n] * x_m[n], \quad (2.4)$$

where x_m are individual audio channels and a set of control vectors denoted by $c_{k,m,l}$ represent various processing operations depending upon how c is defined. If the control vectors are scalars, this is an application of a gain coefficient, for delay, the control vector becomes a delay operator. For filtering and equalization, c becomes an impulse response that is convolved with the time domain signal.

In [48], the authors present a system that seeks to normalize the perceived loudness of each element in a multi-track mixture. The loudness is determined according to the ITU 1770 method in the EBU R-128 recommendation discussed in Section 2.1.1. Loudness levels are computed on a per track basis and a noise gate is used to determine whether there is activity on each track or if the noise floor is the primary signal. The loudness values are smoothed (low pass filtered) over time to prevent transients from having a pronounced and prolonged effect on the system parameters. The output fader parameters are also filtered to prevent artifacts. If the fader values change too rapidly, the system resembles a dynamic range compressor rather than a gain control system. A listening test was conducted that had participants rate the equality of the perceptual loudness of each individual instrument in the mix on a scale of 0-100 as well as the overall quality of the mix. The system performed well with the automatically generated mix significantly outperforming the unmixed audio.

This approach is extended in [95] by Ward *et. al* where a partial loudness model is incorporated to account for the frequency masking phenomenon when there are multiple sources present. Most experiments on perceptual loudness involve measurements of individuals responses to isolated pure tones or complex sources. In [53], Moore *et. al* explore the affect of having multiple audio sources on the perceived loudness of a target source. Ward incorporated this method of modeling partial loudness into a previous automated fader control algorithm. Masked and unmasked loudness levels are computed on both short-term and long-term scales and a correcting gain coefficient is computed. This is an iterative process, where the normalized tracks are then used as input to the system. The system will converge to a state where the corrective gains are below a given threshold and the loudness normalization process is complete.

An automated stereo pan positioning system is described in [49]. The objectives of the system

are to achieve spatial and spectral balance by analyzing the loudness and frequency content of an instrument signal and applying appropriate panning rules based on the analyses. The system applies constraints that low frequency sources should be centered and signals should be panned further from center proportional to the amount of high frequency content they contain. To accomplish this, a user defined panning width which determines the maximum stereo spread and the spectral centroid are used and mapped using either a linear, logarithmic or custom mapping function. A perceptual listening test to evaluate the effectiveness of the system was performed. The mixes generated by professional audio engineers fared better in user ratings for overall preference, and appropriate use of stereo mixing. However the automatic system performed consistently across multiple genres and would occasionally outperform one of the less experienced engineers.

In addition to the cross-adaptive mixing methods presented above, another new direction of research involves evaluating the perceptual differences of DSP effects in regard to semantic labels used in the audio engineering field.

2.2.3 Relating Perceptual Terms to Audio Effects

Mixing engineers and musicians use a wide variety of terms to describe sound and timbre [76]. In the context of a recording or mixing session, the conversation between musicians and the recording/mixing engineer will often use such terms in an attempt to hone in on a desired tone or timbre. Several works attempt to link high level descriptive terms like *bright*, *muddy*, *metallic* and *warm* with parameters of audio effects that manipulate the sound [74, 55, 71, 75, 69].

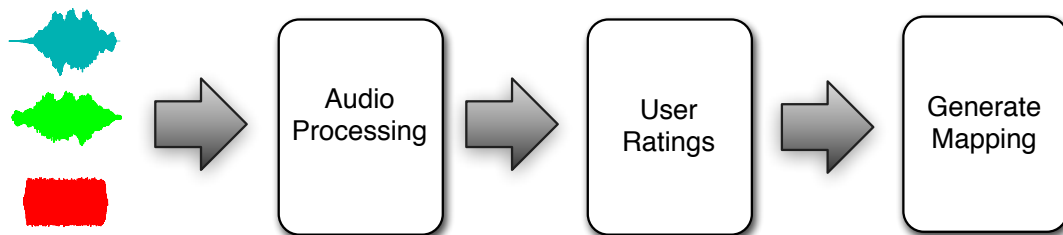


Figure 2.13: Modeling procedure to relate words that describe sound and timbre to audio signal processing parameters.

The general framework for developing these systems relies on gathering perceptual ratings of modifications to audio and recording the parameters used for modification. In [75], Sabin *et. al* attempt to find a personalized method for applying equalization curves to input audio. Not only is there disagreement as to the degree to which an individual describes a sound with a given semantic descriptor (bright/warm/tinny), but the same operation performed on different audio sources can induce perceptual responses. If a boost to midrange frequencies brightens one in instrument, it could make another sound tinny or boxy.

Through applying a variety of equalization curves to audio examples and recording user ratings of semantic audio descriptors in response to the processing they were able to learn which frequency bands effected which terms using regression techniques. From this a personalized equalizer was developed that allowed a user to increase the brightness based upon the learned preference of the user. This was verified by listening tests that found the automatically generated curves closely correlated with manually generated curves by the user. This method was extended to use transfer learning and active learning in [55] to require significantly less examples of user input to associate a characteristic curve with the audio descriptors.

A similar experiment was performed to map words like *bright*, *clear*, and *boomy* to different reverberation settings applied to audio samples. Whereas filters can be intuitively described by magnitude response curves, the same intuition is lacking in the impulse response of a reverb. They specify several metrics that characterize reverberation to semantic descriptors and find that although the audio measures differ significantly between users, their agreement with the perceptual ratings is high. This indicates that the system learns a perceptually relevant model on a per user basis.

These experiments rely on individual determination of how a specific sound or instruments ‘sounds’. The next section discusses the concept of timbre, what makes a specific instrument or sound the way it does and how humans process and organize audio using the concept of timbre. Additionally, methods for modeling timbre computationally are presented.

2.3 Perception and Modeling of Timbre

The American Standards Association [1] defines timbre as “that attribute of auditory sensation in terms of which a listener can judge that two sounds similarly presented and having the same loudness of pitch are dissimilar.” This definition is rather vague, problematic and controversial insofar as it does not actually say anything about what timbre actually *is*. From this, all we know is that timbre

is not pitch and timbre is not loudness. Finding a definition for timbre and the dimensions that define it is an active field of research and no agreed upon definition has yet been coined. This creates complications in accurately describing what timbre is, let alone modeling timbre computationally.

Timbre is often associated with a single sound source or instrument. Humans are very adept at identifying the source instrument when presented with a sound they have heard before. Not many individuals would confuse a trumpet with a piano, however, timbre is not simply defined by the source of a signal. The sound of rapping on the exterior of a piano may sound very similar to knocking on a wooden table. Many other examples can be imagined where instruments played in non-traditional manners would be difficult to identify. Perceptual phenomena such as pitch and loudness are understood to result in part from underlying physical phenomena of the human auditory system. Timbre is a multidimensional property whose very dimensions are still debated in the research community. There is, however, general agreement that the temporal and spectral envelopes play a significant role in determining if two audio signals ‘sound similar’. What follows is a summary of previous research in timbre, the role it plays in perceiving sound sources and what methods of modeling timbre have been explored.

2.3.1 Timbre Perception

In multi-timbral mixtures, each instrument contributes to the mixture in the dimensions of space, time and frequency. Two important elements that are often cited as being essential to the perception of timbre are the spectral envelope and temporal envelope. The temporal envelope represents the overall energy of the signal over time. The curve is generally divided into four sections that describe the components of the envelope. The attack, decay, sustain and release (ADSR) portions of a sonic event can exhibit significantly different characteristics depending on the source that produced the sound. The temporal envelope and its sections are detailed in Figure 2.14.

The attack portion of the temporal envelope describes the rise time of the amplitude of the signal. In general, the attack can be either sharp (fast) or soft (slow) with quick attacks typically being associated with percussive instruments or tonal instruments that are excited by an impulsive event (e.g. guitar, piano). Soft attacks usually occur when a sound is produced via a sustained excitation (e.g. bowed strings, woodwinds, brass). Decay refers to the transition from the attack portion to the sustained, or steady-state, section of the note where the amplitude remains relatively consistent. The release denotes the rate at which the event progresses from steady-state to silence. Although instruments often have particular ADSR envelope characteristics that are associated with them,

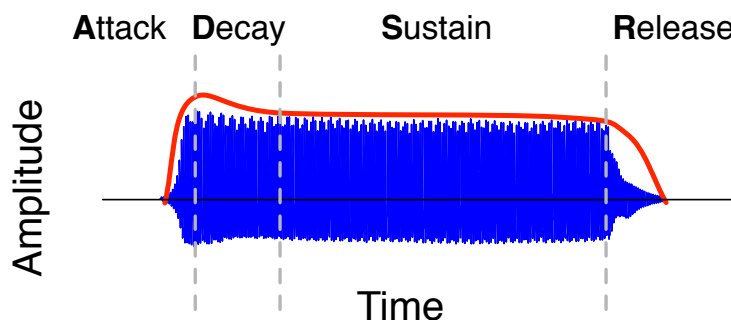


Figure 2.14: Attack, decay, sustain and release segments of the ADSR envelope.

sounds created using synthesizers or software commonly allow the ADSR envelope to be explicitly described and modified. Equally as important (if not more important) is the spectral envelope of a sound source. For harmonic sounds this is often described as the dB per octave rolloff of the harmonic amplitudes. For non-harmonic sounds, the spectral envelope is simply the general contour or shape of the magnitude frequency spectrum. For harmonic sounds produced by instruments, this spectral shape remains fairly stable through the range of the instruments. One interpretation is that an instrument has a spectral envelope that is sampled by the fundamental frequency and its associated harmonics, with the general contour remaining steady as the frequency content changes.

Figure 2.15 shows temporal and spectral envelopes for a bass guitar, cello, kick drum and snare drum. Comparing these plots reveals significant information about the characteristics of each instrument. The bass and cello have similar fundamental frequencies but differ significantly in both their temporal and spectral envelopes. The bass in (a) has a sharp attack followed by a slow release and an unclear sustain portion. The release portion rolls off fairly quickly as the finger is released from the string. The cello (c) has a much longer attack which is proportional to the release. The sustain is also not well defined in the cello and the decay is much more extended than the bass guitar. Note the difference in time scale between (a) and (c). The spectral envelopes in Figure 2.15 (b) and (d) exhibit similarities since both instruments possess a significant amount of low frequency content. Note that the high frequency rolloff in the bass is more pronounced than in the cello due to the higher harmonic content present in the cello signal.

The kick drum (f) envelope is comprised mostly of low frequency, experiencing over 40 dB of rolloff before it reaches 500Hz. The snare drum lacks very low frequencies and exhibits a quick rolloff up to 500 Hz and then a slow decline in energy to around 2000Hz before it flattens out. Although

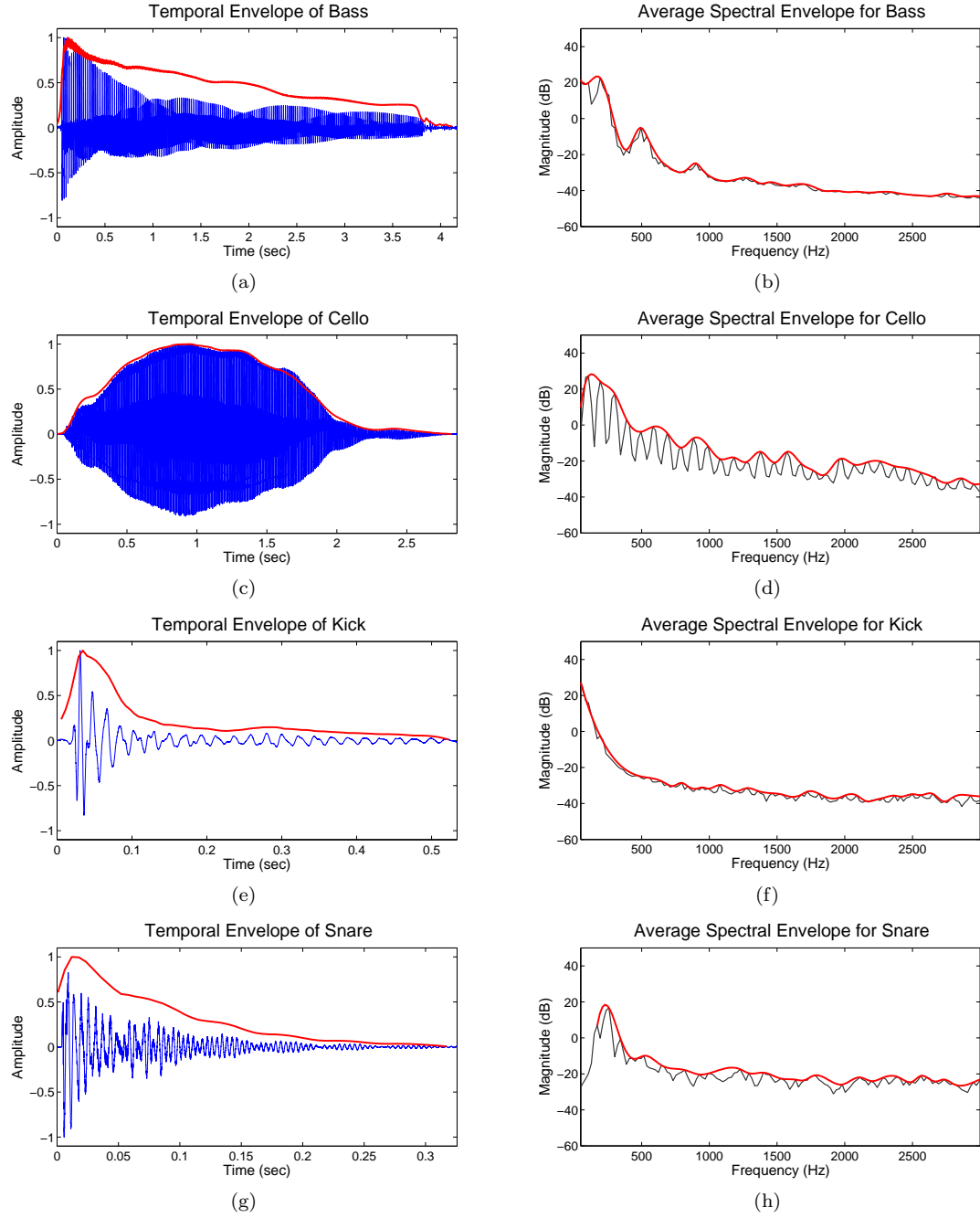


Figure 2.15: Temporal and spectral envelopes for bass guitar (a-b), cello (c-d), kick drum (e-f) and snare drum (g-h) .

the kick drum (e) and snare drum (g) possess many more similarities in their temporal envelopes than the bass and cello, the decay between the drums differs significantly. The energy of the kick drum dissipates much more quickly than the snare drum.

Experiments exploring the role of timbre in identifying sounds show disagreement between results from different researchers [29]. In several early experiments, identification of sounds was more difficult for subjects when the attack segment of a tone was removed versus when the decay segment was removed [5, 14, 21]. Investigating the role of melodic context, Campbell used a set of two-note legato phrases on six instruments [10, 11]. The transitional period between the two notes was varied between 20ms and 110ms. Subjects found the longer ‘legato transients’ to be more informative than the attack, steady state or shorter transient segments. In an experiment comparing the role of the steady state and transient across single note and legato musical phrases, Kendall found that the attack and legato transients were not as significant as in previous research [41]. Much of this disagreement can be attributed to the definitions of attack, decay, steady state and transient. At the time, these terms were not defined in a quantifiable manner and led to unfair comparisons between experimental results.

More recently, Hajda attempted to provide more formal definitions for the ADSR envelope segments using characteristics of the overall energy and average spectral content. This method, the Amplitude/Centroid Trajectory (ACT) bases the segments on the first derivative and global and local maxima/minima of the RMS amplitude and spectral centroid values. Results showed that the salience of the attack and transients versus the steady state depended upon whether a tone was impulsive (e.g. plucked strings, piano, marimba) or continuant (e.g. bowed strings, flute, clarinet). It was also found that when continuant instruments were played in a staccato manner, the attack and transient was more salient than the steady state due to the extremely short duration and rapid decay resulting from the staccato performance.

Work on obtaining semantic descriptors for decomposing the multidimensional aspects of timbre into its component parts has yielded fairly consistent results. Common methods for achieving this goal are semantic differential analysis and variation verbal attribute magnitude estimation (VAME). The former involves subjects rating where a sound lies on a scale whose extremes are polar opposites, such as ‘brightness’ and ‘dullness’. VAME uses semantic descriptors and their negation (bright/not bright) as the labels for the ends of the scale. A dimensionality reduction technique such as Factor Analysis (FA) or Principal Components Analysis (PCA) is often applied to determine the most salient descriptors for timbre. Complications arise in this method due to the subjective nature of the descriptions and overlap in subjects’ association with the terms. Nevertheless, many of these studies find similar descriptors as the most salient dimensions over a variety of data sets. Common perceptual axes are brightness, luminance, texture and fullness, relating to the following semantic

descriptors: bright, dull, sharp, full, warm, harsh, thin and nasal.

Multidimensional Scaling and Dimensionality Reduction

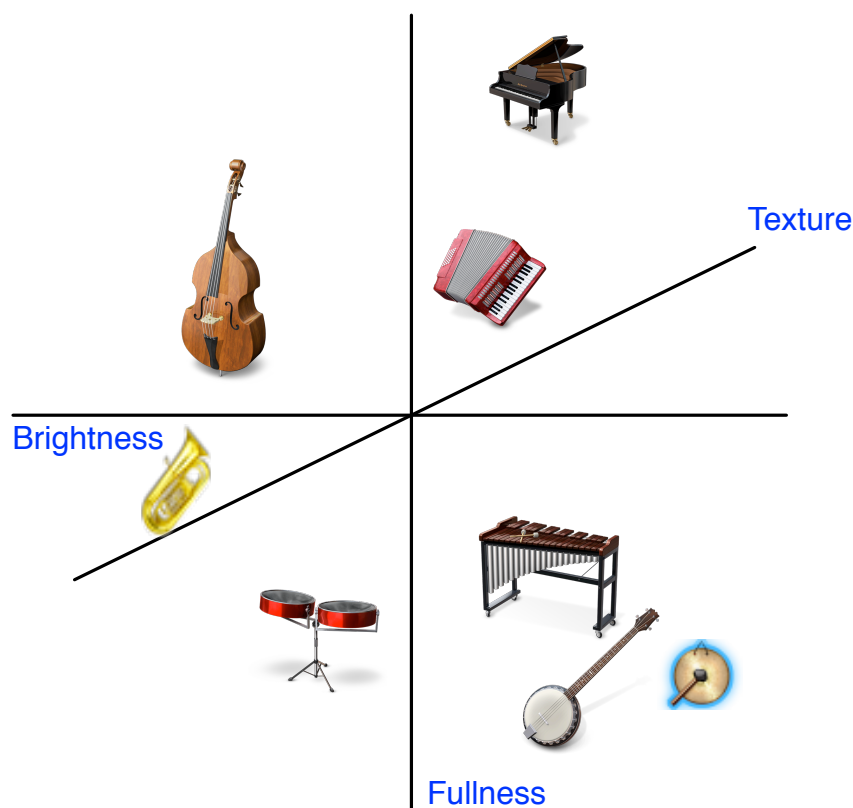


Figure 2.16: An example timbre space resulting from multidimensional scaling.

A significant portion of research in the perception of timbre has focused on multidimensional scaling. The basic framework for these experiments relies on collecting perceptual data about the sounds and performing some transformation to arrange the sounds in a geometric space. The goal is that sonic events that are perceived as similar will be closely grouped and sounds that are perceived as dissimilar will be farther apart in the space. The perceptual data collected is a pairwise similarity comparison between two sounds. Once every possible pair has been rated, the data is transformed to span the space so that the distances are preserved in the lower dimensional representation (usually

two or three dimensions). One significant difference between finding salient descriptors using VAME with dimensionality reduction and MDS is that the dimensions in MDS must be intuitively extracted by the researcher. If two dimensions are chosen for the representation, qualitative analysis of the groupings in the space is the only way to reveal what the individual components are. An example of a result obtained from MDS is shown in Figure 2.16. In this result, the y-axis is related to the spectral energy distribution, the x-axis corresponds to the onset-offset patterns of tones and the azimuth relates to the temporal evolution of the attack portion of the tone’s envelope.

Other experiments have found very similar results both repeating the semantic MDS task as well as using other methods of dimensionality reduction to produce a timbre space with perceptually relevant axes [98, 42, 89, 30, 9, 97, 50].

Modeling Timbre

In addition to the music perception work on describing timbre using semantic descriptors and MDS, much of the timbre perception research has focused on analysis by synthesis. In these experiments, tones are synthesized digitally based on characteristics of the spectral and temporal envelopes. This method allows for much more tightly controlled experiments but suffers from a lack of realism in the audio presented to the subjects. Some authors even describe that the participants had trouble differentiating between their opinion of the sound they were listening to versus their memory of the instrument that the synthesized tone is approximating.

Further research tries to show correlations between the space derived from multidimensional scaling and features computed directly from the audio or spectrum [32]. Multivariate-regression techniques and self-organizing maps (SOM) are employed to determine features that have high correlation with the organization of the timbre space. From these experiments, features such as spectral centroid, spectral flux and spectral irregularity in addition to others were shown to be correlated with salient dimensions of timbre. A list of features commonly used in timbre analysis and synthesis are detailed in Table 2.2.

Beyond the simple spectral features used for sound synthesis, much of the work on modeling timbre computationally revolves around the Music Information Retrieval (Music-IR) community and specifically the instrument recognition and song similarity tasks. Instrument recognition systems leverage machine learning methods to represent the underlying structure in the data given a set of features and instrument labels. Building upon the significance of the spectral envelope from the music perception literature, many instrument recognition systems rely upon some type of spectral

Feature Name	Description
Mean Coefficient of Variation	Average variation of spectral components
MFCCs	Approximation of the spectral envelope
Spectral Contrast	Estimation of the harmonicity of the signal
Bandwidth	The frequency range present in the signal
Centroid	Center of mass of the spectrum (brightness)
Flux	The change in energy from the previous frame
Rolloff	Frequency below which X% of energy lies
Zero-Crossing Rate	Number of zero-crossings in time domain signal
Band Energy Ratio	Ratio of energy between two filterbank channels
Sub-Band Features	Features computed on filterbank channels

Table 2.2: Common features extracted for timbre analysis.

envelope feature.

Two main approaches involve models that capture the dynamic information of each example and models that represent the global statistics of the sounds. Methods that account for the global statistics include K-Nearest Neighbors (K-NN) as well as kernelized K-NN, Naive Bayes (NB), Decision Trees (DT), Neural Networks (NN), Support Vector Machines (SVM) and Gaussian Mixture Models (GMM) [22, 32]. Work that models the temporal evolution of the features uses Hidden Markov Models (HMM), and Gaussian Processes (GP)[57, 85]. The next sections detain examples of such systems.

2.3.2 Modeling Global Timbre

Aucouturier *et al.* seek to develop a quantitative model of polyphonic timbre and complex instrument textures. Their approach is to describe the timbre of a song as a whole rather than attempt to decompose the signal into its separate sources and model the timbre of the resulting individual instruments. Rather than describe the timbre of a song as *acoustic*, *crisp* or *muddy*, the goal of this procedure is to determine the similarity of two songs based on timbre. The overall system of the proposed method is outlined below:

- Divide the signal into overlapping frames and multiply by a window function
- Compute a feature vector for each frame
- Use the feature vectors across all frames to develop a statistical model of timbre
- Compare timbre models to determine whether two songs sound similar

Feature Extraction

Many instrument recognition systems use the spectral envelope as a means to classify a given audio sample. The authors suggest that the spectral envelope maintains a relatively steady shape over a short time for a mixture of instruments. Figure 2.17 shows the spectral shape for five seconds of the song *Eleanor Rigby* by The Beatles. The plot shows the spectrum of the audio at different time instances and the basic spectral shape for the whole clip is depicted as the thick red line. The lighter lines indicate the beginning of the audio clip and as time increases the lines become darker. A feature that provides an approximation of the spectral envelope is the widely used mel frequency cepstral coefficients (MFCC). MFCCs provide a good, compact approximation of the spectral envelope and are frequently used throughout the literature for speech/speaker recognition and music information retrieval (Music-IR). The procedure for calculating MFCCs is outlined in Appendix A.

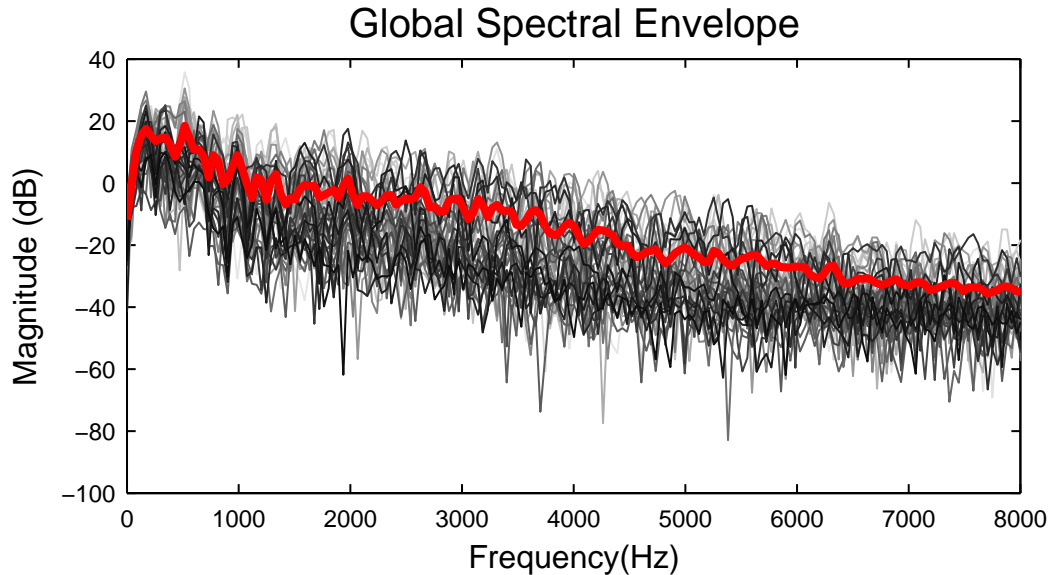


Figure 2.17: The change in the spectrum and overall shape of the spectrum (red line) for 5 seconds of audio.

Statistical Modeling

Since the goal is to recognize a statistically emergent shape, a mixture of Gaussians is used to model the feature data extracted from the audio. A Gaussian mixture model (GMM) models

the probability density associated with a data set as a wighted combination of individual Gaussian distributions as

$$p(F_t) = \sum_{m=1}^M \pi_m \mathcal{N}(F_t, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \quad (2.5)$$

where π_m are the mixture coefficients and F_t is the feature vector observed at time t . The parameters involved in the modeling process include the mean and covariance of each individual Gaussian component and the number of Gaussians used to model the data.

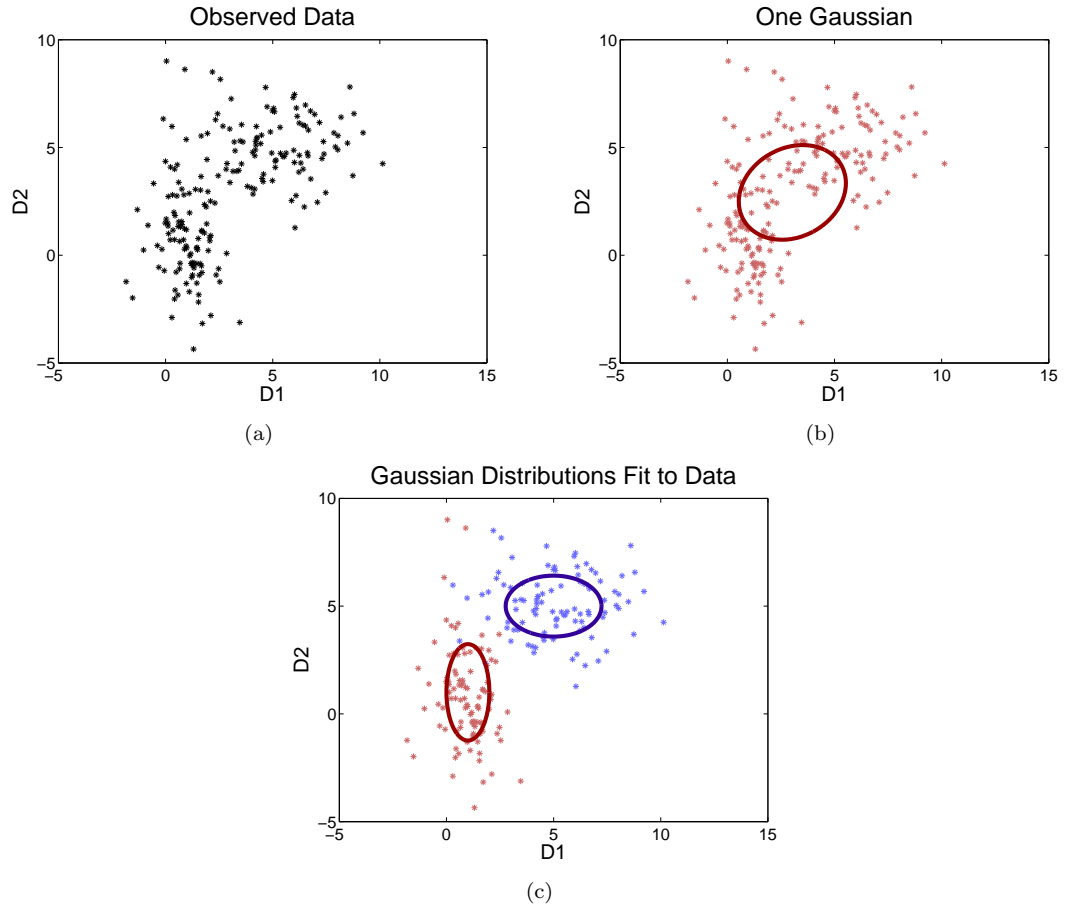


Figure 2.18: Unlabeled data set (a), one Gaussian (b) and two Gaussians (c).

Consider the data set shown in Figure 2.18(a). To represent this data with a multivariate Gaussian distribution, it suffices to find the empirical mean and covariance as shown in (b) where

the ellipse represents one standard deviation, σ , from the mean. This does not seem to fit the data very well as there are a significant number of outliers present. Modeling the data with two Gaussians, as in (c), there seems to be a better characterization of the data set. Using three Gaussians may also yield a satisfactory model, but as the number of Gaussians, m , increases, the possibility of over-fitting the data becomes significant.

In an unsupervised learning problem there is no information about what data points are related or how many classes are present. In order to model this data using a Gaussian mixture model, assumptions must first be made about the number of Gaussians present and the initial parameters associated with each Gaussian. The parameter estimates are calculated using the k -means algorithm where k is equal to the number of Gaussians to train, and the variance is assumed to be the distance to the closest estimated mean value. Using the same simple data set in Figure 2.18, and assuming two Gaussians, Figure 2.19 represents the k -means algorithm for parameter initialization.

The k -means algorithm needs a starting seed for the mean values, $\boldsymbol{\mu}_k$ (k clusters), from which to iterate and converge on an answer. This seed value can be chosen from a uniformly distributed random variable over the range of all possible values, a subset of random sample points from the data or a variety of other schema. Once the seed values are chosen, the distortion measure, J , is minimized first with respect to r_{nk} then with respect to $\boldsymbol{\mu}_k$ [6],

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \quad (2.6)$$

where

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg \min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\| \\ 0 & \text{otherwise.} \end{cases} \quad (2.7)$$

This equates to labeling each observed data point with the label associated with the closest mean value, then once all data points have been labeled, calculate the means of the clusters. The calculated means will be different from the initial guesses and each data point is again labeled with its closest mean value. The process iterates until convergence. Once the initial parameter estimates for each Gaussian of the GMM have been determined, the model is trained using the Expectation Maximization (EM) algorithm. Appendix B describes this process in detail.

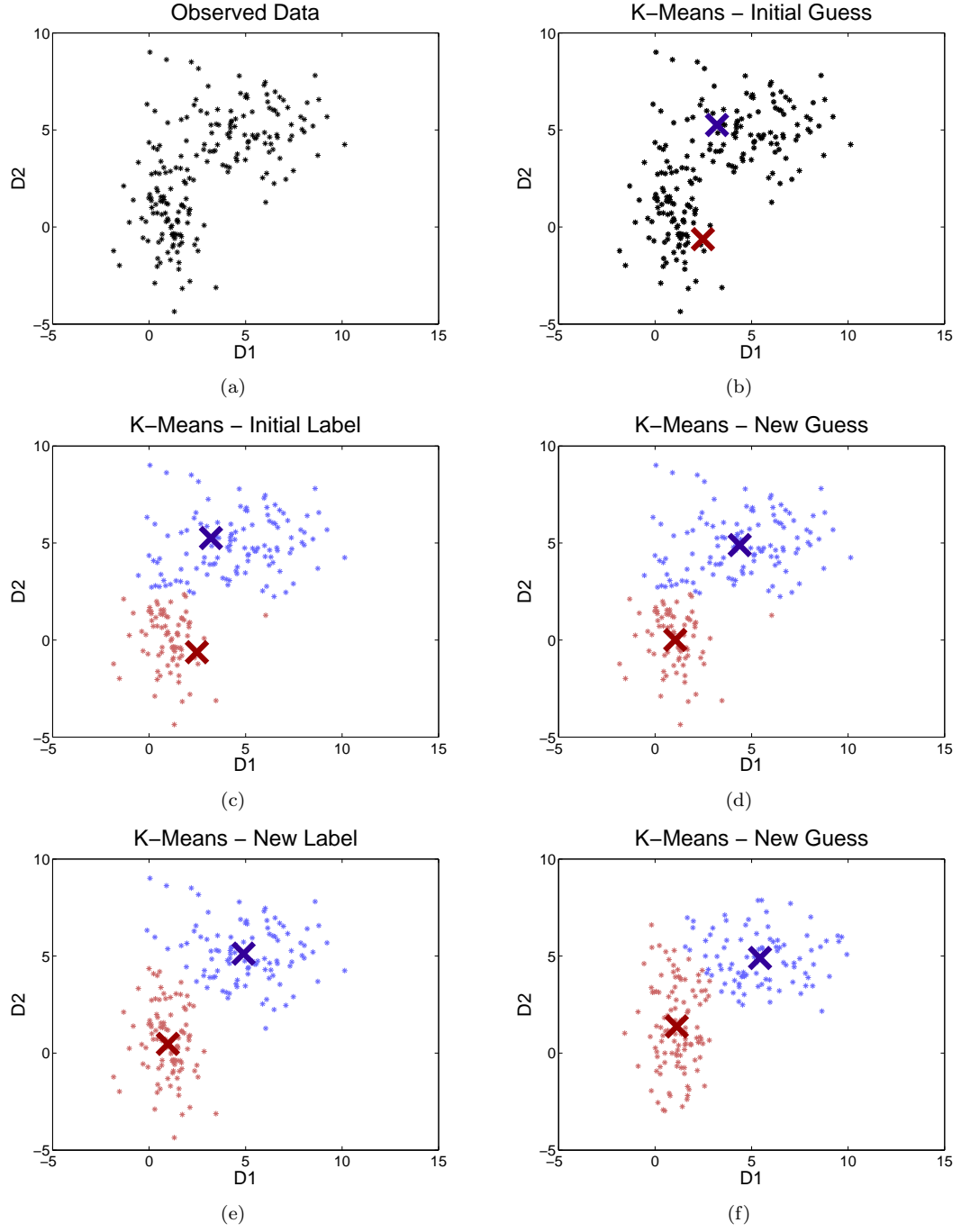


Figure 2.19: Unlabeled data set (a) and iterative k -means process (b)-(f).

Model Evaluation

A corpus of songs consisting of 350 titles from 37 artists of various genres was used for training and evaluation of the model. Aucouturier *et al.* state that the “songs were chosen in order to have

clusters that are *timbrally* consistent (all songs in each cluster sound the same)” [2]. They accomplish this by choosing songs by the same artist and same album then perform slight modifications to the grouping through subjective tests. In order to evaluate the distance measure in Equation 2.9, the number of songs in the same cluster closest to the test song is determined and compared to the total number of songs in the cluster. That is, for a cluster of size N_i , calculate the ratio of the number of songs closest to the song S_i from the same cluster as S_i , divided by the total number of songs in the cluster. This measure is known as R-precision.

$$p(S_i) = \frac{\text{card}(S_k | C_{S_k} = C_S)}{N_i} \quad (2.8)$$

The R-precision measure was used to find the optimal number of both MFCC coefficients and Gaussian distributions to model timbre. Iterating through the number of MFCCs and Gaussians in increments of ten from [10, 50] and [10,100] respectively, the authors found that $M = 50$ Gaussians and $N = 20$ MFCCs gave the best R-precision.

A sampling method is employed to compare the timbre models for two songs and evaluate their similarity. Given two songs, A and B, a large number of sample points, S^A , is taken from song A and the likelihood that they came from the model of song B is computed. The same is done for a sample of song B and the result is normalized. A value of $NS = 1500$ was found to be a large enough sample size for evaluation.

$$D(A, B) = \prod_{i=1}^{NS} \frac{P(S_i^A | A) P(S_i^B | B)}{P(S_i^A | B) P(S_i^B | A)} \quad (2.9)$$

This distance provides a quantitative measure of how similar the songs are within the context of the model, but to evaluate whether they sound the same a person must listen to them and make a value judgment based on their perception.

The authors give the example of a model query where a song title is entered and the n closest songs are located based on their timbre models. For the song “L’Instant de Vérité”, a jazz piano solo, many piano songs are returned from various genres including classical, jazz and musicals. Notably, the song “Singin’ in the Rain” was returned for this query which may seem like an unlikely candidate, however in music discovery and exploration systems the unexpected can be desirable since the goal of the listener is to locate new music based on the input song.

Continued Investigation

The selection of ground truth data for modeling timbre is difficult due to the subjective nature of the task. Aucouturier *et al.* employ the “same artist - same album” approach as was stated above. In post production of an album, global processing is often applied to the entire album in the mixing and mastering stages. Effects such as equalization and compression are applied to all the songs on the album using the same parameters for each song. In effect this is modifying the spectral envelope of every song in the same fashion. This phenomenon is known as the album-effect and has been shown to induce better than expected results in many Music-IR systems. Appendix C provides an investigation of this effect in relation to the timbre model presented in this section.

2.3.3 Dynamic Timbre Modeling

The previous section presents a model of the long term spectral statistics of a song to infer information about timbre. Music however, is a dynamic process and by nature changes over time. Burred *et al.* include temporal information in their model of timbre citing that not only does the spectral envelope have a significant impact on timbre perception, but the temporal envelope also considerably contributes to the sonic texture perceived by a listener [8]. They seek to model the timbre of individual instruments which would lend to instrument detection/recognition, source separation and sound synthesis. To this end they model the change in the spectral envelope over time as a Gaussian process.

Feature Extraction

As discussed previously, the spectral envelope is an informative feature to model timbre. To obtain an accurate representation of a specific instrument, a data set of many notes in the instrument’s range is necessary as well as various articulations and dynamics since the features may differ over the range of the instrument. In order to achieve this, the authors concatenate many note examples in time to develop their feature vectors. The prominent peaks (harmonic partials) of the spectrum are selected and tracked from frame to frame.

Given that the notes of the training examples are known, a fixed number of partials ($p = 20$) is extracted for each note. Since different notes are concatenated to form the feature vectors, this introduces the problem of properly representing them in matrix form. One device employed, Partial

Indexing (PI), simply places the amplitude of each partial in a row of a data matrix, \mathbf{X}

$$\begin{array}{ccc}
 A4 & C5 & E5 \\
 \left[\begin{array}{ccc|ccc}
 A_{r_1}(1760) \dots A_{r_{N_1}}(1760) & A_{r_1}(2092) \dots A_{r_{N_2}}(2092) & A_{r_1}(2636) \dots A_{r_{N_3}}(2636) \\
 A_{r_1}(1320) \dots A_{r_{N_1}}(1320) & A_{r_1}(1569) \dots A_{r_{N_2}}(1569) & A_{r_1}(1977) \dots A_{r_{N_3}}(1977) \\
 A_{r_1}(880) \dots A_{r_{N_1}}(880) & A_{r_1}(1046) \dots A_{r_{N_2}}(1046) & A_{r_1}(1318) \dots A_{r_{N_3}}(1318) \\
 A_{r_1}(440) \dots A_{r_{N_1}}(440) & A_{r_1}(523) \dots A_{r_{N_2}}(523) & A_{r_1}(659) \dots A_{r_{N_3}}(659)
 \end{array} \right]
 \end{array}$$

where $A_{rn}(F_p)$ indicates the amplitude of the frequency of the p th harmonic in the r th frame and N_1, N_2 and N_3 indicate last frame of each note. In other words, the bottom row is the fundamental frequency, f_0 , of each note (A4, C5, E5) for a given frame of audio and each row above represents a harmonic (multiple of f_0). The problem inherent in the partial indexing method is that each row contains amplitudes of harmonic partials that are located at different frequencies as indicated above, in effect misaligning fundamental frequency-invariant features in the data, that is features that occur at the same frequency regardless of the note being played.



Figure 2.20: System diagram of the spectro-temporal envelope extraction process.

A modified method involves performing Envelope Interpolation (EI) to align the frequencies in the data matrix. In this method, the partials are extracted and tracked over time as in the previous method, then the spectral envelope is approximated by interpolating between each partial. The interpolated function is then sampled at G regular intervals across a given frequency range. This process is depicted in Figure 2.21. In matrix form this means that the columns of \mathbf{X} all contain amplitude values that span the same frequency range as opposed to the PI method where each column spans a different frequency range.

Once the matrix containing the spectro-temporal envelope has been computed, Principal Components Analysis (PCA) is performed to reduce the dimensionality of the data. After mean centering and variance normalization, the resultant projection is

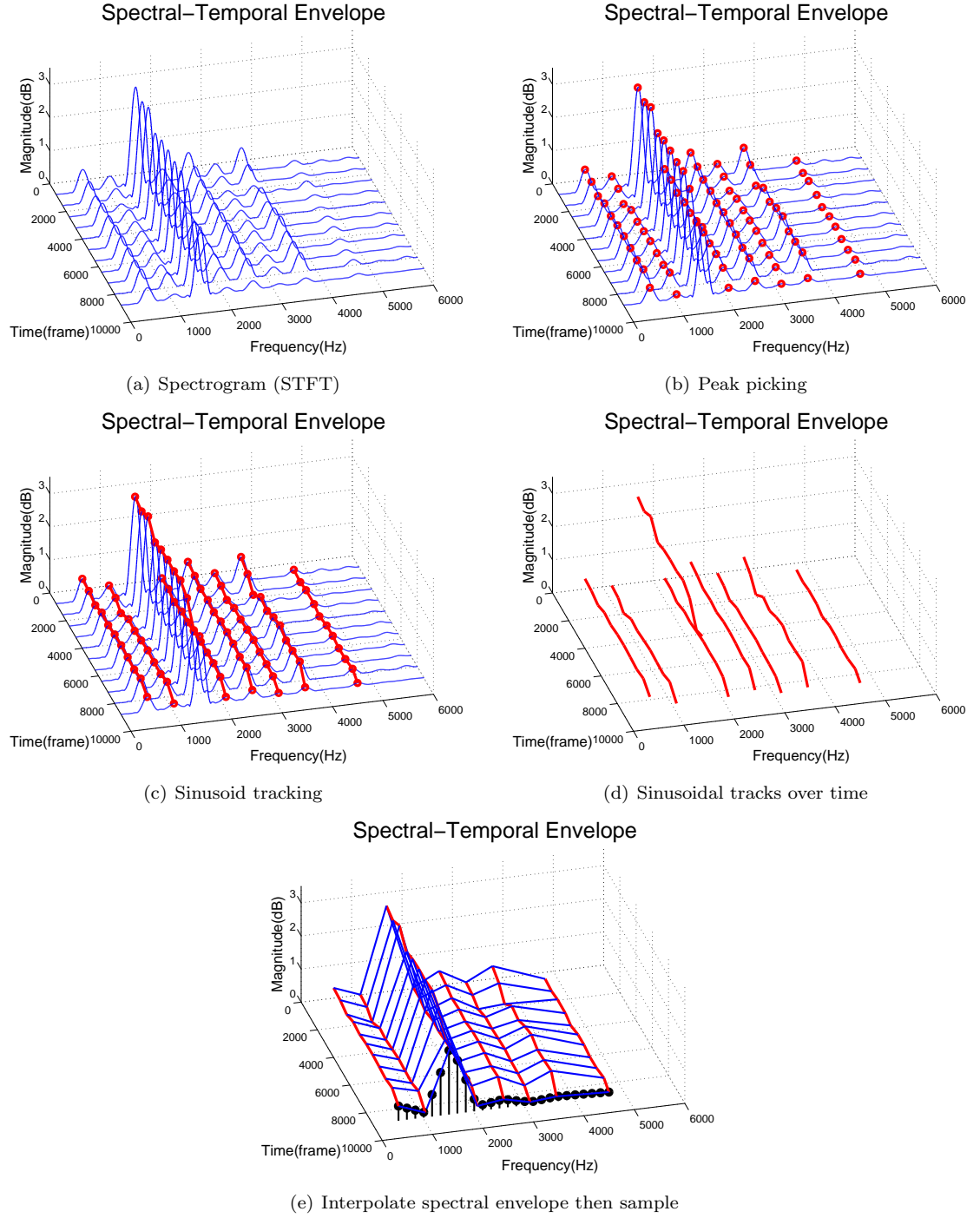


Figure 2.21: System overview for spectro-temporal envelope extraction.

$$\mathbf{Y}_\rho = \mathbf{\Lambda}_\rho^{-1/2} \mathbf{P}_\rho^T (\mathbf{X} - E\{\mathbf{X}\}) \quad (2.10)$$

where $\mathbf{\Lambda}_\rho = \text{diag}(\lambda_1, \dots, \lambda_D)$ is a diagonal matrix consisting of the D largest eigenvalues of the covariance matrix $\mathbf{\Sigma}_x$, and \mathbf{P}_ρ contains the corresponding eigenvectors (ρ indicates reduced dimensionality).

Statistical Modeling

In order to preserve the essential time information in the data the authors choose to model the frequency vectors as the result of an underlying random process. Therefore, each instrument is modeled as a trajectory of the feature vectors. The resulting trajectories for each sample of an instrument are combined to represent a single instrument prototype curve. The trajectories must be of the same length to do this, which requires interpolating the length of each individual trajectory to be the same as the length of the longest sample trajectory. If R_{max} is the length in frames of the longest trajectory, then all other trajectories are interpolated to have length R_{max} . Every point in the prototype curve treated as a Gaussian random variable, $\mathbf{p}_{ir} \sim \mathcal{N}(\boldsymbol{\mu}_{ir}, \mathbf{\Sigma}_{ir})$ with empirical mean and covariance matrix given by

$$\begin{aligned} \boldsymbol{\mu}_{ir} &= \frac{1}{S_i} \sum_{s=1}^{S_i} \tilde{\mathbf{y}}_{sir} \\ \mathbf{\Sigma}_{ir} &= \text{diag}(\sigma_{i1}^2, \sigma_{i2}^2, \dots, \sigma_{iR_{max}}^2), \quad \sigma_{ir}^2 = \frac{1}{S_i - 1} \sum_{s=1}^{S_i} (\tilde{\mathbf{y}}_{sir} - \boldsymbol{\mu}_{ir})^2 \end{aligned} \quad (2.11)$$

Hence each $C_i = (\mathbf{p}_{i1}, \mathbf{p}_{i2}, \dots, \mathbf{p}_{iR_{max}})$ prototype curve represents a D -dimensional non-stationary Gaussian random process whose mean and covariance changes over time

$$C_i \sim GP(\boldsymbol{\mu}_i(r), \mathbf{\Sigma}_i(r)) \quad (2.12)$$

where C_i is the curve of the i th instrument and r indexes time. The database used to generate the curves consisted of 423 total song files for the five instruments. Two or three instruments of each instrument type were played at three dynamic levels - piano (soft), mezzo-forte (moderately loud), forte (loud) - covering a range of one octave from C4 to B4.

The first dimension in the PCA space corresponds to the overall spectral shape and energy which agrees with the assumption that the spectral envelope is a determining factor in human perception of timbre. The trace along the second dimension illustrates a trade-off between high frequency energy and low frequency energy, creating a point about which the ratio of high to low frequency content

pivots.

Model Evaluation

To evaluate the efficacy of the model, an instrument classification task is performed to assess the model’s ability to differentiate between various timbres. A total of 1098 instrument samples over all five classes (trumpet, piano, clarinet, violin and oboe) and encompassing the same dynamic range as used in the model development comprised the data corpus for this task. In order to compare the curve extracted from the test sample to the prototype curve, the test sample must also be interpolated to be the same length as R_{max} . The test sample is classified according to the average Euclidean distance between its mean points and the mean points of the i instrument prototype curves, C_i ,

$$d(\tilde{U}, C_i) = \frac{1}{R_{max}} \sum_{r=1}^{R_{max}} \sqrt{\sum_{k=1}^D (\tilde{\mu}_{rk} - \mu_{irk})^2} \quad (2.13)$$

where $\tilde{\mu}_{rk}$ is the interpolated mean vector for the r th frame of the test sample. The averaged results of the ten-fold cross validation classification task are shown in Table 2.3(b). This shows that the partial indexing approach, which does not align frequency in the data matrix is outperformed by the envelope interpolation process indicating that fundamental frequency variant features are more important in describing timbre than f_0 -invariant features.

A polyphonic instrument recognition experiment was also performed to evaluate the effectiveness of the model to capture elements of timbre. This experiment again used 1098 samples, using 66% of them as the training data and the remaining 33% were used to develop 100 mixtures of instruments. Mixtures involving only one pitch from each instrument class are denoted as *simple mixtures* and test samples with more than one note per instrument are *complex mixtures*.

An onset detection phase was introduced in which new partial tracks that occurred signified the beginning of a note. This is necessary to ensure that the test trajectory corresponds to a single note and does not overlap multiple notes. In addition to the Euclidean distance measure, a likelihood approach was used where the maximum probability that a note came from a particular instrument was found. The results of the polyphonic instrument recognition test are shown in Table 2.3(a).

From the research presented in Sections 2.3.2 and 2.3.3 we see that the spectral envelope and the manner in which it evolves over time are central to similarity in the perceptual realm as well as computational tasks such as instrument recognition. This knowledge and results presented next

(a) Polyphonic classification results					
	Simple Mixtures			Complex Mixtures	
Number of Instruments	2	3	4	2	3
Euclidean Distance	68.48	52.25	41.28	64.66	50.64
Likelihood	73.15	55.56	54.18	63.68	56.40

(b) Monophonic classification results		
Method	Accuracy	STD
PI	74.9%	2.8%
EI	94.9%	2.1%

Table 2.3: Instrument recognition results for polyphonic and monophonic audio samples. [8].

from previous research serve as motivation for the experiments presented in later chapters.

2.4 Perceptual Feature Evaluation

The work presented here was originally published in [77] and later turned into a book chapter [79]. It serves as a basis for experiments presented in Chapter 7 that seek to find representations for timbre that correlate with human perception not just statistics of labeled data (genre/mood/tags). The work that follows is oriented specifically toward the domain of music emotion recognition (MER) but the same basic concepts and experimental design can be applied across various topics.

A musical piece is made up of a combination of different attributes such as key, mode, tempo, instrumentation, etc. While not one of these attributes fully describes a piece of music, each one contributes to the listener’s perception of the piece. These experiments hope to establish which compositional attributes significantly determine emotion and which parameters are less relevant. These parameters are not the sole contributors to the emotion of the music, but are within our ability to measure from the symbolic dataset we use in our experiments, and therefore are the focus of this study [39]. Specifically, we want to determine whether these compositional building blocks induce changes in the acoustic feature domain.

We motivate our experiments from findings that have been verified by several independent experiments in psychology [33, 73, 96]. When discussing emotion, we refer to happy versus sad temperament as valence and higher and lower intensity of that temperament as arousal [91]. Mode and tempo have been shown to consistently elicit a change in perceived emotion in user studies. Mode is the selection of notes (scale) that form the basic tonal substance of a composition and tempo is the speed of a composition [70]. Research shows that major modes tend to elicit happier emotional

responses, while the inverse is true for minor modes [17, 27, 28, 96]. Tempo also determines a user’s perception of music, with higher tempi generally inducing stronger positive valence and arousal responses [17, 27, 26, 73, 96].

2.4.1 Data Collection

In previous studies (such as [96]), several controlled variations of musical phrases are provided to each participant. Since we are studying the changes in the acoustic feature domain, we require samples that we can easily manipulate in terms of mode and tempo and that provide a wide enough range to ensure we are accurately representing all possible variations in the feature space. To this end, we put together a dataset of 50 Beatles MIDI files, attained online¹, spanning 5 albums (Sgt. Peppers, Revolver, Let It Be, Rubber Soul, Magical Mystery Tour). In order to remove the effect of instrumentation, each song was synthesized as a piano reduction and a random twenty second clip of each song was used for our labeling task.

Mechanical Turk Annotation Task

In order to annotate our clip pairs, we use the Mechanical Turk online crowd-sourcing engine to gain input from a wide variety of subjects [88]. In our Human Intelligence Task (HIT), we ask participants to label four uniformly selected song pairs from each of the three categories: original MIDI rendering, MFCC reconstructions, and chromagram reconstructions. For each pair of clips participants are asked to label which one exhibits more positive emotion and which clip is more intense. The three categories of audio sources are presented on three separate pages. The participants are always comparing chroma reconstructions to chroma reconstructions, MFCC reconstructions to MFCC reconstructions or MIDI renderings to MIDI renderings. Subjects never compare a reconstruction to the original audio. For each round, we randomly select a clip to repeat as a means of verification. If a user labels the duplicated verification clip differently during the round with the original audio, their data is removed from the dataset.

2.4.2 Experiments and Results

Our first set of experiments investigates the emotional information retained in some of the most common acoustic features used in Music-IR, MFCCs and chromagrams. As described above, users listen to a pair of clips that was reconstructed from features (MFCC or chroma) and rate which is

¹<http://earlybeatles.com/>

more positive and which has more emotional intensity. We seek to quantify how much information about musical emotion is retained in these acoustic features by how strongly emotion ratings of the reconstructions correlate with that of the originals. We first relate the user ratings to musical tempo and mode, and then we explore which features exhibit high variance with changes in tempo and mode or are invariant to altering these musical qualities.

Running the task for three days, we collected a total of 3661 completed HITs, and accepted 1426 for an approval rating of 39%, which is similar to previous work annotating music data with MTurk [45, 47, 88]. The final dataset contains 17112 individual song pair annotations, distributed among 457 unique Turkers, with each Turker completing on average ~ 2.5 HITs. With a total of 160 pairs, this equates to ~ 35.65 ratings per pair.

For each pair and for each audio type, we compute the percentage of subjects that rated clip A as more positive (valence) and the percentage that labeled clip A as more intense (arousal)

$$p_v = \frac{1}{N} \sum_{n=1}^N \mathbf{1}\{A_n = \text{HigherValence}\}, p_a = \frac{1}{N} \sum_{n=1}^N \mathbf{1}\{A_n = \text{HigherArousal}\} \quad (2.14)$$

where N is the total number of annotations for a given clip, p_v is the percentage of annotators that labeled clip A as higher valence, and p_a is the percentage of annotators that labeled clip A as higher arousal. For each song pair, we then compare the percentage of Turkers who rated song A as more positive in the original audio to those who rated song A more positive in the reconstructions, yielding the normalized difference error for all songs.

Audio Source	Normalized Difference Error	
	Valence	Arousal
MFCC Reconstructions	0.133 ± 0.094	0.104 ± 0.080
Chroma Reconstructions	0.120 ± 0.095	0.121 ± 0.082

Table 2.4: Normalized difference error between the valence/arousal ratings for the reconstructions versus the originals.

In Table 2.4, we show the error statistics for the deviation between the two groups. The paired ratings of each type are also verified with a paired Student’s t-test to verify that they do not fall under the alternative hypothesis that there is a significant change, but as we are looking for proof

that there is no change, average error remains the best indicator.

Relationships Between Musical Attributes and Emotional Affect

The general trend of major tonality being associated with positive emotional affect and higher tempo corresponding to an increase in arousal or valence was shown in previous research above. What follows is an analysis of the data for trends relating major/minor modes and tempo to valence and arousal.

The entire dataset S is divided into a subset $M \subset S$ that consists of pairs that contain one major mode song and one minor mode song, as well as a subset $T \subset S$ in which pairs differ in tempo by more than 10 beats per minute (bpm). For subset M , the percentage of users who labeled the major song as more positive and the percentage of users who label the major song as more intense is calculated. Similarly, for subset T , the tempo and intensity data are compared to the user ratings for valence and arousal. Looking at Table 2.5, the results are commensurate with the findings from the various psychology studies referenced in Section 2.4, namely that major songs are happier and faster songs are more intense.

Null Hypothesis	Agreement Ratio
Major Key Labeled as More Positive Valence	0.667
Faster Tempo Labeled More Positive Valence	0.570
Major Key Labeled as More Positive Arousal	0.528
Faster Tempo Labeled as More Positive Arousal	0.498

Table 2.5: Percentage of paired comparisons that yielded the desired perceptual result for mode and tempo.

One area where we expected larger agreement is the relationship between tempo and intensity. We only have the beats per minute for each song, and we label the faster song as the one with a higher bpm. The note lengths and emphasis in relation to the tempo are disregarded in this analysis and may be a source of uncertainty in the result. Depending upon the predominant note value (quarter/eighth/sixteenth), a slower tempo can sound faster than a song with a higher number of beats per minute. These are two different compositions, not the same clip at two different tempos.

This section provided a perceptual evaluation of emotional content in audio reconstructions from acoustic features. In addition, the findings agree with those of previous work showing correlation

between major keys and increased positive emotion as well as increased tempo and increased positive emotion and activity. For tempo, mode and key we have provided a variational analysis for a large number of acoustic features. This style of analysis is used later in Chapter 7 to show salience of feature representations for timbre and instrument recognition.

3. Methods, Models and Features

This chapter introduces the mathematical models and methods used in the experiments presented in later chapters. Methods for regression, dimensionality reduction, basis decomposition and state space modeling will be discussed.

3.1 Multiple Linear Regression

Linear regression models the relationship between some scalar dependent variable α and a dependent variable y through a projection given by β . We assume that each value in α is a linear combination of (in this case) features $\{y_1, \dots, y_m\}$,

$$\alpha = \mathbf{Y}\beta \quad (3.1)$$

where \mathbf{Y} is an $N \times M$ matrix, M is the number of features and N is the number of examples. The projection matrix for mapping from \mathbf{Y} to α is determined in the least squares sense through the following minimization

$$\hat{\beta} = \min_{\beta} \|\mathbf{Y}\beta - \alpha\|_2^2. \quad (3.2)$$

3.2 Linear Dynamical Systems

Linear dynamical systems models the statistical properties of real-valued multivariate observations. A latent state variable models the evolution of the sequence, capturing the dynamic nature of the data. Figure 3.1 shows a depiction of an inputless linear dynamical system, the variables in Equations 3.3–3.6 are shown in the diagram.

We formulate the linear dynamical system as follows

$$\alpha_t = \mathbf{A}\alpha_{t-1} + \mathbf{w}_t \quad (3.3)$$

$$\mathbf{y}_t = \mathbf{C}\alpha_t + \mathbf{v}_t + \bar{y}_t. \quad (3.4)$$

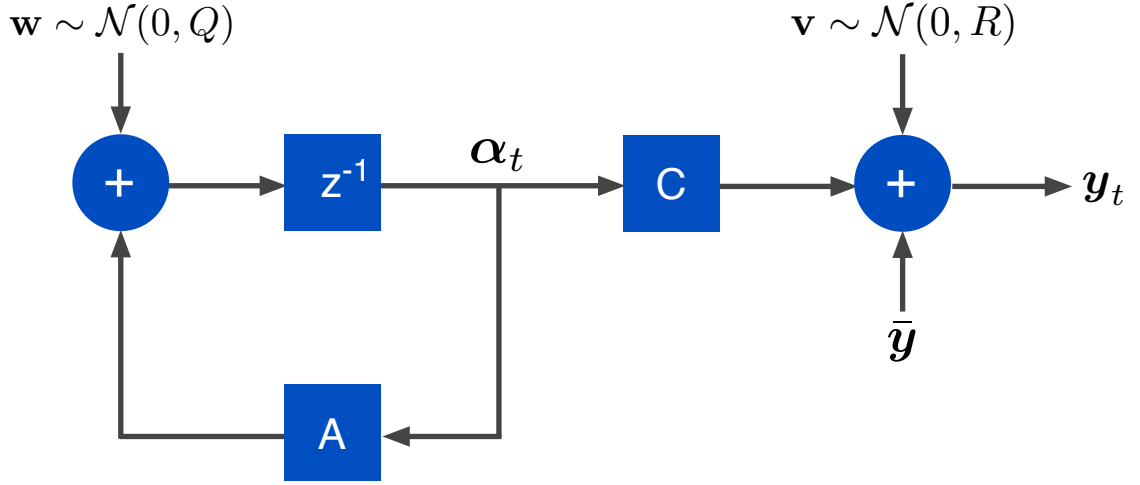


Figure 3.1: Diagram of a linear dynamical system modeling a noisy process.

Here, \mathbf{w}_t and \mathbf{v}_t are sampled from zero mean Gaussian noise sources

$$\mathbf{w} \sim \mathcal{N}(0, Q) \quad (3.5)$$

$$\mathbf{v}_t \sim \mathcal{N}(0, R). \quad (3.6)$$

The dynamics matrix \mathbf{A} models the evolution of the data as a linear transformation in each time step and \mathbf{C} translates the α values from the latent state space to the observation space, $y \in Y^{\mathbf{R}}$.

To train the model \mathbf{A} and \mathbf{C} are estimated through constraint generation and least squares, respectively. A constraint generation approach is used to estimate \mathbf{A} since a stable solution is guaranteed [84]. The covariances \mathbf{Q} and \mathbf{R} are computed from the residuals of \mathbf{A} and \mathbf{C} . Prior to training, the data is mean centered due to the model assumption that the variables are Gaussian and zero mean. The feature $\bar{\mathbf{y}}$ and weight $\bar{\alpha}$ means are retained for the testing phase.

3.3 Dynamic Texture Mixtures

Linear dynamical systems are often referred to as dynamic textures. Dynamic textures were developed in the computer vision community to model sequences that exhibit stationary characteristics in space and time [12, 19]. They were shown to successfully represent the varying statistical properties of audio based on timbre features in [4]. A similar mathematical formulation used in Equations 3.3-3.6. It is instructive to consider the graphical model associated with the dynamic texture and

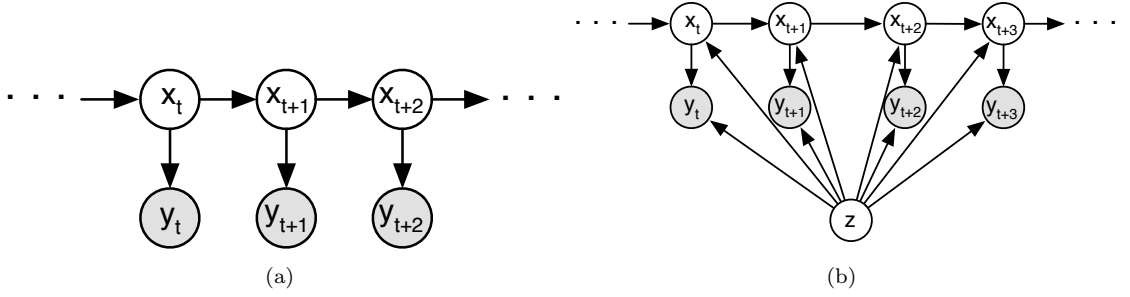


Figure 3.2: Graphical model for (a) dynamic texture (b) dynamic texture mixture model.

the dynamic texture mixture. Figure 3.2(a) shows the graphical depiction of the dynamic texture where \mathbf{y}_t are our observations and \mathbf{x}_t are our hidden states. Note that this bears much resemblance to a Hidden Markov Model (HMM). The LDS and HMM models bear deep resemblance both in terms of their structure and general methods of inference. The graphical model in Figure 3.2(a) is the same for an HMM, the key difference being that an HMM has discrete states for the latent variable whereas the LDS contains a continuous distribution over the latent variables.

Figure 3.2(b) shows the graphical model for a *mixture* of dynamic textures. If (a) is a dynamic texture with parameters $\Theta = \{\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \bar{\mathbf{y}}\}$ then (b) represents the addition of the mixture component priors $\mathbf{z} = \{z_1, z_2, \dots, z_k\}$ such that the individual LDS component parameters are now given by $\Theta_{\mathbf{k}} = \{\mathbf{A}_k, \mathbf{C}_k, \mathbf{Q}_k, \mathbf{R}_k, \bar{\mathbf{y}}_k\}$. The system of equations defining the dynamic texture mixture are

$$\mathbf{x}_t = \mathbf{A}_z \mathbf{x}_{t-1} + \mathbf{w}_t \quad (3.7)$$

$$\mathbf{y}_t = \mathbf{C}_z \mathbf{x}_t + \mathbf{v}_t + \bar{\mathbf{y}}_z \quad (3.8)$$

$$\mathbf{w} \sim \mathcal{N}(0, \mathbf{Q}) \quad (3.9)$$

$$\mathbf{v} \sim \mathcal{N}(0, \mathbf{R}). \quad (3.10)$$

Here we have introduced the mixture component variable

$$z \sim \text{multinomial}(p_1, \dots, p_K), \quad \text{with } \sum_{k=1}^K p_k = 1. \quad (3.11)$$

To learn a DTM, a signal is separated into N segments $\{y^{(i)}\}_{i=1}^N$ with $y^{(i)} = \{y_1^{(i)}, \dots, y_\tau^{(i)}\}$ where τ is the segment length. The parameters Θ that best fit the data are learned in the maximum-likelihood

sense,

$$\boldsymbol{\Theta}^* = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}} \sum_{i=1}^N \log p(y^{(i)}; \boldsymbol{\Theta}). \quad (3.12)$$

The log likelihood of the data is maximized with respect to the parameters. This is accomplished using the Expectation-Maximization (EM) algorithm.

3.4 Principal Component Analysis

A common technique used for dimensionality reduction, data visualization and feature extraction is Principal Component Analysis (PCA). This method is defined as an orthogonal projection of data into a principal subspace such that the variance of each dimension of the projected data is maximized. For a set of vector observations \mathbf{x}_n of dimensionality D we project the data into a space of dimension K , where $K < D$. For simplicity consider the case $K = 1$ and let us define a vector \mathbf{p} as a unit vector such that $\mathbf{p}^T \mathbf{p} = 1$. The projection of the data into this single dimension is then $\mathbf{y} = \mathbf{p}^T \mathbf{x}_n$. The variance of the projected data is then given by

$$\frac{1}{N} \sum_{n=1}^N (\mathbf{p}^T \mathbf{x}_n - \mathbf{p}^T \bar{\mathbf{x}})^2 = \mathbf{p}^T \mathbf{S} \mathbf{p} \quad (3.13)$$

where \mathbf{S} is the covariance matrix of the data and $\bar{\mathbf{x}}$ is the sample mean

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n. \quad (3.14)$$

We want to maximize the projected variance $\mathbf{p}^T \mathbf{S} \mathbf{p}$ with respect to \mathbf{p} . To do this, we introduce a Lagrange multiplier λ ,

$$\mathbf{p}^T \mathbf{S} \mathbf{p} + \lambda(1 - \mathbf{p}^T \mathbf{p}). \quad (3.15)$$

Differentiating the above equation with respect to \mathbf{p} and setting it equal to zero we arrive at the following

$$\mathbf{S} \mathbf{p} = \lambda \mathbf{p}, \quad (3.16)$$

which requires that \mathbf{p} is an eigenvector of \mathbf{S} . Multiplying both sides of Equation 3.16 by \mathbf{p}^T results in the variance in the projected domain

$$\mathbf{p}^T \mathbf{S} \mathbf{p} = \lambda. \quad (3.17)$$

Hence, the projected variance is at a maximum when p is the eigenvector corresponding to the largest eigenvalue. In matrix form, generalized for an M -dimensional projection we have

$$\mathbf{Y} = \mathbf{\Lambda}^{-1/2} \mathbf{P}^T (\mathbf{X} - E\{\mathbf{X}\}) \quad (3.18)$$

3.5 Non-Negative Matrix Factorization

Given a non-negative matrix \mathbf{V} , the problem of finding a matrix decomposition such that

$$\mathbf{V} \approx \mathbf{WH} \quad (3.19)$$

where both \mathbf{V} and \mathbf{H} are non-negative matrix factors of \mathbf{V} is known as Non-Negative Matrix Factorization (NMF). This decomposition technique has been shown to be useful for a variety of problems in signal processing for images and audio data. The basic formulation is as follows. For a $n \times m$ matrix \mathbf{V} with m examples and n features, the data is approximated factorized into an $n \times k$ matrix \mathbf{W} and $k \times m$ matrix \mathbf{H} . The value chosen for k determines the number of components used to reconstruct the data and therefore \mathbf{W} and \mathbf{H} are a reduced dimensional representation of \mathbf{V} . Each vector in \mathbf{V} is approximated as a linear combination of the basis vectors in \mathbf{W} as

$$\mathbf{v}_k \approx \mathbf{W} \mathbf{h}_k \quad (3.20)$$

To find an approximation \mathbf{WH} for \mathbf{V} we define a cost, or distance to minimize between the original data and the component reconstruction. The square of the Euclidian distance between two matrices \mathbf{A} and \mathbf{B} is given by

$$\|\mathbf{A} - \mathbf{B}\|^2 = \sum_{ij} (A_{ij} - B_{ij})^2. \quad (3.21)$$

A measure of *divergence* is defined as

$$D(\mathbf{A} \parallel \mathbf{B}) = \sum_{ij} \left(A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij} \right), \quad (3.22)$$

and reduces to the Kullback Leibler divergence when $\sum_{ij} A_{ij} = \sum_{ij} B_{ij} = 1$ such that \mathbf{A} and \mathbf{B} are specified as probability distributions. Equations 3.21 and 3.22 are convex in \mathbf{W} and \mathbf{H} respectively. Convexity is not guaranteed across both variables simultaneously, therefore we alternate between minimizing Equation 3.21 which is convex in \mathbf{W} and Equation 3.22 which is convex in \mathbf{H} . Minimizing

$\|\mathbf{V} - \mathbf{WH}\|$ with respect to \mathbf{W} and \mathbf{H} and enforcing the constraint $\mathbf{W}, \mathbf{H} \geq 0$, the multiplicative update rules are

$$H_{km} \leftarrow H_{km} \frac{(W^T V)_{km}}{(W^T W H)_{km}}, \quad W_{nk} \leftarrow W_{nk} \frac{(V H^T)_{nk}}{(W H H^T)_{nk}}. \quad (3.23)$$

Minimizing $D(\mathbf{V} \|\mathbf{WH})$ with respect to \mathbf{W} and \mathbf{H} and enforcing the constraint $\mathbf{W}, \mathbf{H} \geq 0$, the multiplicative update rules are

$$H_{km} \leftarrow H_{km} \frac{\sum_n W_{nk} V_{nm} / (\mathbf{WH})_{nm}}{\sum_n W_{nk}}, \quad W_{nk} \leftarrow W_{nk} \frac{\sum_m H_{km} V_{nm} / (\mathbf{WH})_{nm}}{\sum_m H_{km}}. \quad (3.24)$$

The matrices \mathbf{W} and \mathbf{H} are computed by alternating between Equations 3.23 and 3.24 for a specified number of iterations or until the change in the cost function per iteration goes below a given threshold.

3.6 Probabilistic Latent Component Analysis

Probabilistic Latent Components Analysis (PLCA) has seen an increase in use in the audio domain due to its flexibility to learn convolutive bases, impose sparsity constraints and enforce shift invariance in a two dimensional basis [86]. The basic formulation of PLCA is very similar to that of NMF in that it is non-negative in both the components and activations. In fact in certain limiting cases, it has been shown to be numerically equivalent to NMF [87]. It is instructive to think of it as a probabilistic interpretation of NMF with a latent prior z which allows for imposing constraints on the learned representation through prior probabilities.

PLCA models a distribution over N dimensional data $\mathbf{x} = \{x_1, x_2, \dots, x_N\}$ as a sum of latent distributions

$$P(\mathbf{x}) = \sum_z P(z) \prod_{j=1}^N P(x_j|z). \quad (3.25)$$

$P(x_j|z)$ is a latent marginal distribution across the dimension of variable x_j , conditioned on the latent variable z , and $P(z)$ is the prior probability of the latent component. Therefore, $P(\mathbf{x})$ is a distribution composed of a weighted sum of marginal distribution products. Both $P(x_j|z)$ and $P(z)$ are estimated from the observation density $P(\mathbf{x})$. The marginal distributions $P(x_j|z)$ are estimated using an EM variant where the contribution of the latent variable z is computed in the expectation

step as

$$R(\mathbf{x}, z) = \frac{P(z) \prod_{j=1}^N P(x_j|z)}{\sum_{z'} P(z') \prod_{j=1}^N P(x_j|z')} \quad (3.26)$$

Here we note that $R(\mathbf{x}, z)$ is the contribution of the latent variable since the normalization is over all latent states except the current state being estimated, hence this is not a probability and does not sum to one. In the maximization step, we use the latent contributions to estimate the new marginal densities

$$P(z) = \sum_j \sum_{x_j} P(\mathbf{x}) R(\mathbf{x}, z) \quad (3.27)$$

$$P(x_j|z) = \frac{\sum_{i:i \neq j} \sum_{x_i} P(\mathbf{x}) R(\mathbf{x}, z)}{P(z)} \quad (3.28)$$

3.6.1 Convolutional Formulation

The model in Equation 3.25 can be extended to produce shift-invariance by defining the decomposition as a set of kernel distributions and impulse distributions. The kernel distributions are small two dimensional patches that are convolved with a sparse impulse distribution and weighted by the latent probabilities to produce the original distribution $P(\mathbf{x})$. Our model now becomes

$$P(x, y) = \sum_z P(z) \sum_{\tau_x, \tau_y} P(\tau_x, \tau_y|z) P(x - \tau_x, y - \tau_y|z). \quad (3.29)$$

Here $P(\tau_x, \tau_y|z)$ is a two dimensional kernel distribution and is restricted such that $P(\tau_x, \tau_y|z) = 0 \forall (\tau_x, \tau_y) \notin \mathfrak{R}_{\tau_x, \tau_y}$ where $\mathfrak{R}_{\tau_x, \tau_y}$ is a region chosen such that it is smaller than the impulse distribution. When the kernel distribution is convolved with the two dimensional impulse distribution $P(x - \tau_x, y - \tau_y|z)$ and weighted by the latent variables, it generates an estimate of the data distribution $P(x, y)$. The expectation step for this convolutional two dimensional case becomes

$$R(x, y, \tau_x, \tau_y, z) = \frac{P(z) P(\tau_x, \tau_y|z) P(x - \tau_x, y - \tau_y|z)}{\sum_z P(z') \sum_{\tau'_x, \tau'_y} P(\tau'_x, \tau'_y|z') P(x - \tau'_x, y - \tau'_y|z')}, \quad (3.30)$$

and the subsequent maximization steps are

$$P(z) = \sum_{x, y, \tau_x, \tau_y} P(x, y) R(x, y, \tau_x, \tau_y, z) \quad (3.31)$$

$$P(\tau_x, \tau_y|z) = \frac{\sum_{x, y} P(x, y) R(x, y, \tau_x, \tau_y, z)}{P(z)} \quad (3.32)$$

$$P(x, y|z) = \frac{\sum_{\tau_x, \tau_y} P(x + \tau_x, y + \tau_y) R(x + \tau_x, y + \tau_y, \tau_x, \tau_y, z)}{\sum_{x', y', \tau_x, \tau_y} P(x' + \tau_x, y' + \tau_y) R(x' + \tau_x, y' + \tau_y, \tau_x, \tau_y, z)}. \quad (3.33)$$

A graphical depiction of the convolutive PLCA model is shown in Figure 3.3. The kernel distribu-

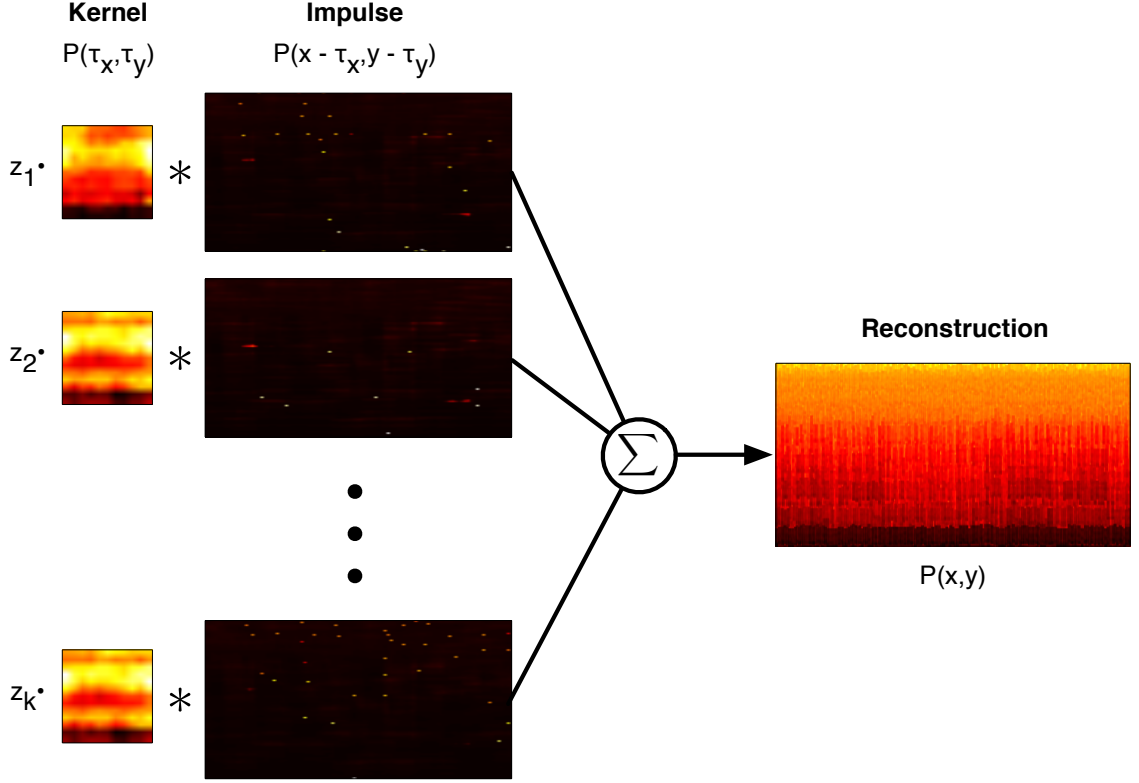


Figure 3.3: A depiction of basis decomposition using convolutive PLCA. Two-dimensional kernel distributions are learned with corresponding activations (impulse distributions). These two components are convolved and multiplied by the latent weights (z) to produce the reconstruction of the original distribution.

tions are convolved with the impulse distributions and weighted by the latent component probabilities z . These marginal distributions are then summed to produce an estimate of the true distribution $P(x, y)$. There is no implicit formulation in the model that determines which distribution is the impulse and which is the kernel. To alleviate this concern, a sparsity constraint is introduced. An entropic a-priori distribution is imposed on the component distributions to minimize their entropy [86]. Let us define θ as the distribution we would like to enforce the entropic prior on. We then specify the *a priori* distribution of θ as $P(\theta) = e^{-\beta \mathcal{H}(\theta)}$ where $\mathcal{H}(\theta)$ is the entropy of θ . The sparsity

constraint adds two additional steps to the parameter estimation process,

$$\frac{\omega}{\theta_i} + \beta + \beta \log(\theta_i) + \lambda = 0 \quad (3.34)$$

$$\theta = \frac{-\omega/\beta}{\mathcal{W}(-\omega e^{(1+\lambda/\beta)}/\beta)}, \quad (3.35)$$

with $\mathcal{W}(\cdot)$ being Lambert's function. If for instance $\theta = P(x_j|z)$ then ω is

$$\omega = \sum_{x_k} P(\mathbf{x}) R(\mathbf{x}, z) \quad \forall k \neq j \quad (3.36)$$

It is important to note that θ could be any distribution in the model (kernel/impulse/prior).

3.7 Datasets

In the Music Information Retrieval (Music-IR) community, large datasets for training and evaluating models are notoriously hard to obtain and share due to the commercial nature of the content. This difficulty is compounded in multi-track sources for several reasons. Music production using DAWs was not commonplace until the recent past and a significant amount of multi-track source audio older than 15 years is archived on analog tape or digital audio tape (DAT). Second, record labels had little incentive to release source audio since the home studio was still rather expensive to own. In the past decade, as technology advanced and home music production became common, bands have released multi-track sources for fans to remix and create derivative work. This section describes two datasets used in the subsequent experiments. The first is a set of stems from the RockBand[®] video game and the second is a collection of multi-track audio from a variety of sources that are publicly available.

3.7.1 Rockband Dataset

There are 48 artists in the RockBand[®] dataset and one song was selected randomly from each of the artists resulting in a total of 48 songs. Only one song was chosen from each artist due to time constraints encountered in generating the data and to prevent over-representation in the dataset. The 'final mix' experienced during gameplay was acquired by recording the optical audio output of the game console onto a computer and aligning it to the source tracks. The game console mix was used, as opposed to the radio/album release, due to synchronization issues between the source

files and the commercial version. It was evident that time stretching/compression was performed on many of the RockBand[®] releases since the song from the commercial release was often not the same length as the version from the game console. Most likely this was done to align the beats so that they occur on regular exact intervals to facilitate gameplay.

Preprocessing and Normalization

There were several inconsistencies in the dataset which we had to account for in order to make comparisons between songs more accurate and to facilitate modeling in the system described in Section 3.2. The number and type of sources varied between each song, with a minimum track count of eight and maximum of 14. For example, many songs had individual stereo (L and R) waveforms for each instrument, whereas other songs only had mono tracks for some instruments and stereo tracks for others. Additionally, not all songs had individual tracks for the kick drum, snare drum or overhead drum microphones.

To deal with this discrepancy, we opted to form five mono tracks for each song: bass, drums, guitar, vocals and backup. The instruments in the backup track vary from song to song and may contain vocal harmonies, synthesizers, percussion, guitar or a variety of other instruments, however the content of the backup track within a song is fairly consistent. Given the variance in the dataset, this method created more uniformity between the content of each song.

To create a single mono track for each instrument class, we mixed all audio that belonged to the given instrument class according to the track weights computed using the method described in Section 6.1. A diagram of the preprocessing step is shown in Figure 3.4.

3.7.2 Multiple Genre Dataset

The second dataset consists of 135 songs across a variety of genres. The genres include Acoustic, Alternative, Country, Dance, Electronic, Hip-Hop, Indie, Jazz, Rock and Metal. The songs were obtained from three primary sources: Weathervane Music¹, Sound on Sound² and a multi-track dataset used for song structure segmentation [31]. Each track is converted to a monaural source at 44.1kHz sampling rate and labeled with the instrument present in the track.

The tracks in every song are labeled by three individuals and the majority label for each track was retained as ground truth. The labelers are students in the music industry program at Drexel

¹<http://weathervanemusic.org/>

²<http://www.soundonsound.com/>

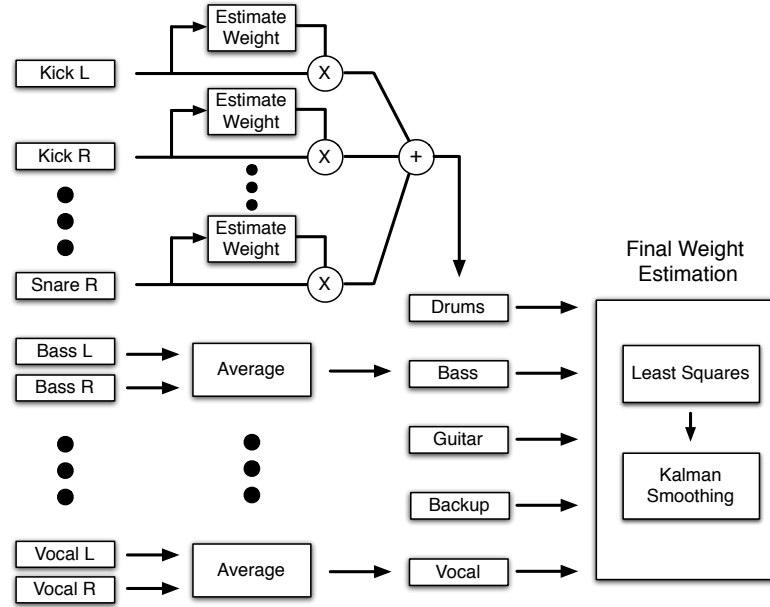


Figure 3.4: Diagram of dataset preprocessing for each song in the RockBand dataset.

University and the author. The filenames for each audio track are used when possible and normalized to a standard label for a single instrument class. Instrument classes are differentiated on a fine level (clean/distorted electric guitar) and may be combined into superclasses (electric guitar) if desired. The electric guitar is a specific example where fine level labels are desired since the distorted and clean versions are treated very differently by engineers and have much different roles in the mix. The dataset is publicly available online³.

There is much more variation in this dataset than in the one compiled in Section 3.7.1. All of the material in the RockBand dataset possesses similar instrumentation and was commercially released. In addition to spanning multiple genres, the open dataset is not all commercially available material and varies in terms of the quality of the signal capture (i.e. experience of the recording engineer) as many of the songs come from novice home studio users.

³<http://music.ece.drexel.edu/research/AutoMix>

4. Instrument Based Processing

The experiments in this chapter are designed to explore the efficacy of an approach to mixing audio that uses information about the instrument present in a track to make processing decisions. In this approach we attempt to codify some common practices and apply them to multi-track drum audio. Several professional and student mixing engineers were interviewed about the process of mixing audio and it was unsurprising to find that all of them specified that their approach is dependent upon the source material (i.e. genre, instrumentation). It is quite difficult to define a set of hard and fast rules for mixing audio yet there do exist some commonalities that many agree upon. We apply some basic techniques to improve the balance and quality of the drums via stereo panning, filtering and level adjustment. The motivation for the processing techniques employed in the following subsections are derived from the engineer interviews as well as authoritative sources on mixing [83, 37].

There are several concerns when combining the signals from multiple drum microphones to produce a mixture. Problems with phase coherence between the different microphones can often occur and result in a comb filtering effect applied to the instruments [83]. This is the case with bleed (leakage) between microphones on different instruments as well as multiple microphones on a single instrument (as in the top/bottom heads of a snare drum). In properly recorded material this effect is usually anticipated for and dealt with during signal capture and therefore not considered in this paper.

We consider three processing areas: level balancing, stereo panning and equalization. Two basic approaches for level adjustment are serial (faders down) and parallel (faders up) [37, 83]. The serial approach involves adding in layers one at a time and the parallel approach starts with all layers active and adjusts levels accordingly. We opt for the parallel approach where the level of each instrument track is evaluated individually against the rest of the mix. There are also two main approaches to using the ambient (overhead/room) mics. One primarily uses the overheads as the main drum signal and uses the individual instrument mics as reinforcement when needed. The alternate approach is to use the close microphones as the primary signal source and use the overhead microphones to increase the amount of cymbals and add ‘air’ to the mix. We opt to use the latter approach in this work.

For panning, one may start with a stereo spread of the overhead mics and pan the close mi-

crophones according to their position in that signal. Another common approach is to pan the kick and snare dead center since they are the driving force of the rhythm section. This is the option we choose in our model.

The equalization applied is minimal and was obtained from the interviews of engineers. The interviewees expressed reservation about making generalizations without hearing the source material and knowing what other instruments are in the mixture, yet these are the same issues they expressed with nearly all aspects of mixing, namely that each session is different and must be approached individually. Nevertheless, a filtering scheme was developed to boost frequency ranges that often need boosting and cut frequency ranges that often need attenuating. Ideally, this would be done adaptively through comparing bandwise energy ratios and making adjustments accordingly.

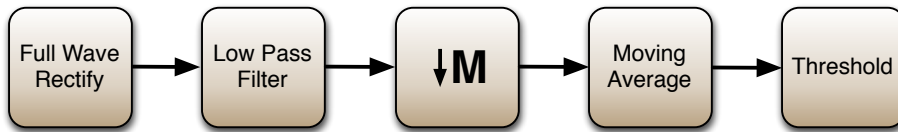


Figure 4.1: Processing chain to calculate the active areas of an instrument track.

Before processing, each track is analyzed to determine where the instrument is playing on each track. We only want to compare signal characteristics where there is an active instrument in a track, not where there is just the noise floor. Figure 4.1 depicts the computation of the active regions in each track. The first four steps, full-wave rectification, low pass filtering, downsampling and smoothing with a moving average filter produce the temporal envelope of the signal and the threshold determines active regions. After thresholding, any segments less than 150ms long are discarded.

4.1 Stereo Panning

In [49, 63] a dynamic cross-adaptive model is used to actively pan tracks as they come in and out of the instrument mixture based on several constraints related to spectral and spatial balance and masking. Here we attempt to leverage common practices in drum kit panning and apply them to the individual tracks of a drum kit. This results in a static value being applied to the entire track for the duration of the song regardless of the presence or absence of instrument playing at any

Instrument Class	Panning Value (θ)	Gain Values $\{\alpha, \beta, \lambda\}$
Kick Drum	0 (center)	$\{0.9, 1.2, 2\}$
Snare Drum	0 (center)	$\{0.9, 1.2, 2\}$
Toms	Spaced $\{-25, 25\}$	$\{0.8, 1.3, 4\}$
Overhead/Room	$\{-35, 35\}$	$\{0.8, 1.3, 4\}$

Table 4.1: Mixing parameter values for individual drum tracks.

given time. Panning a drum kit is one aspect of mixing that is fairly consistent between engineers. Qualitatively, the stereo balance of the drum mix is as follows:

1. Kick drum panned center
2. Snare drum panned center
3. Toms panned from left to right
4. Overhead microphones panned left and right

Panning is accomplished by applying the sine-cosine panning law

$$L_{pan} = \cos(45^\circ - \theta) \quad (4.1)$$

$$R_{pan} = \sin(45^\circ - \theta). \quad (4.2)$$

Here $\theta \in [-45^\circ, 45^\circ]$ and represents the angle offset from the center of the stereo field with -45° being panned fully to the left and $+45^\circ$ panned fully right. This method of panning maintains the perceived loudness of the signal as it is varied from left to right. Table 4.1 shows the parameter values used to pan the tracks.

The kick and snare drums are panned in the center of the stereo field. The toms are spaced linearly from left to right with 25° being the maximum offset from the center position. The overhead tracks are panned alternating left and right at the specified value in Table 4.1.

4.2 Relative Levels

After panning, the loudness of each track is computed and compared against the loudness of the rest of the tracks to determine any boost or attenuation that is desired for each track. The loudness

of each track is calculated by filtering the signal using the inverse of the ISO 226 normal equal-loudness-level contours (at 75 phons) and then computing the RMS energy over a 23ms window [35]. The level of 75 phons was chosen based on preferred listening levels shown in [34]. The loudness of the target track (\mathbf{x}_{loud}) is compared to the loudness of the sum of the remaining tracks (\mathbf{y}_{loud}) and a loudness ratio is computed,

$$r_{loud} = \frac{1}{T} \sum_{\tau} \frac{x_{loud}^{(\tau)}}{y_{loud}^{(\tau)}}, \quad (4.3)$$

where x and y are in dB and T is the total number of short time frames in the current song being analyzed. The loudness ratio is then used to attenuate or boost the level of the track in question. The gain of the track is determined using the following equation

$$g = 10^{(-\frac{1}{\lambda} \log(r_{loud}))}. \quad (4.4)$$

Equation 4.4 offers control over the amount of level correction that is applied to each instrument through the parameter λ . As λ increases, the amount of level correction is reduced as shown in Figure 4.2.

Loudness is computed on each channel (L/R) after panning and the average of the loudness ratios is used to determine the gain of the instrument. There are three parameters $\{\alpha, \beta, \lambda\}$ for each instrument type that determine how the loudness ratio affects the gain, g , applied to the track. The α and β parameters define thresholds for the loudness ratio necessary to apply loudness correction. For example, if we require $r_{loud} < \alpha$ or $r_{loud} > \beta$ where $\alpha = 0.8$ and $\beta = 1.2$ before applying gain g , then the track will have no level correction if $r_{loud} \in [0.8, 1.2]$ and will have loudness correction specified in Equation 4.4 otherwise. The parameters in Table 4.1 are specified to err on the side of more kick and snare drum than overhead and tom microphones since the kick and snare instruments are generally more prominent in rock music.

4.3 Equalization

The desired frequency content for a specific instrument is very genre dependent. For example in an electronic track the kick drum generally contains more low frequency content and may be prominent even into the sub-bass range. In heavy metal, the sound of the beater striking the kick drum is often desirable and the signal may need to be boosted in the high-mid frequency range.

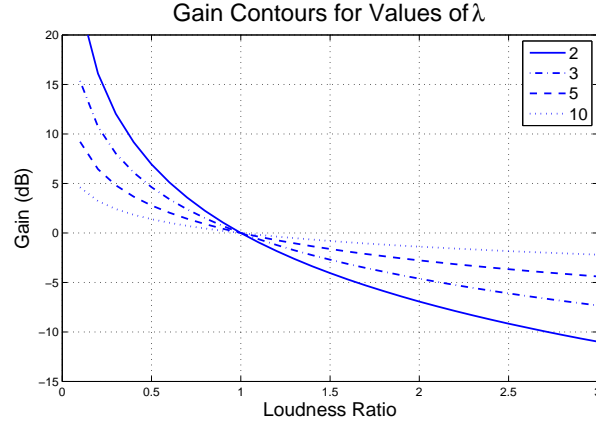


Figure 4.2: Contours of gain attenuation for various γ .

For these reasons we chose to apply only subtle equalization based on some common operations. The kick drum has a 2dB boost from 1kHz-6kHz, a 2dB cut from 400Hz-900Hz and a 2dB boost of 100Hz with a quality factor of 4.5. The snare drum has a 3dB high shelving boost starting at 10kHz. These modifications are designed to give the kick drum slightly more punch and the snare drum more brilliance.

4.4 Drum Type Classification

For an unknown set of tracks, the drums would need to be identified to apply the common practices outlined above. Here we explore a preliminary experiment to classify a track in terms of the drum content it contains. The approach is fairly standard for supervised learning and is meant to serve as a benchmark of the difficulty of this particular dataset.

Features	Features (cont.)
MFCC	RMS
Centroid	Bandwidth
Flux	Zero-Crossing Rate
Number of Segments	Inter Onset Interval
Segment Length	

Table 4.2: Features used in drum type classification.

A support vector machine (SVM) classifier with radial basis function (RBF) kernel is trained and

evaluated via 5-fold cross validation using LIBSVM [13]. This is a four class problem ($C \in \{1, 2, 3, 4\}$) with the four classes being kick drum, snare drum, tom-tom and overhead. The features used in the experiment are listed in Table 4.2 and include mel-frequency cepstral coefficients (MFCC) (20 dimensions), spectral features and time domain features as well as information about the amount of time active audio is present in the track. The first and second derivatives of each feature (non-singleton) is also included in the dataset. This results in 138 total feature dimensions which is then reduced through principle components analysis (PCA). The classifier achieved an average accuracy across all folds of 0.504.

For a four class problem, this result is not particularly promising, but the model and features used are not as advanced as those in [80, 92, 25, 22]. Although the data is in multi-track format, there are still several instruments present via the bleed of the microphones. For the tom-tom drums, the majority of the track resembles an overhead microphone signal of low amplitude until the drum is (with relative infrequency) struck. This type of real-world situation increases the difficulty of performing classification.

4.5 Listening Evaluation

To evaluate the ability of the model to appropriately mix the drum tracks together, a listening test is performed where participants noted their preference for the individual monaural tracks summed versus the mix generated with the model. The ground truth instrument labels are used for generating the mixes using the model. Ten songs were selected at random from the dataset and a 15 second clip for each song was selected so that as many of the individual drum tracks were active as possible. Most songs in the dataset do not have drum stems associated with them, only the raw unmixed multi-track session and the final professional mix. The majority of songs that do have mixed drum stems are from the same studio and use very similar processing chains. Therefore to avoid over-representing that subset we only included the summed mix and the automatic mix across a larger number of sources.

The clip pairs were presented with the summed version and the automatically mixed version appearing in random order. Each participant was presented clip pairs one at a time and asked which clip sounds more balanced. They could choose Clip A, Clip B or No Preference. The participants were asked to provide their level of experience with audio mixing and production, the distribution is shown in Table 4.3. Subjects are graduate and undergraduate students at Drexel in the music

Production Familiarity	Participants
None	4
Novice	4
Intermediate	6
Expert	1

Table 4.3: Listening test participant familiarity with audio mixing and production.

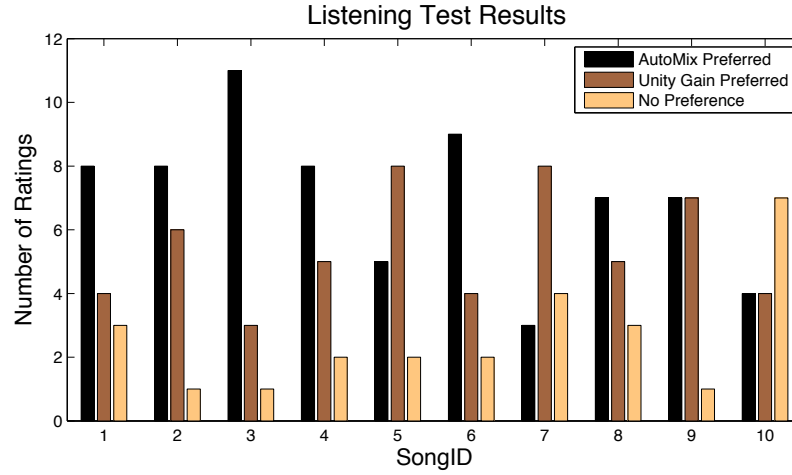


Figure 4.3: Listening test results showing the number of ratings for each clip pair.

industry and engineering programs. Most subjects are male, with only two participants being female. There were 15 total participants in the study with about half having little experience working with audio production and the other half having significant experience.

Figure 4.3 shows the results of the listening test. For six of the ten songs, the model is preferred over the summed mix and listeners prefer two of the ten monaural summed mixes. Songs 7, 8 and 10 contain some drum loops from a library and do not adhere to the ‘standard’ recording technique of having kick, snare, tom and overhead microphones. The dataset represents a variety of material from various sources and varying quality. Some material is recorded professionally and sounds reasonably balanced through just summing the tracks.

The method obtains fair performance on a certain class of song in the dataset but is not able to gracefully handle inconsistencies in recording quality present in the dataset. One caveat of working with multi-track audio is the lack of standardization for recording sessions. This makes obtaining well labeled consistent datasets to train models a difficult task in itself. The work here demonstrates the

possible potential of a hierarchical system that combines both best practices and common techniques of mixing engineers with more sophisticated models of instrument identification, however there is significant room for improvement.

For the classification task, there exist more advanced methods in the literature, yet most apply to individual instrument samples and not full recorded tracks. Including more information about the temporal evolution of a signal as well as taking advantage of the audio in multiple drum tracks while classifying each track could improve results significantly.

Genre information plays a significant role in the desired drum sound for a given song. A jazz kit requires much different treatment than a dance or house drum beat, however genre recognition is not a solved problem and the definitions of genres are constantly evolving. This is an aspect of automatic mixing where it would make sense to expose a parameter to the user and offer ‘presets’ similar to most audio plugins.

More adaptive methods can be used on the track level processing that computes the active segments and loudness as in [95]. Perhaps the most important aspect is further user evaluation and iteration based on listening test results. The ultimate goal of automated mixing systems is to make the mix sound better to the user. Mixing audio often demands an iterative coarse-to-fine approach where the engineer is constantly making changes and then evaluating those decisions in the context of the mix [37, 83].

This is an introductory work that explores the potential of a hierarchical approach to multi-track mixing using instrument class as a guide to processing techniques. While the classification and listening evaluation results have room for improvement, a system basing mixing decisions on the instruments in the mixture warrants further investigation.

5. Representing Dynamic Timbre

In the experiments outlined in Chapter 4, the instrument classification task was based on the assumption that the engineer can use instrument type to define processing decisions. This may serve as a reasonable assumption in many cases but it would be more intuitive to view different tracks in terms of their spectro-temporal profile, that is the time and frequency evolution characteristics of sounds rather than restricting analysis to the physical (or synthesized) sound source.

As an example consider the following situation: a snare sound is captured and recorded in the context of a drum kit. The musician or producer desires a different sound and instead uses a white noise source modified to have a temporal envelope that represents a sharp attack. They then apply an equalizer to remove extreme low frequencies and provide significant rolloff in the higher frequency ranges. This makes the noise burst sound less wideband and more like a snare drum. The spectral and temporal envelopes of a real snare and a modified noise source are shown in Figure 5.1. The temporal envelopes for the real and synthetic snare are very similar. The actual snare envelope appears more natural as it is a result of a damped physical system settling to a resting state. In the case of the synthesized snare-like sound, the envelope exhibits a more linear taper at the tail of the sound. The spectral envelope in (b) shows a resonance at approximately 200 Hz. The synthesized example in (d) also shares this same basic characteristic but has much more mid-range frequency content. Using a more surgical approach to equalization, a closer approximation to the envelope in (b) could be attained, however in practice it is often desirable to have a synthetic sound in order for it to sound different yet still maintain the basic properties that make it fill the role of the snare drum. In the context of a supervised classification experiment, these two sources would exhibit very similar acoustic features yet have different labels. Two options would be to either develop a labeling system that favors this type of similarity or to learn unsupervised groupings based on these trends.

This chapter explores representations of the spectro-temporal characteristics of instrument sounds. Similar to the work presented in Section 2.3.3 is an attempt to capture the spectral shape and timbre of sound and how it evolves over time. To achieve this goal, dynamic textures and dynamic texture mixtures model latent structure evolution in the data to capture the most prominent characteristics that define the audio source spectrally and temporally.

Section 3.2 described how a linear dynamical system (LDS) could be used in the framework of a supervised machine learning task. The time varying mixing coefficients of the multi-channel mixture

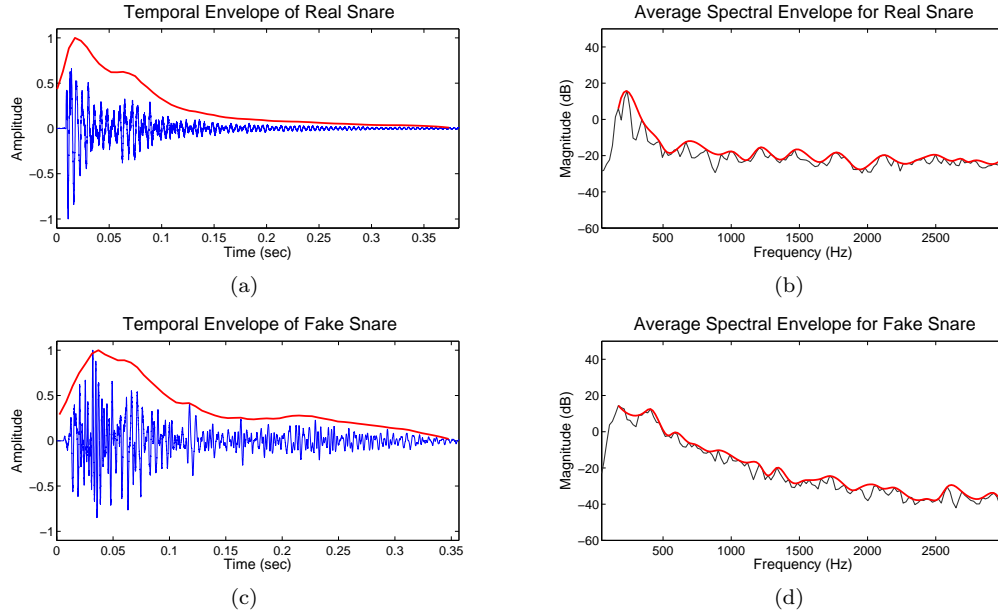


Figure 5.1: Spectral and temporal envelopes for snare drum (a-b) and a white noise burst with modified temporal and spectral envelope (c-d).

were modeled as the hidden state vector of an LDS. The observation space was a set of acoustic features from each track present in the system. To predict the fader values for an unknown mixture, audio features were computed and Kalman filtering was performed to obtain the hidden state vector at time t . To illustrate the efficacy of a LDS to capture timbral dynamics in audio the model is used to encode the time-frequency evolution of tones in a system and then reconstruct the audio from the LDS.

5.1 Modeling Instruments as Dynamic Textures

In order to obtain a representation of the evolving spectro-temporal characteristics of the audio spectrograms of instrument tones are modeled as a dynamic texture (DT) [4]. Dynamic textures were developed in the computer vision community to model sequences that exhibit stationary characteristics in space and time [12, 19]. The characterization of musical instrument sounds (e.g. depressed piano keys and plucked-guitars) as dynamic textures is based on the assumption that tones can be viewed as short-time stationary signals. The following experiments were presented in full in [78], what follows is a summary of that work. To capture the temporal evolution between successive Short-Time Fourier Transform (STFT) frames, each frame is considered the output of a

linear dynamical system at time step t .

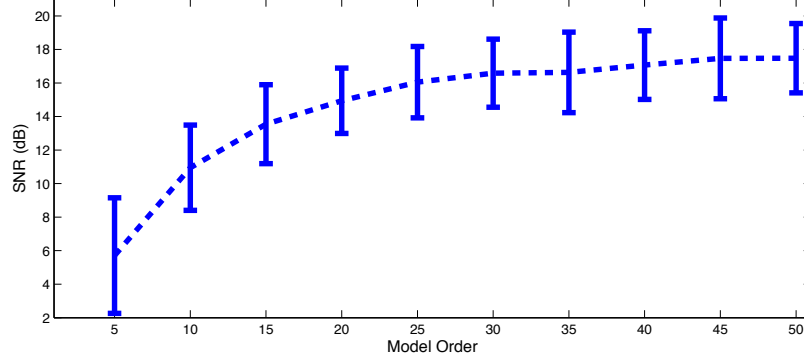


Figure 5.2: Average SNR and standard deviation computed for 21 piano tones against the model dimension n .

5.1.1 Parameter Estimation

The LDS parameters are estimated by computing the STFT of the signal using a 23 msec Hann window with 50% overlap. This decomposition yields the spectrogram $Y = [\mathbf{y}_1 \mathbf{y}_2 \dots \mathbf{y}_\tau]$ where each \mathbf{y} represents the stacked real and imaginary N Discrete Fourier Transform (DFT) coefficients for the underlying segment of the signal. The STFT is factored using singular value decomposition (SVD) [54] such that $Y \approx U \Sigma V^H$. C and X are estimated as,

$$C = U \quad X = \Sigma V^H, \quad (5.1)$$

where $U \in \mathbf{R}^{N \times N}$, $\Sigma \in \mathbf{R}^{N \times \tau}$ and $V^H \in \mathbf{R}^{\tau \times \tau}$. Note that $X = [\mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_\tau] \in \mathbf{R}^{N \times \tau}$ is the matrix of hidden state variables and the LDS is driven by the initial state vector \mathbf{x}_1 .

The system's dynamics matrix A is determined by predicting the transitions between the hidden state variables in X such that,

$$AX_{0:\tau-1} = X_{1:\tau}. \quad (5.2)$$

A can be determined using least squares estimation, though this approach does not guarantee stability, which is problematic when modeling and synthesizing audio signals. Instead, a constraint

generation approach is employed for estimating A , proposed by Siddiqi *et al* [84]. This technique determines A from a set of stable matrices that best satisfies (5.2). Once C and A are estimated, the covariances Q and R are estimated from the model residuals using the minimum variance unbiased estimator of a Gaussian covariance.

While the LDS models the temporal evolution of the signal, the actual estimation of the parameters is only performed on a single time step. An alternative to the above approach is to structure Y as a block Hankel matrix where each column incorporates future observations of the STFT [54]. This has the effect of estimating state variables that account for the present and future outputs.

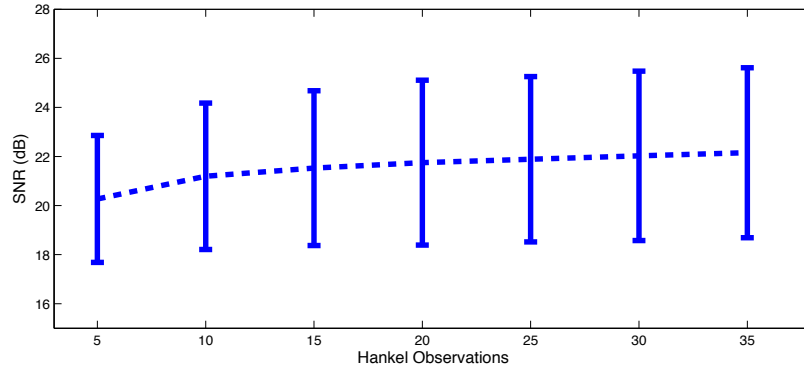


Figure 5.3: Average SNR and standard deviation computed for for 21 piano tones by varying the number of Hankel observations with $n = 40$

5.1.2 Model Reduction and Synthesis

After estimating the LDS for a particular tone, it is desirable to reduce the model dimensionality while still being able to accurately represent the signal's STFT. The symmetric coefficients for each frame of the signal's FFT are eliminated as redundant information. Secondly, additional reduction is achieved by choosing a model order $n \ll N$. The reduced-order model parameters are obtained by truncation, taking C and X in (5.1) such that $U \in \mathbf{R}^{N \times n}$, $\Sigma \in \mathbf{R}^{n \times n}$ and $V^H \in \mathbf{R}^{n \times \tau}$.

The tones are re-synthesized by taking the output of the LDS as each frame in the STFT. The redundant magnitude and phase information is used to reconstruct the full FFT frame and the inverse FFT is applied to yield the windowed audio signal. This procedure is repeated for each output frame and the signal is reconstructed using overlap-add (OLA) corresponding to the analysis rate used to derive the model [67].

Figure 5.2 illustrates the average Signal-to-Noise-Ratio (SNR) computed between 21 analyzed piano tones and their reconstructions using LDS models with increasing model order. As expected, the SNR improves as additional dimensions are used to model the tones. The benefit of incorporating future frames of the STFT in parameter estimation in terms of SNR are also investigated. Figure 5.3 demonstrates that including the Hankel observations provides additional SNR improvement in the reconstructed tones.

Through informal listening tests, using $n \geq 20$ yields reconstructed tones that closely approximate the original signal perceptually. However, tones exhibiting high frequency components in the initial transient were not properly suppressed. This artifact can be corrected by increasing the model order, or by including future observations in the LDS estimation. All of the audio examples discussed are available online.¹

5.1.3 Modeling Timbre Variation

While the approach presented in Section 5.1 is capable of modeling the acoustic characteristics of a particular instrument sample, a model that is generalizable in terms of accounting for instrument- and timbre-specific characteristics is desirable.

5.1.4 Joint Analysis

This analysis is restricted to piano tones produced by varying the key-stroke velocity to produce “hard”, “medium” and “soft” tones. Pianists use the key-stroke velocity to convey desired musical expressions by producing different timbres. These timbre differences are observable in the STFT matrix since each tone will have unique time-frequency characteristics corresponding to the velocity used to depress the key and in turn, excite the string. Thus, the aim is to learn the LDS parameters for a piano note played with different velocities where C accounts for the associated timbre of each articulation and A is jointly learned to describe the temporal structure of *all* the tones.

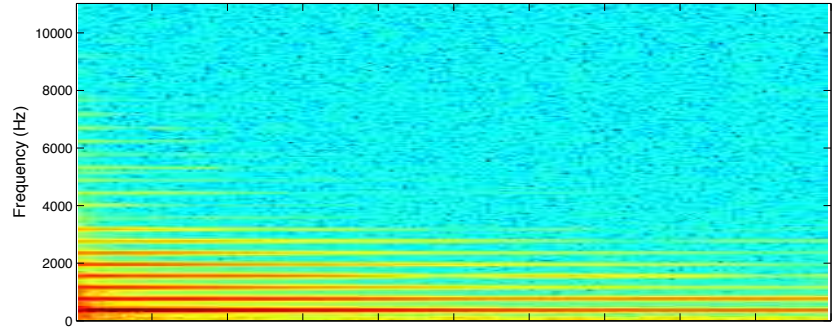
For a particular note played with the described key-stroke velocities, the STFT matrix for each tone is concatenated into a joint observation matrix $Y_J = \begin{bmatrix} Y_s & Y_m & Y_h \end{bmatrix}$ where s , m , and h indicate the soft, medium and hard velocities, respectively. As described in Section 5.1.1 the SVD of Y_J is computed to yield the hidden state variables $X_J = \begin{bmatrix} X_s & X_m & X_h \end{bmatrix}$. Note that the hidden state variable matrix has transition regions between each note velocity s , m and h that are undesirable.

¹<http://music.ece.drexel.edu/research/InstrumentLDS>

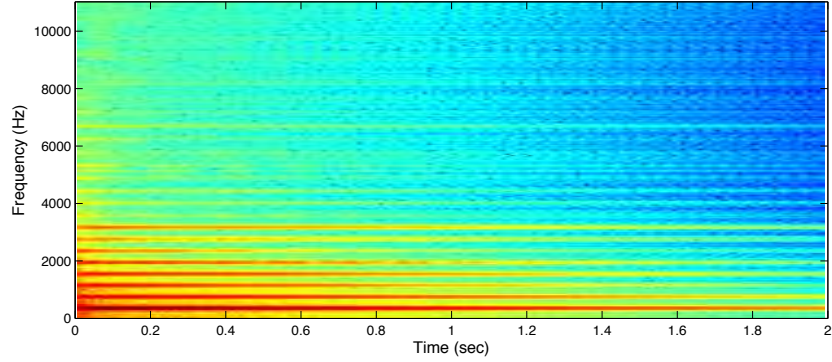
These regions are ignored by solving for a joint dynamics matrix A_J which satisfies

$$A_J \begin{bmatrix} \mathbf{x}_{s,0} \dots \mathbf{x}_{s,\tau-1} & \mathbf{x}_{m,0} \dots \mathbf{x}_{m,\tau-1} & \mathbf{x}_{h,0} \dots \mathbf{x}_{h,\tau-1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{s,1} \dots \mathbf{x}_{s,\tau} & \mathbf{x}_{m,1} \dots \mathbf{x}_{m,\tau} & \mathbf{x}_{h,1} \dots \mathbf{x}_{h,\tau} \end{bmatrix}.$$

As in the individually learned models, constraint generation is used to obtain stable system dynamics.



(a)



(b)

Figure 5.4: Top: Log-magnitude spectrogram for a piano tone produced with a “hard” articulation. Bottom: Re-synthesized piano tone generated from the output of the estimated LDS model.

For each velocity, its observation matrix is determined by solving

$$C_a = Y_a X_a^{-1}, \quad (5.3)$$

where a indicates the velocity of the note. The STFT matrix for each velocity is reconstructed using the initial hidden state vector, x_o from the corresponding hidden state variable matrix X_a and the tone is synthesized using OLA on the frames. For joint modeling, the state and observation noise sources are not included since that they add unwanted noise from the residual computation. An example of an original and reconstructed tone are shown in Figure 5.4.

The joint modeling approach presented in this section also has the benefit of reducing the number of parameters required to synthesize a variety of tones. Using the approach presented in Section 5.1, each tone is modeled with individual A, C and x_0 parameters. Joint modeling can represent several tones with a single dynamics matrix, while describing the tone's timbral characteristics through a unique observation matrix.

5.1.5 Altering Timbre

In the previous section, it was demonstrated that various velocities for a particular tone of a musical instrument could be characterized by a common dynamics matrix and separate observation matrices that encode the spectro-temporal characteristics of a tone. Here, parameterized synthesis is explored by modifying a single observation matrix to create tones of varying velocity. By weighting the observation matrix of a given note, a higher velocity note into transformed into a softer velocity.

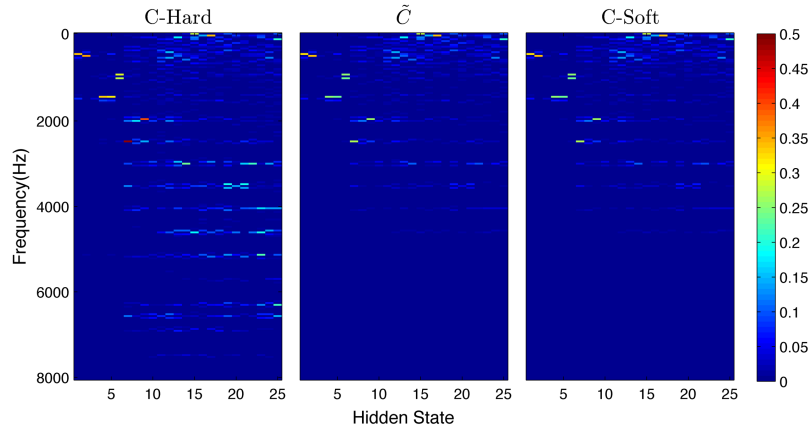


Figure 5.5: C and \tilde{C} for hard velocity and re-weighted to be a lower velocity.

Define the re-weighting as $\tilde{C} = WC$ where W is a diagonal matrix of weighting coefficients.

Scaling each row vector c_n by a constant w_n ensures that the resulting \tilde{C} remains a set of orthogonal basis vectors to project the hidden states into the observation space.

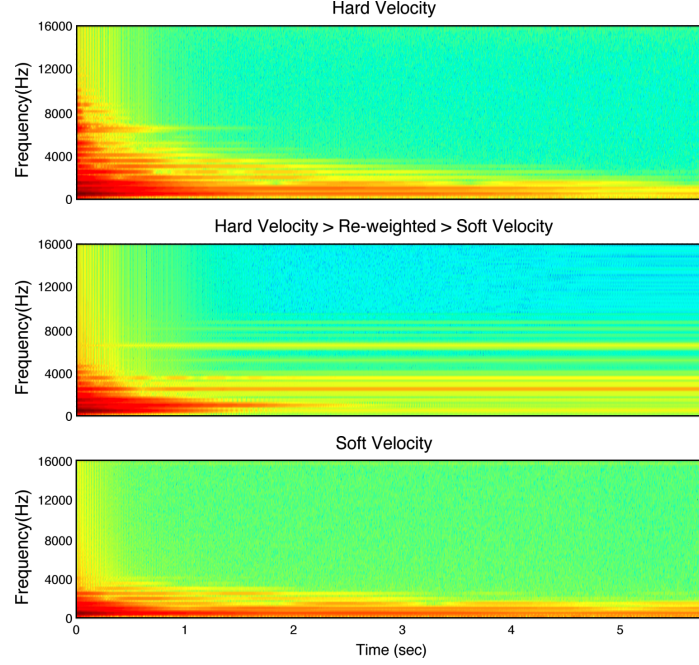


Figure 5.6: **Top:** Spectrogram of piano note B3 played with hard velocity. **Middle:** The same note with a re-weighted observation matrix to change the velocity. **Bottom:** The original sample of piano note B3 played with soft velocity.

5.1.6 Results

Figure 5.5 shows observation matrices for the note B3. The weighting coefficients used to generate \tilde{C} from $C\text{-Hard}$ are unity gain from DC up to the fundamental frequency, then linearly taper off to zero from the fundamental frequency to the fourth harmonic. This method essentially filters out frequencies that are present in the higher velocity note but not present in the lower velocity notes. Furthermore, this reduces the data required to represent different notes since only n weighting coefficients are required to transform the velocity of the note.

The spectrogram of the tones associated with each observation matrix shown in Figure 5.5 are shown in Figure 5.6. The general differences between the hard velocity (top) and soft velocity (bottom) are captured in the tone (middle) produced by re-weighting the observation matrix. There

are many mid range frequencies (~ 4000 Hz) that do not exhibit the decay characteristics that are observable in the original hard and soft velocity tones. This suggests that the evolution of the hidden states contributes to the decay characteristics of the frequencies and must be modified as well to more accurately produce a velocity transformation. The LDS is effective in capturing the evolution of a note on small time scales. To attempt to capture the dynamics and timbral characteristics on a larger time scale dynamic texture mixtures (DTM) which are probabilistic mixture models of linear dynamical systems are discussed.

6. Supervised Learning of Instrument Mixtures

This chapter discusses several approaches to combine multi-track sources using gain coefficients learned directly from data.

6.1 Weight Estimation

Using the dataset of RockBand stems and the mixed output audio described in Section 3.7.1, we do not have access to the exact fader values used to create the final output mix, therefore we must estimate these parameters in order to train a supervised machine learning model. The weight estimation process is subject to several unknowns including additional compression and equalization of the stem tracks on the game console in producing the final mix. We use our estimated weights as ground truth in a supervised machine learning task and estimate a series of weighting coefficients for each track from a set of acoustic features extracted from the audio.

The process of mixing multi-track source files down to a single track is a linear combination of the audio sources in the time domain

$$\alpha_{1t}u_{1t} + \alpha_{2t}u_{2t} + \cdots + \alpha_{kt}u_{kt} = v_t, \quad (6.1)$$

where $\{\alpha_{1t}, \dots, \alpha_{kt}\}$ are the mixing coefficients of the k tracks at time t and $\{u_{1t}, \dots, u_{kt}\}$ are the time domain waveforms of each track.

Since the Fourier transform is a linear operator, we assume that the spectrum of the final mix at frame n is a linear combination of the spectra of the source tracks at frame n . Considering a single frame in time, we have

$$\begin{bmatrix} U_{11} & U_{12} & U_{13} & \cdots & U_{1k} \\ U_{21} & U_{22} & U_{23} & \cdots & U_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ U_{N1} & U_{N2} & U_{N3} & \cdots & U_{Nk} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix} \approx \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_N \end{bmatrix} \quad (6.2)$$

$$\mathbf{U}\alpha \approx \mathbf{V},$$

where each column in \mathbf{U} is the magnitude spectrum of the k th track and \mathbf{V} is the spectrum of the final mix with a total of N frames in the song. We are careful here to note that Equation 6.2 is a

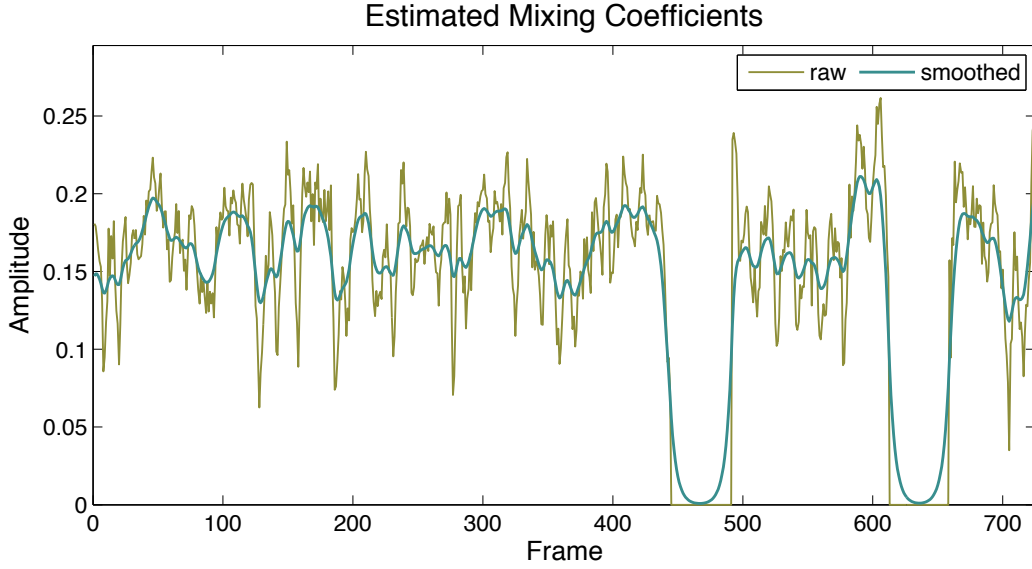


Figure 6.1: Extracted weights for bass guitar using NNLS, Kalman smoothing and normalization.

course estimation of the actual combination of tracks. Using magnitude spectra, the combination of tracks becomes

$$\alpha_1 \mathcal{F}\{u_1\} + \alpha_2 \mathcal{F}\{u_2\} + \dots + \alpha_k \mathcal{F}\{u_k\} = \mathcal{F}\{v\} \quad (6.3)$$

$$\alpha_1 U_1(e^{j\omega}) + \alpha_2 U_2(e^{j\omega}) + \dots + \alpha_k U_k(e^{j\omega}) = V(e^{j\omega}) \quad (6.4)$$

$$|\alpha_1 U_1(e^{j\omega}) + \alpha_2 U_2(e^{j\omega}) + \dots + \alpha_k U_k(e^{j\omega})| \approx |V(e^{j\omega})|, \quad (6.5)$$

As the number of tracks increases, the estimate of the weights becomes less accurate due to the interdependence of the α values.

Given a set of multi-track stems and the resulting audio produced by mixing the individual tracks, we can estimate the mixing coefficients, α_k , using non-negative least squares (NNLS) [44].

$$\hat{\alpha} = \min_{\alpha} \|\mathbf{U}\alpha - \mathbf{V}\|_2^2 \quad \alpha \geq 0 \quad (6.6)$$

We select NNLS to estimate the weights since the mixing process is additive by definition. Using unconstrained least squares, we experience both very large values for some weights since the algorithm can increase the weight of tracks that contain very little energy to reduce the overall error.

We perform this analysis on a frame-by-frame basis using a 1 second rectangular window and

overlap the frames by 0.75 seconds. In each frame, we compute the spectrogram of each individual track using a 1024 sample window with a 512 sample overlap. We vectorize and concatenate the spectrograms to attain the form given in Equation 6.2 then compute the weights. A resolution of 0.25 seconds for changing fader values is sufficient to capture the dynamic changes in each track.

To improve the initial estimate of the weights, we only include tracks that contain audio in the given frame. Assuming we have k tracks, if $\text{RMS}(u_{kt}) < 0.01$, then we negate the track in the estimate of the weight vector for the current frame and use $k - p$ tracks, where p is the number of inactive tracks. Removing these tracks prevents very large weight coefficients from being calculated for tracks that have very little energy, in addition to using NNLS as opposed to unconstrained least squares. The value of 0.01 was empirically determined to provide good peak suppression in the weight estimates.

We then process the weight vector using Kalman smoothing to reduce the noise that still remains in the signal [40]. The initial weight estimates as well as the smoothed weights are depicted in Figure 6.1. In the following section, we assume that the mixing coefficients are Gaussian when modeling the data. A histogram showing the distributions of mixing coefficients for multiple instruments is shown in Figure 6.2. It is significant to note that while these coefficients produce a mix that is perceptually very similar to the original track, they are not the actual ground truth weights. We provide online audio examples of the original song and the mix using the estimated weights.

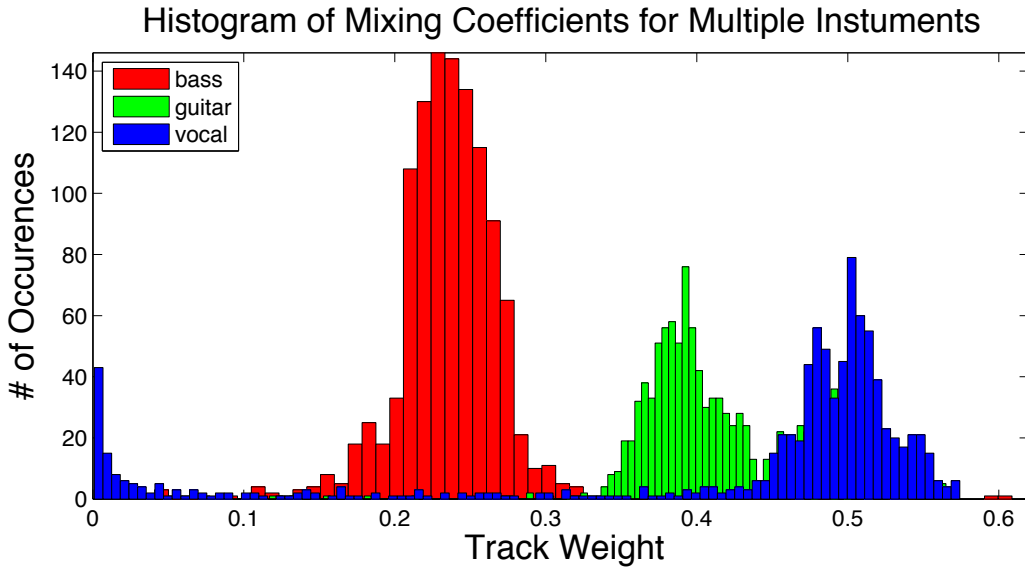


Figure 6.2: Histogram of linear mixing coefficients.

6.2 Modeling

We train two different models using acoustic features to predict the time-varying mixing coefficients for an unknown input song. We first use multiple linear regression (MLR) to find the projection from features to weights that minimizes error in the least squares sense. To model time dependence between the mixing coefficients of a given track, we use a linear dynamical system (LDS) and compute the latent states using Kalman filtering.

We extract a set of simple time domain and spectral domain features to train the models:

- Spectral Centroid
- Root Mean Square (RMS) Energy
- Slope/Intercept from fitting a line to the spectrum

A depiction of the overall system architecture showing the multiple modeling methods is shown in Figure 6.3.

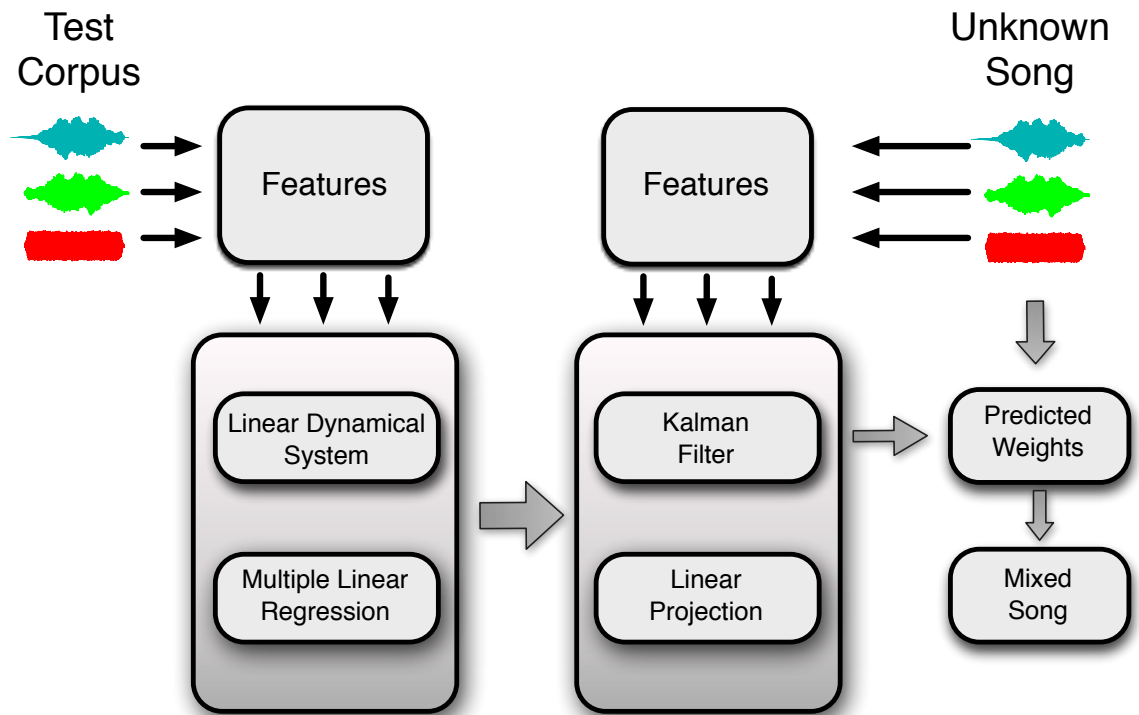


Figure 6.3: Supervised machine learning of gain coefficients using LDS and MLR.

6.2.1 Multiple Linear Regression

We assume that each weight vector α is a linear combination of our features $\{y_1, \dots, y_m\}$

$$\alpha = Y\beta \quad (6.7)$$

where Y is an $N \times M$ matrix, M is the number of features we have per frame, N is the number of frames and k indexes the track. We compute the projection matrix as in Equation 3.2 and use it to compute the weighting coefficients of an unknown song,

$$\hat{\alpha} = Y\hat{\beta}. \quad (6.8)$$

This model assumes that the mixing coefficients are independent with respect to time. In the next section we describe a model that considers the time dependence of the data.

6.2.2 Linear Dynamical System

We treat the time-varying mixing coefficients α as the latent states resulting from some noisy process and our features, y as noisy observations of the output of a linear dynamical system as described in Section 3.2.

For an unknown set of stems, we compute our acoustic features for each track and remove the training feature bias, \bar{y} . We then perform the forward Kalman recursions using the \mathbf{A} , \mathbf{C} , \mathbf{Q} and \mathbf{R} parameters learned during training to get an estimate of the weighting coefficients. Adding the weight bias $\bar{\alpha}$ to this result yields our final estimate of the mixing coefficients.

6.2.3 Results

Training and testing is performed in a typical manner for a supervised machine learning task. Given the relatively small size ($N = 48$) of the dataset we opt to use leave-one-out cross-validation, training on $N - 1$ songs and testing on the remaining song. This process is repeated for all N songs such that each is a test song only once.

We define \mathbf{Y}_{train} as a matrix formed by concatenating the features of all songs, and α_{train} as the matrix formed by concatenating all weighting coefficients for all songs. These quantities are then used to train the parameters of an LDS. We perform Kalman filtering on the remaining test song using the parameters learned in the training phase to estimate the time-varying weights for the

Track	LDS	MLR
backup	0.0126 \pm 0.0076	0.0091 \pm 0.0075
bass	0.0191 \pm 0.0183	0.0086 \pm 0.0102
drums	0.1452 \pm 0.1237	0.0590 \pm 0.0444
guitar	0.0158 \pm 0.0169	0.0075 \pm 0.0077
vocal	0.0188 \pm 0.0107	0.0149 \pm 0.0124

Table 6.1: Average mean squared error across all songs between ground truth weights and predicted weights for MLR and LDS.

song.

Figures 6.4 and 6.5 show the predicted and actual weights plotted on the same axis for each instrument in the song “Constant Motion” by Dream Theater. The resulting weights from MLR fit the data better and result in a lower error and the weights computed through Kalman filtering are much smoother yet sometimes exhibit bias or offset from the actual values. Table 6.1 shows the average mean squared error for all songs in the database for both algorithms.

Using a low dimensional feature set, we are able to generate a mix that is comparable to the desired result. Audio examples of the original mix, the drum sub-mixes and the reconstructed mix using the predicted weights can be found online at the previously specified link. A listening analysis performed by the authors finds that the LDS and MLR models yield very similar perceptual results. For comparison, we generated audio mixes using a simple averaging of all tracks. The result of this oversimplified model is hardly comparable to the results from the automatic mixing system. Although these results are good, we note that the weights estimated in Section 6.1 are not the true parameters. Additionally, the architecture is restricted by the definition of the state vector α . In our case, the states represent the weights associated with specific stems. In a real-world scenario, the number and types of instruments and tracks will vary for each song. This system would be incapable of handling an input of more or less than five tracks or from songs that do not contain the typical rock instrumentation. This limitation is addressed in the next section where the modeling topology is altered to accommodate a variety of inputs.

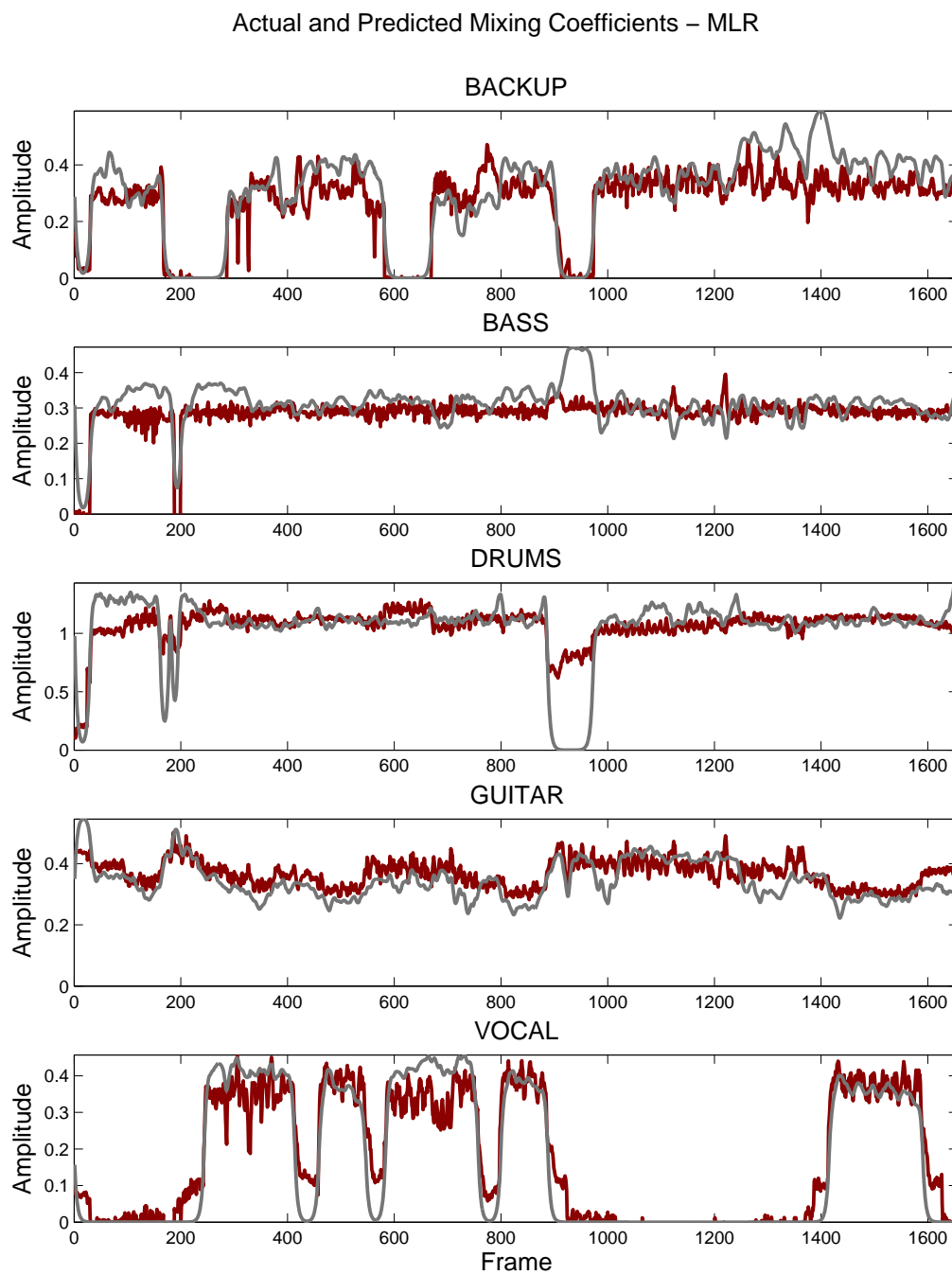


Figure 6.4: Results for weighting coefficient prediction using multiple linear regression (MLR). The estimated ground truth weights are shown in gray and the predicted coefficients are depicted in red.

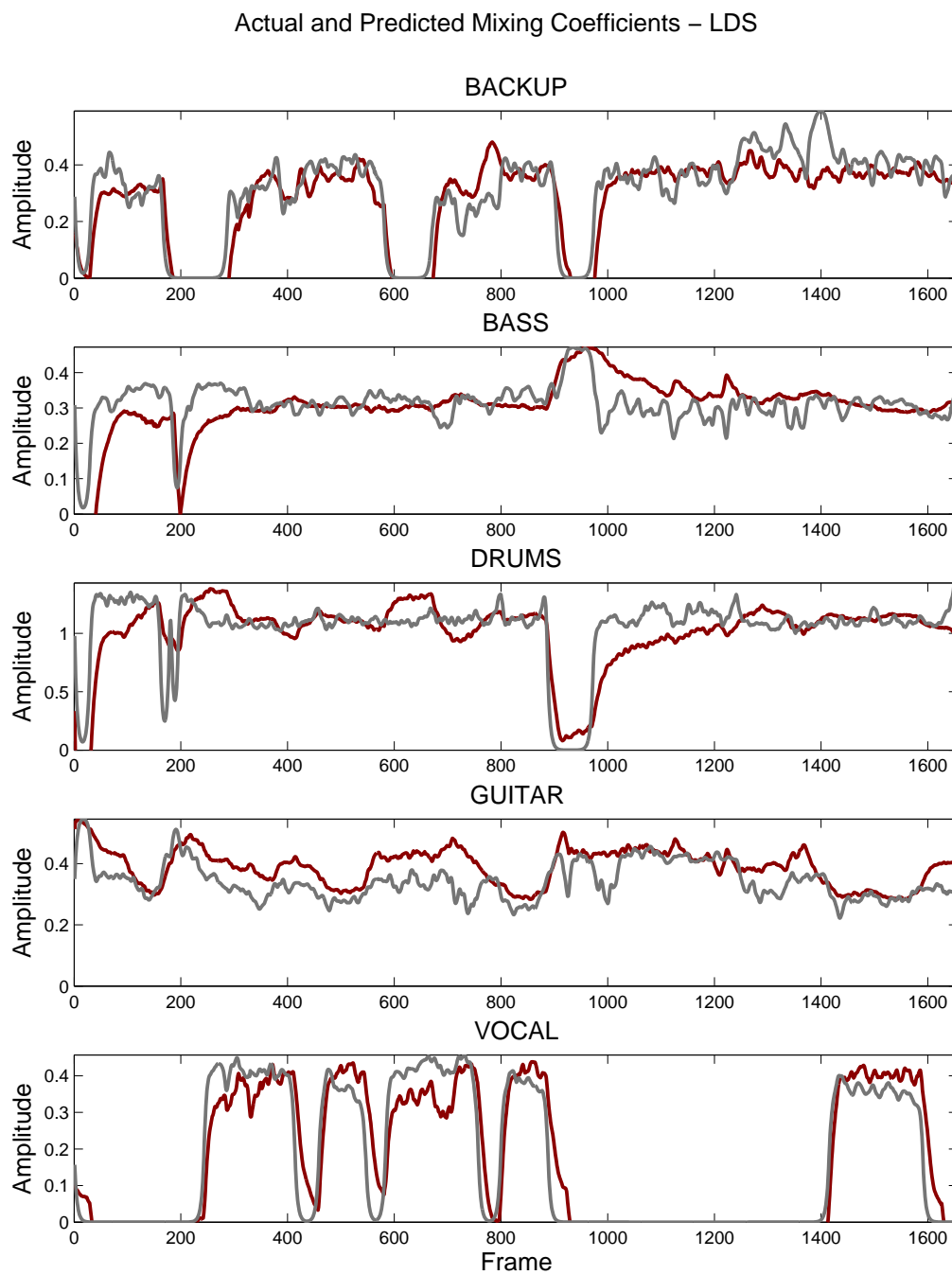


Figure 6.5: Results for weighting coefficient prediction using a linear dynamical system (LDS). The estimated ground truth weights are shown in gray and the predicted coefficients are depicted in red.

6.3 Improved Architecture

We again use the weights estimated in Section 6.1 as labels in a supervised machine learning task. Recall that our state vector is the weights of each instrument at time step t

$$\boldsymbol{\alpha}_t = [\alpha_1 \alpha_2 \dots \alpha_k]^T, \quad (6.9)$$

and is the limiting factor in our model. The structure of the output vector is

$$\mathbf{y}_t = \left[F_1^{(1)} \dots F_m^{(1)} F_1^{(2)} \dots F_m^{(2)} F_1^{(k)} \dots F_m^{(k)} \right]^T \quad (6.10)$$

where we have m features, F , for each of the k instruments in the mixture.

In this framework, we are constrained in terms of the number and type of instruments we can use the automatic mixing system for. Since each α_k is associated with a specific instrument, omitting or adding tracks changes the dimension of the hidden state vector and in turn makes predicting weights for a set of tracks that are not explicitly in the form described in (6.9) and (6.10) intractable.

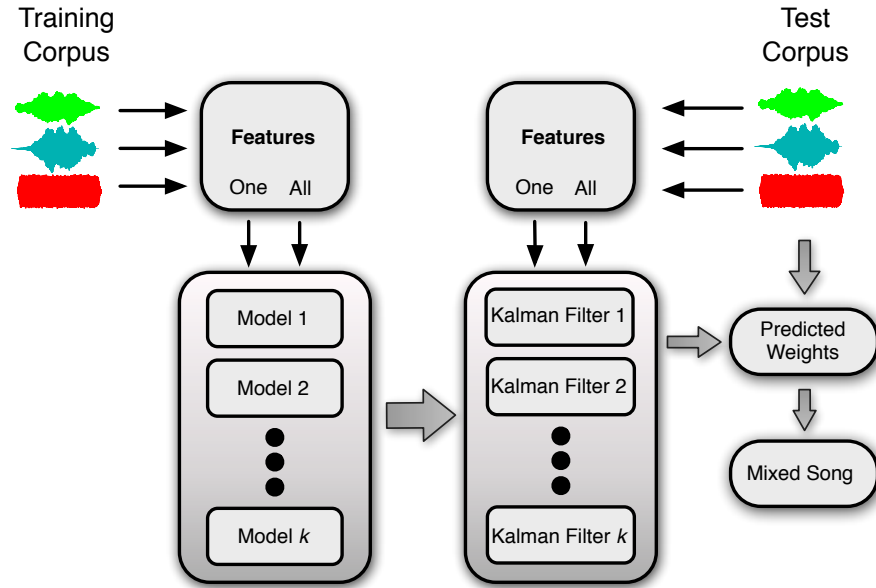


Figure 6.6: System diagram detailing the ‘One Vs. All’ method for mixing coefficient prediction.

Instead of modeling the time varying mixing coefficients of all tracks as the hidden states of the

Track	All Tracks	One Vs. All	Best Features
backup	0.0126	0.0110	0.0087
bass	0.0191	0.0163	0.0088
drums	0.1452	0.1283	0.0489
guitar	0.0158	0.0151	0.0115
vocal	0.0188	0.0160	0.0108

Table 6.2: Results for LOOCV on the database. The MSE for each track across all songs is shown for the All Tracks method and the One Versus All approach. The Best Features column is the result from sequential feature selection.

LDS, we consider only one instrument at a time. Our new state vector consists of the weight for the j th track and its first and second derivatives

$$\boldsymbol{\alpha}_t = \begin{bmatrix} \alpha_j & \dot{\alpha}_j & \ddot{\alpha}_j \end{bmatrix}^T \quad (6.11)$$

The derivatives of the weight vector are used to provide the model with more information about the dynamic evolution of the mixing coefficients. Note that only the weights for one instrument are included in the state vector. By eliminating the weight values of the other instruments, we are training the model to consider only how well the current instrument ‘sits’ in the mix, not how the weights of all instruments evolve together.

The output vector \mathbf{y}_t is comprised of the feature set for the instrument we are trying to predict stacked with the average of the features from all other instruments

$$\mathbf{y}_t = \left[F_1^{(j)} \dots F_m^{(j)} \frac{1}{K-1} \sum_{k \neq j}^K F_1^{(k)} \dots \frac{1}{K-1} \sum_{k \neq j}^K F_m^{(k)} \right]^T \quad (6.12)$$

If $j = 1$, then we are using m features associated with the first track and averaging the features associated with the tracks $k \neq j$, reducing the dimensionality of the feature vector from km to $2m$.

Comparing (6.10) to (6.12), we observe that in (6.12) there is no dependency on which position (k) the features for a given instrument are located. The only prior knowledge the model requires is the type of the j th instrument for which we are predicting time-varying weights. As a result, in this framework there is no limitation on the number or type of instruments that can be mixed using the system, provided that there exists training data for the target instrument j . A system diagram showing the new modeling method is shown in Figure 6.6.

To evaluate the efficacy of this modified estimation approach, we perform the same experiment

Feature	Description
RMS energy	Root mean square energy
Spectral flux	Change in spectral energy
Spectral bandwidth	Range of frequencies where most energy lies
Octave-based sub-bands	Energy in octave spaced frequency bands
MFCC	Mel-Frequency Cepstral Coefficients
Spectral centroid	Mean or center of gravity of the spectrum
Spectral peaks	Energy around a local sub-band maxima
Spectral valleys	Energy around a local sub-band minima
Slope/Intercept	Parameters of a line fit to the spectrum of a frame

Table 6.3: Spectral and time domain features used in mixing coefficient prediction task.

outlined in Section 6.2 and compare the results of the two methods. Using the 48 songs in our dataset, we perform leave-one-out cross-validation (LOOCV), training an LDS on 47 tracks and predicting the weights for the remaining track. We repeat the process using each track as a test song only once and average the mean squared error (MSE) between our estimated ground truth values and our predictions from the LDS. The results are shown in Table 6.2. We refer to the method described in Section 6.2 as All Tracks (AT) and the modified approach in this section as One Versus All (OVA). The OVA results are computed using the same feature set $\{\textit{centroid}, \textit{RMS}, \textit{slope}, \textit{intercept}\}$ that was used in the previous experiment.

The table shows an average improvement of 11.66% in terms of MSE for all instrument types in the dataset. The OVA method provides increased performance in terms of the MSE of the weight predictions as well as increased flexibility. The new topology enables the system to mix songs that do not have the same number of tracks as the normalized RockBand dataset we compiled.

6.3.1 Feature Analysis

Having shown that the OVA method outperforms the AT method, we proceed to investigate which features are the most informative. We explore an extended feature set within the framework described in the previous section and analyze the performance of each individual feature as well as combinations of features. Table 6.3 lists the array of spectral and time domain features we selected for our experiment [38, 18, 93]. The features are chosen to contain information about the total energy of the signal, energy within various frequency bands, spectral shape and dynamic spectral

evolution.

All experiments are performed using LOOCV on the entire dataset. In the first experiment, we test the performance of each individual feature using the average MSE over all songs as our error metric. Table 6.4 shows the results for each feature for each track type in the dataset. There is no single feature that appears to be dominant for mixing coefficient prediction.

Backup		Bass		Drums		Guitar		Vocal	
Feature	Error	Feature	Error	Feature	Error	Feature	Error	Feature	Error
Bandwidth	0.0511	Flux	0.0590	Centroid	0.7322	Bandwidth	0.0756	Flux	0.1183
Flux	0.0526	Bandwidth	0.0590	RMS	0.8415	Valley	0.0878	Centroid	0.1240
Sub-Bands	0.0580	Slope	0.0618	Slope	0.8713	Intercept	0.0908	Bandwidth	0.1251
Intercept	0.0587	Intercept	0.0622	Bandwidth	0.8861	Slope	0.0920	Valley	0.1262
Slope	0.0589	RMS	0.0716	Intercept	0.8932	Flux	0.0936	Peak	0.1302
Peak	0.0607	Valley	0.0741	Peak	0.9260	Sub-Bands	0.0974	Intercept	0.1316
RMS	0.0629	Sub-Bands	0.0743	Valley	0.9381	RMS	0.0987	Sub-Bands	0.1317
Centroid	0.0636	Peak	0.0752	Sub-Bands	0.9649	Peak	0.1019	Slope	0.1318
MFCC	0.0659	Centroid	0.0801	MFCC	1.1785	Centroid	0.1095	RMS	0.1320
Valley	0.0680	MFCC	0.0821	Flux	3.5767	MFCC	0.1127	MFCC	0.1373

Table 6.4: Mean squared error for all features and individual instruments. Features for each instrument are listed in order of best performance to worst performance. The best combination of features for each instrument is in boldface.

Using these results, we employ sequential feature selection to increase the performance of our system [51]. The best performing feature for each instrument in Table 6.4 is stacked with each remaining feature, and the MSE for LOOCV is computed for each combination. The best feature from this result is retained and the process is repeated until all features have been used. The results of this analysis are depicted in Figure 6.7. The best performing number of features for each instrument is indicated with a diamond. Since some of our features may contain similar information, adding additional features eventually becomes redundant and the increase in the size of the feature space outweighs the gain in information.

6.3.2 Results

The overall results for using the best performing feature ensemble are detailed in Table 6.2 under the column Best Results. The table shows that the OVA approach more accurately models the mixing coefficients and the addition of more features improves the results. Mean squared error does not provide any intuition about where each model fails or performs well. Figures 6.8 and 6.9 show comparisons between the AT and OVA models. Both models were trained with the feature set used

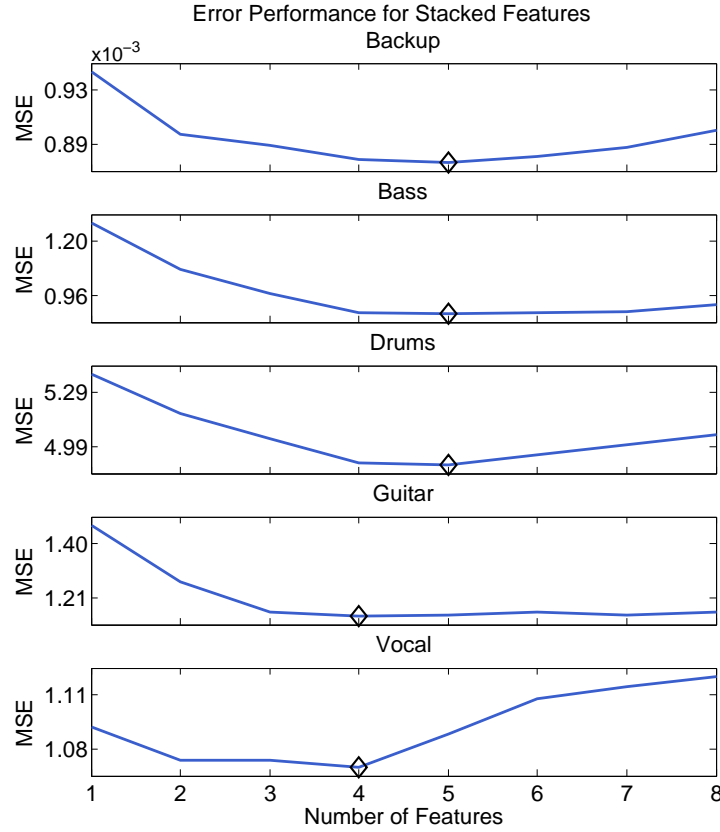


Figure 6.7: MSE versus the number of stacked features used in training an LDS for each track. Note that the scale of each sub-plot varies. The minimum is indicated for each track.

in Section 3.2. There is relatively small deviation in the bass and guitar predictions for each method on both songs. The most significant difference is in the ability of the OVA model to track the vocal weights as evidenced by the relatively flat predictions from the AT model contrasted with the OVA model predictions that follow the contour of the ground truth weights.

In Figures 6.10 and 6.11 we observe the effect of increasing the number of features used to train the model. The predictions using the best feature for each instrument from Table 6.4 are shown in gray and the highest performing ensemble of features is depicted in orange. Adding features creates the most improvement in the drum track where the contour and bias of the predictions closely follows the ground truth for both songs. Although this is only a small sample of the dataset, this representation informs us of improvements that can be made to the system.

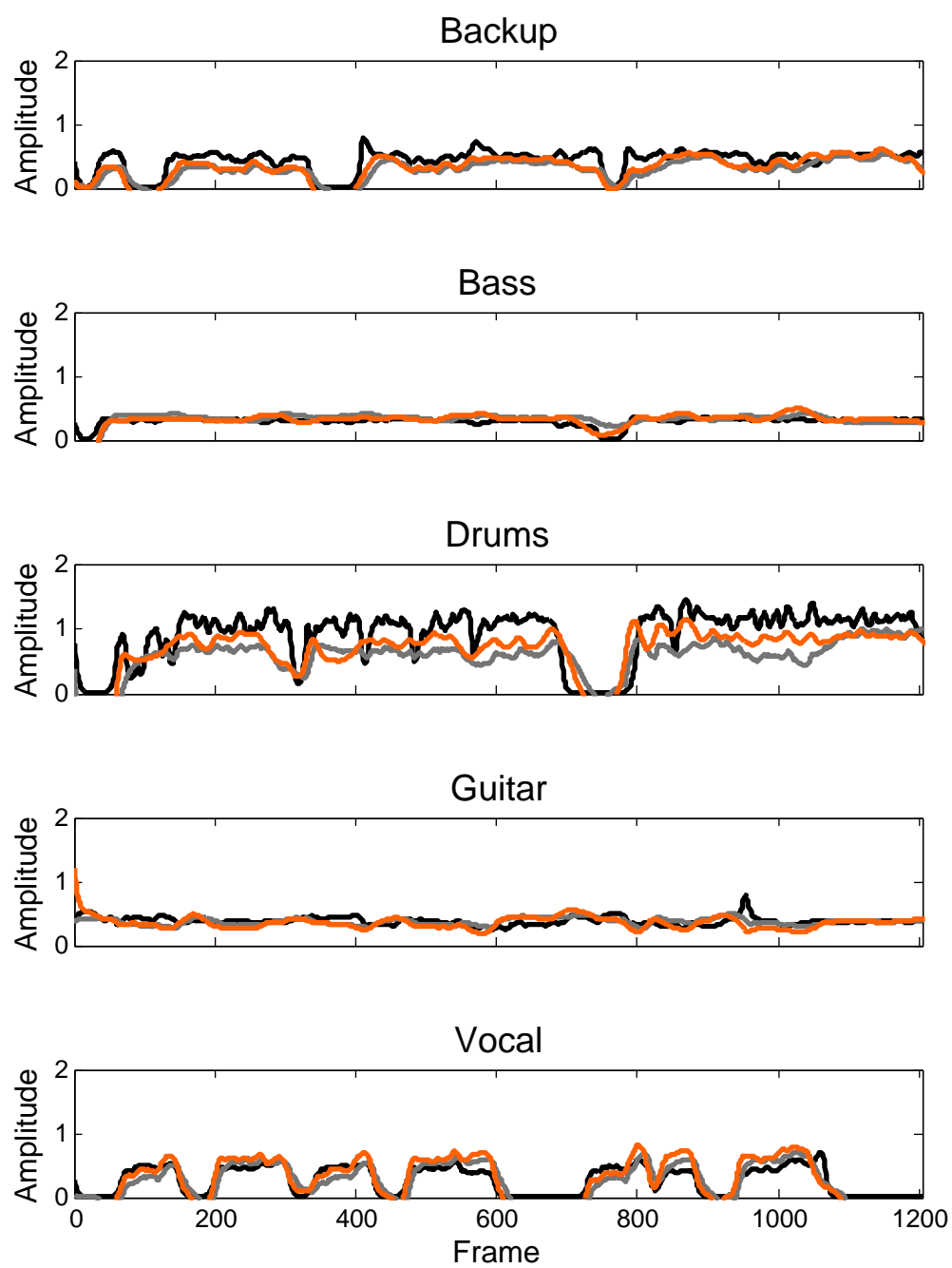


Figure 6.8: Comparison of ground truth (black) values with AT (gray) and OVA (orange) models for 'More Than A Feeling' by Boston.

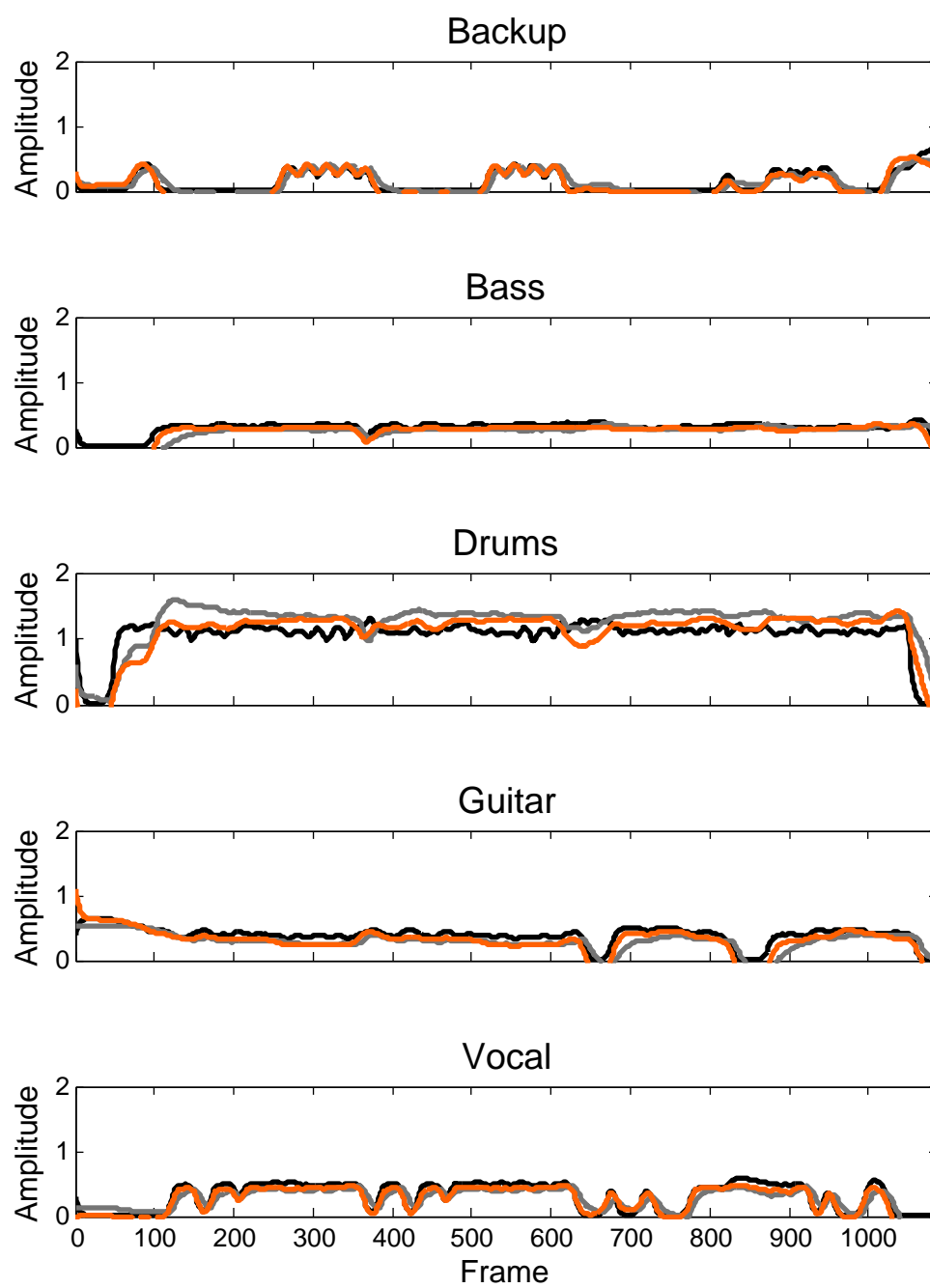


Figure 6.9: Comparison of ground truth (black) values with AT (gray) and OVA (orange) models for ‘Hammerhead’ by The Offspring.

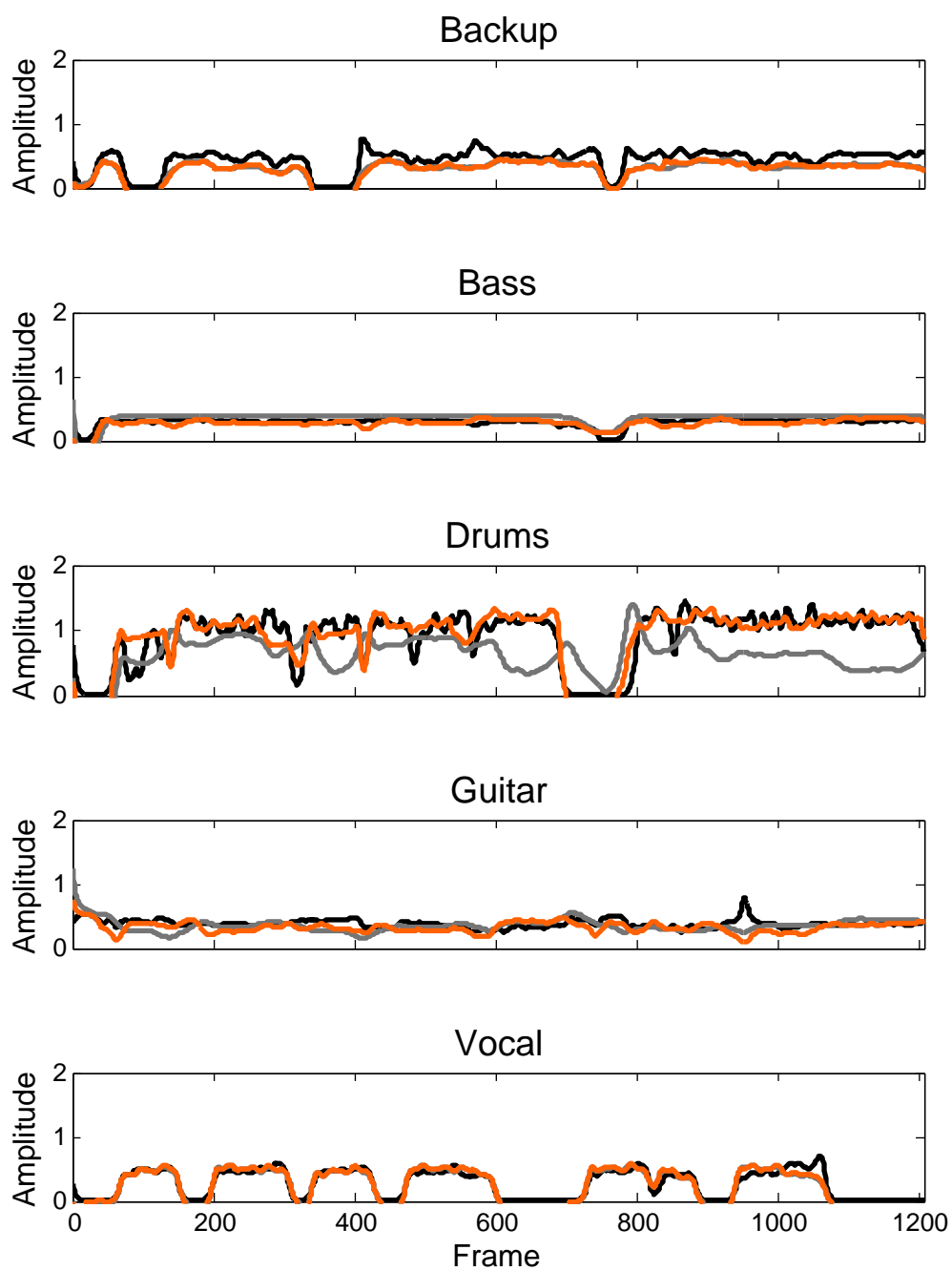


Figure 6.10: Comparison of ground truth (black) values with OVA model using the single best feature (gray) and using the best combination of features (orange) for ‘More Than A Feeling’ by Boston.

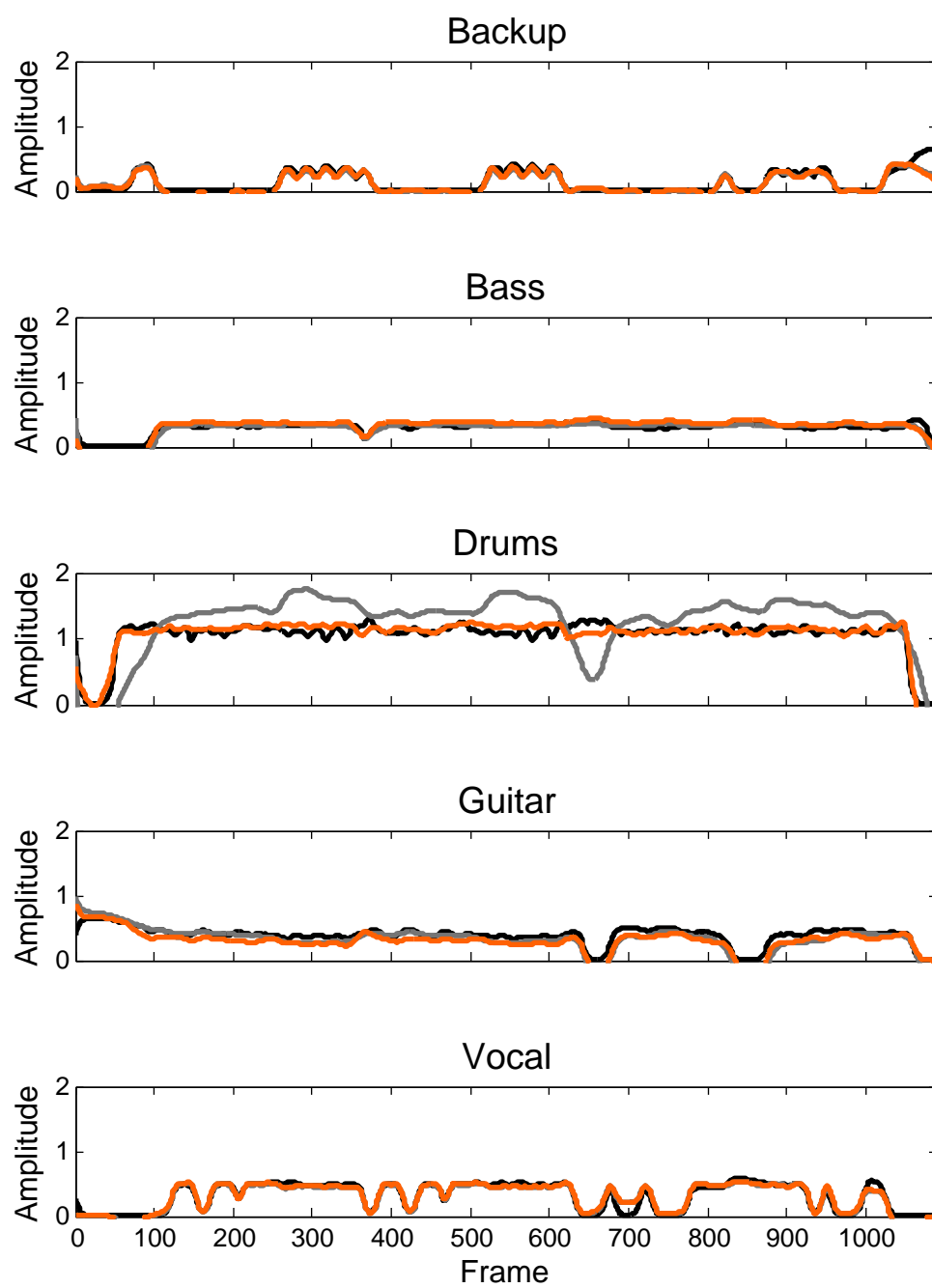


Figure 6.11: Comparison of ground truth (black) values with OVA model using the single best feature (gray) and using the best combination of features (orange) for ‘Hammerhead’ by The Offspring.

7. Perceptual Evaluation of Features

In Section 2.4, a set of experiments performed for evaluating features based on perceptual measures was shown. This work was the first of its kind to attempt to quantify how well the features that researchers use in perceptually motivated experiments in Music-IR actually equate to the human observation. In this case, music emotion recognition was the specific domain. This chapter presents a set of experiments in a similar vein with instrument (timbre) recognition as the goal and basis of evaluation.

7.1 Feature Extraction

The feature extraction process outlined here is developed to reduce computational load, provide a more compact representation and approximate some of the known effects of the auditory system. We first downsample our monaural audio files to 22,050 Hz and compute the Short Time Fourier Transform (STFT) of the signal using a hanning window with a frame size of 46.4 ms (1024 frames). We compute a 1024 point Discrete Fourier Transform on each frame and employ a hop length of 23.2 ms (512 samples) between frames. This provides us with a frequency resolution of 21.49 Hz in our STFT.

7.1.1 Auditory Model

Computational auditory models are derived from the physical characteristics of the human auditory system. One key aspect of auditory processing is the logarithmic organization of hearing induced by the basilar membrane [52]. As explained in Section 2.1.1, different sections of the basilar membrane respond differently to various frequencies, leading to dynamic frequency sensitivity over the range of human hearing. The critical band is the bandwidth of an auditory filter that roughly defines the range in which another sound will cause masking and effect the perceived loudness of the sounds. The bandwidth is defined in terms of the center frequency of the filter since sensitivity to frequency changes decreases with increase in frequency. The filter channels have approximately equal energy and are spaced logarithmically over the range of human hearing. Not only does the critical band filterbank provide a rough approximation of the way the auditory system works, it also reduces the dimensionality of the input data to the system. A comparison of the spectrogram (513

dimensions) and the output from a 108-band critical band filterbank is shown in Figure 7.1.

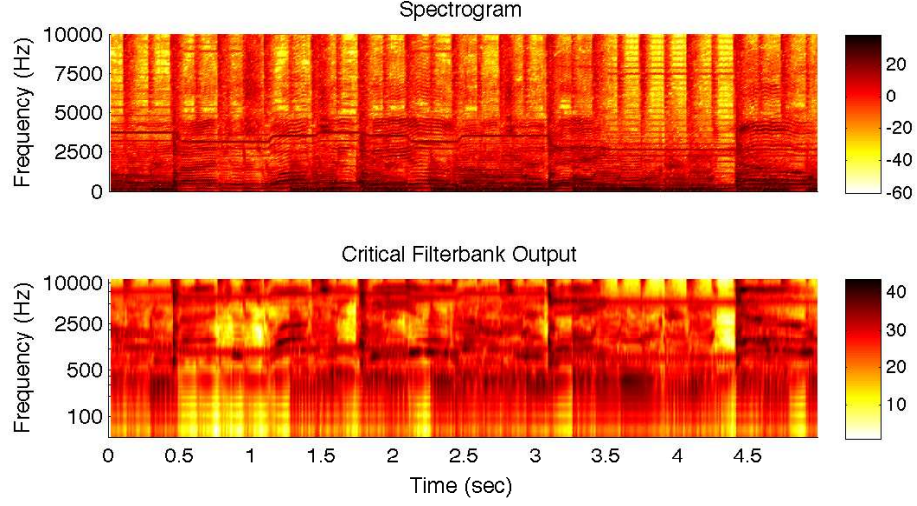


Figure 7.1: Log frequency spectrogram and critical band filterbank outputs of the song *No Phone* by Cake.

More commonly used throughout the signal processing and Music-IR literature is the mel scale. The mel scale is based on experiments that attempt to measure the perceptual nature of pitch [16]. In experiments, listeners were asked to adjust the frequency of tones so that each tone was twice as high in pitch as another. The results derived from the data form the relationship between the Hertz and mel scales. The transformation between the two values is similar to a log scale which is intuitive due to the logarithmic organization of pitch in the human auditory system. A mel scale filterbank for converting a linear frequency spectrum to the mel scale is shown in Figure 7.2. There are 128 mel spaced filters used between 20Hz and 11025Hz. The x axis is logarithmic in frequency and shows that the mel filters are not truly logarithmically spaced. There is an inflection point around 1kHz where the spacing below is much wider than the spacing above this point showing the inconsistencies between a true logarithmic scale and mel scale.

7.2 Basis Decomposition of Spectral Representations

We explore several methods of representing our frequency domain transformations of the data that are commonly used throughout the Music-IR literature. Principal Component Analysis (PCA), Non-Negative Matrix Factorization (NMF) and Probabilistic Latent Component Analysis (PLCA)

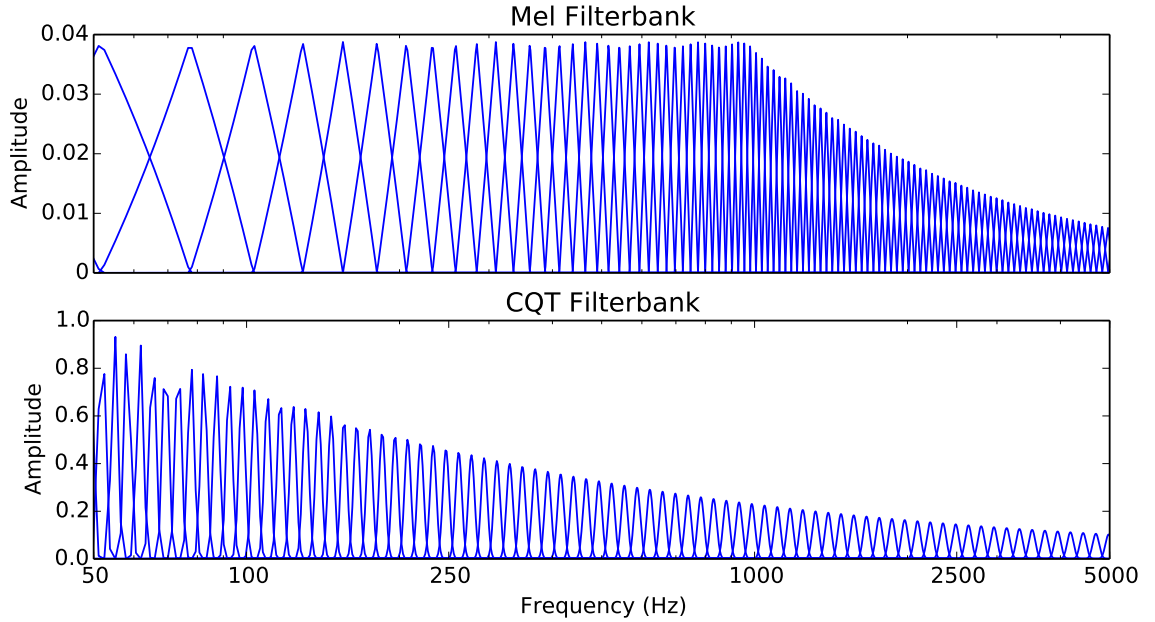


Figure 7.2: Mel and constant-Q filterbanks depicted in log frequency scale

decompose a matrix into a set of basis vectors or components and activations that are then used to reconstruct the data. Often, a reduced set of components (fewer than the dimensionality of the data) are used to capture relevant statistical information in the data or perform dimensionality reduction. Our goal in applying these methods is to generate a set of representative functions that can capture aspects of timbre.

The general experimental framework is depicted in Figure 7.3. We compute the STFT for the produced mix of a song in the dataset and then apply the perceptual weighting filterbank to produce a mel scaled spectrogram or constant-Q transform. Then the matrix is decomposed into a set of components and activations (top). We detail the algorithms used for this decomposition in the following sections. Once a decomposition is attained, the bases is used to reconstruct the individual tracks that form the mixture. The perceptually weighted STFT of each individual track is computed and then represented in terms of the bases trained on the full mixture. A track is transformed into the space with reduced dimensionality defined by the basis decomposition and then projected back into the original perceptually filtered time-frequency domain. We use the same number of latent components for each model and then measure their ability to capture aspects of timbre through perceptual listening tests.

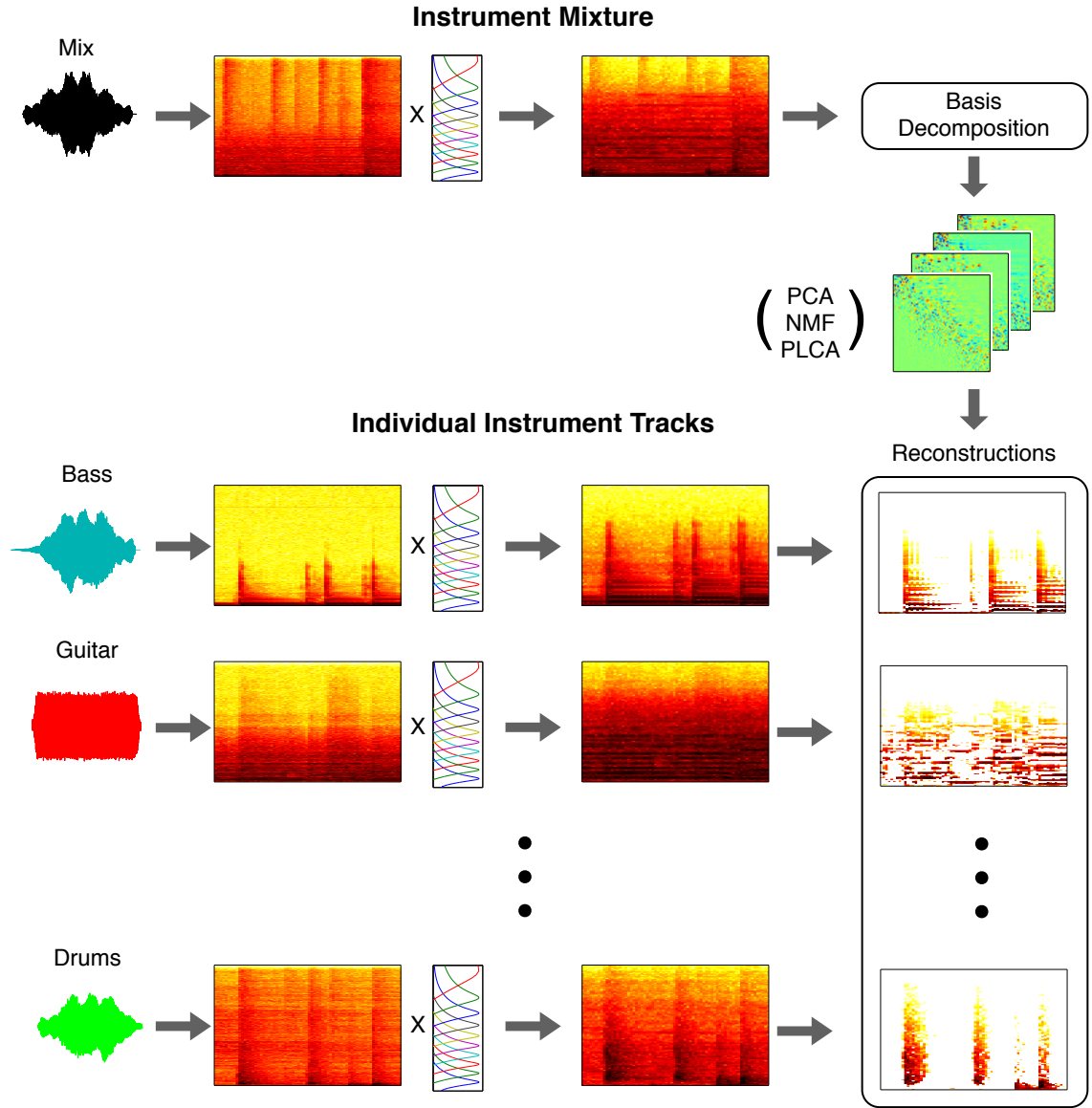


Figure 7.3: The spectrogram of an instrument mixture is perceptually filtered and a set of bases functions are computed using PCA, NMF and PLCA. The resulting bases from the mixture are used to reconstruct the individual instrument files.

7.3 Listening Evaluation of Timbre Reconstructions

The goal of the experiment is to determine which of the basis decomposition methods presented in the Section 7.2 is able to capture the most relevant spectro-temporal characteristics across various instrument sources. For each song in the dataset we compute a 128 bin mel spectrogram from 20 to

11025 Hz. Next, following the diagram in Figure 7.3, for each of the methods (PCA, NMF, PLCA) we reconstruct the audio of the individual tracks in the mixture from the bases computed on the mixture. For example if a song contains bass, drums, guitar, vocals and piano. We would compute the bases from the final produced mixture (converted to monaural) and use those to individually reconstruct the bass, drums, guitar, vocals and piano tracks. The bases computed on the mixture will capture the most relevant qualities based on the statistical formulation of each method.

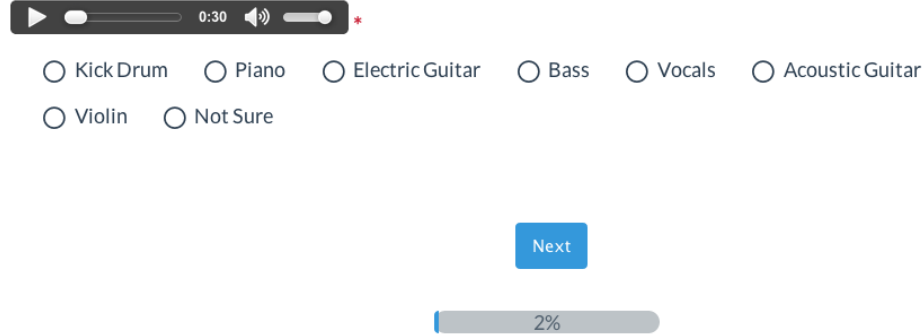
Instrument Type	Number of Examples
Acoustic Guitar	3
Bass	5
Electric Guitar	3
Kick Drum	3
Piano	1
Violin	1
Vocals	4

Table 7.1: Number and type of instruments used in the reconstruction listening experiment.

Reconstructing the individual tracks will show whether the information captured in the representation is relevant for the perception of timbre. We design a randomized listening test where participants are asked to identify the instrument in the mixture. We select five songs from the dataset and use four from each song as our query examples. Each track has a reconstruction for PCA, NMF and PLCA resulting in a total of 60 questions for each participant. The instruments represented in the dataset are shown in Table 7.1.

The participants are asked the following question “One of the following instruments is present in this audio clip. Which one is it?”. Then they are presented with the 7 instrument choices found in the subset of the data used for this evaluation. A screenshot of the survey question is shown in Figure 7.4, notice the addition of the ‘Not Sure’ category in the instrument choices. After an informal listening analysis, it was evident that some of the reconstructions were inconclusive in terms of which type of instrument belonged to the audio source present in the clip. This option is included to gain insight into what qualities are being captured (or not) by the basis decompositions. Given the inclusion of this option, the participants were instructed “Each clip only has one instrument. Make your best guess as to which instrument is contained in the audio example. If the clip bears no

52. One of the following instruments is present in this audio clip. Which one is it?



☐ Kick Drum
 ☐ Piano
 ☐ Electric Guitar
 ☐ Bass
 ☐ Vocals
 ☐ Acoustic Guitar
 ☐ Violin
 ☐ Not Sure

Next

2%

Figure 7.4: An example question from the perceptual survey asking participants to identify the instrument present in the audio reconstruction.

Decomposition	Accuracy
Original Audio	0.97
Mel Spectrogram	0.84
PLCA	0.73
PCA	0.40
NMF	0.32

Table 7.2: Number and type of instruments used in the reconstruction listening experiment.

resemblance to any instrument then select *Not Sure*”.

7.3.1 Results

There were a total of $N=27$ participants between the ages of 18-34 with 5 females and 22 males. Of the participants, 12 reported more than five years of musical training, six reported 1-5 years of training and nine reported less than one year. Eight people reported that they use a DAW *often*, four reported *sometimes* and the rest had no experience.

The overall accuracy for each decomposition type is displayed in Table 7.2. It is interesting to note that there is not a 100% recognition rate for the examples that were presented in their original CD quality audio format. The mel spectrogram result is the accuracy for the reconstruction from the unprocessed mel spectrogram. At 0.84 this number represents the upper bound of the recognition

rate for the participants to identify the instruments. PLCA performed significantly better than both PCA and NMF in terms of mean accuracy across all examples and participants in the dataset, nearly double that of the next closest algorithm, PCA.

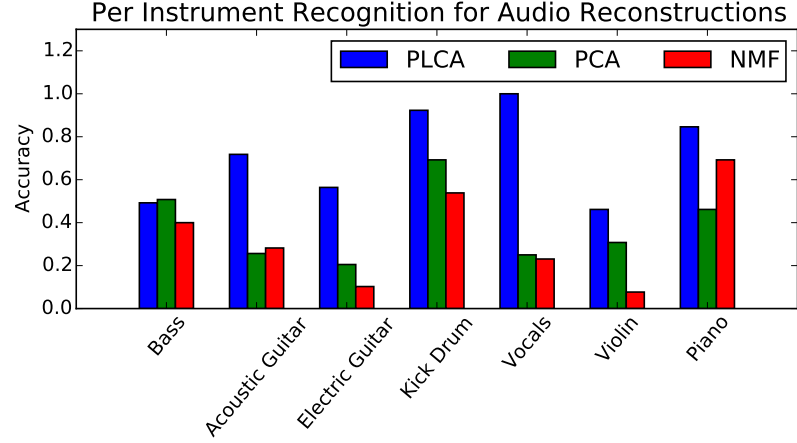


Figure 7.5: Listening test results showing the number of correctly identified instruments based on reconstruction type.

In the bar graph in Figure 7.5, we see the breakdown in accuracy given by instrument type. The PLCA model performs much better than NMF or PCA on vocal tracks. Listening to the examples reveals that the high frequency and ‘breathy’ content in the vocal reconstructions is not discarded as in the NMF and PCA reconstructions.

The bar graph of respondents selecting ‘Not Sure’ in Figure 7.6 indicates the difficulty that individuals have in ascribing a label to some of the reconstructions, in particular the vocals, which have complex frequency content and are often dynamic in terms of pitch whereas the other instruments less frequently utilize glissando.

For each example in the dataset a value $\{0, 1\}$ is assigned based on the correct (1) or incorrect (0) labeling of the instrument in the audio clip. The null hypothesis that the two samples come from the same distribution is performed to test for statistical significance in the findings. Since we are dealing with dichotomous data from matched pairs we apply McNemar’s test. Using $\frac{\alpha}{2} = \frac{1}{2}0.05 = 0.025$ to test for significance, we found a value of $p_{plca,pca} = 6.7 \times 10^{-20}$ for comparing the PLCA to PCA results and $p_{plca,nmf} = 2.1 \times 10^{-15}$ for the PLCA and NMF results. The McNemar’s statistic the NMF and PCA decompositions did not refute the null hypothesis with $p = 0.028$. Although this test has a

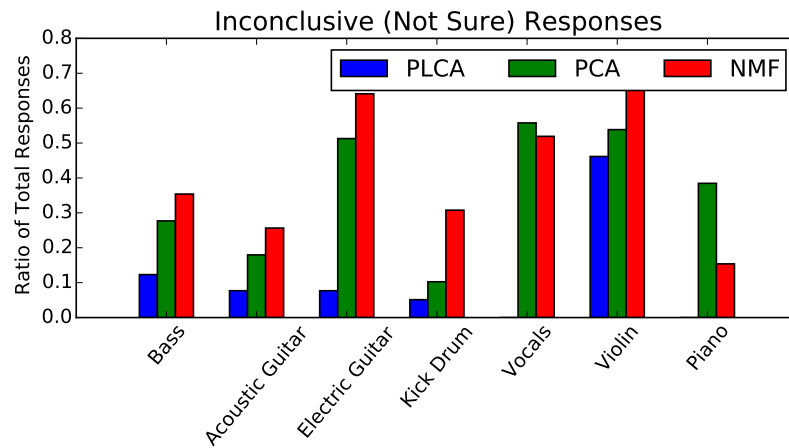


Figure 7.6: Listening test results showing the number of respondents expressing inability to determine instrument class for each reconstruction type.

relatively small sample size, the results strongly show the increased ability of a two-dimensional basis to capture aspects of the audio that are relevant to perception of timbre. When designing features for systems that are supposed to model the way humans process and analyze audio, it is important to ensure that what the system is ‘listening’ to contains information relevant to the problem at hand.

8. Conclusions and Future Directions

Several different research areas have come together in this work, precipitating a cogent exploration of multi-track instrument processing and modeling. The research presented here explored several different facets of instrument mixtures, with the following main contributions:

- Methods were developed for learning mappings from acoustic features to instrument mixture parameters. Regression techniques and state space models (LDS) were shown to be efficacious for producing good mixture results for unknown tracks. Although the LDS proved to be effective at modeling the mixing parameters learned from the data, the topology hindered modeling a variety of data. A one versus all architecture was employed that both improved overall accuracy and provided the benefit of being more forgiving to instrumentation.
- A corpus of instrument tones was represented using linear dynamical systems and then re-synthesized, showing the capability of the model to capture and alter perceptual characteristics.
- Individual instrument examples were reconstructed from features commonly used in Music Information Retrieval. It was shown that the salient aspects of the audio signal that humans use to perceive individual timbres are lost in many commonly used approaches, namely PCA and NMF. The results show that 2-D representations (convolutive PLCA) are much more perceptually relevant in a computational framework.

I introduced a supervised machine learning approach for automatically mixing a set of unknown source tracks into a coherent, well-balanced instrument mixture using a small number of acoustic features. The mixing coefficients were modeled as the hidden states of a linear dynamical system and used acoustic features extracted from the audio as the output of the model. After estimating the parameters of the model on the training data, the time-varying weights of each instrument for an unknown song were predicted using Kalman filtering.

That approach was extended to reduce the constraints on the model and generalizing it to a larger number of instruments. One modification to the system includes modeling the weights of an individual instrument and their first and second derivatives instead of jointly estimating the weights for all of the instrument tracks at once. This removes the restriction that the test song must contain all instrument types that the model was trained on.

Additionally, an extended feature set within this framework and evaluation of the performance of each individual feature as well as combinations of features was executed. The features are chosen to contain information about the total energy of the signal, energy within various frequency bands, spectral shape and dynamic spectral evolution.

Individual instrument tones were shown to be well modeled and re-synthesized using linear dynamical systems. The reconstructions produced good numerical error results and informal listening yielded quality audio examples. The ability to alter the timbre using a single model led to an exploration for salient features that are desirable for identifying timbres.

The ability to represent timbre in a set of reduced dimensionality components was evaluated through basis decomposition reconstructions. Three methods, Principal Component Analysis, Non-Negative Matrix Factorization and convolutive Probabilistic Latent Component Analysis were compared through a listening evaluation. Model components were trained on fully produced mixes and used to reconstruct the individual tracks to investigate whether interactions between various sources would be represented in the bases functions. The two dimensional kernel distributions and sparse impulse distributions were able to capture much more of the spectral evolution of the instrument sources.

Listening tests showed not only better accuracy per instrument but also more confidence in the participants' ability to discern whether or not the source was derived from an instrument at all. The frequency selectivity of the PCA and NMF decompositions prevented them from capturing the spectral contour of the signal mixture and resulted in emphasizing frequency content that was present in the mix but not in the separate source tracks.

8.0.2 Future Research

There are many directions to follow from the work presented in this thesis as the interdisciplinary nature of the work relies on several domains.

Supervised Models for Mixing

As is often the case in machine learning, more data is better but more clean data is best. More multi-track data *is* becoming available and the popularity of digital tools for creating and producing music as well as the collective nature of internet collaboration will surely provide more sources of data for training mixing models.

As many mixes are dynamic in nature with parameters varying over time, models that take

time dependence into account will be necessary. Hidden Markov Models (HMM), Linear Dynamical Systems (LDS), Dynamic Bayesian Networks (DBN) and Recurrent Neural Nets (RNN) are worth investigating for use in training models directly from data.

The evaluation of models will be much easier to accomplish if there is a concrete target model based on parameters from actual mixing sessions rather than having to rely on expensive and/or time consuming perceptual evaluations of resulting mixtures. This will also allow for more rapid iteration and improvement.

Timbre Modeling

Music is inherently time dependent. The majority of the efforts in the Music-IR community thus far have used the bag-of-frames approach assuming independence between frames or computing statistics over large amounts of time. This made sense due to the fledgling nature of the field and the complexity of the tasks involved. As the field has grown we have witnessed a focus on developing models that capture dynamics in a much more powerful way.

Time-frequency basis representations contain much more information that is perceptually relevant as was shown in this work. Hopefully these experiments will lead to a better understanding of how to model the interaction of different sources. The training and reconstruction framework here could easily be inverted where individual instrument bases are used to form a mixture and component activations could be informative about the contributions of separate instruments in the mixture. Similar perceptual analysis of the information that the model selects can help the field develop features that allow us to represent higher levels of abstraction leading to increased ability to model and perform more complex tasks in a more automated fashion.

Appendix A. Calculation of Mel Frequency Cepstral Coefficients

To compute MFCCs the frequency spectrum is first warped to the Mel scale which is a non linear scale that models human auditory perception [94]. This transformation, depicted in Figure A.1(a), is calculated as

$$F_{mel} = 2620 \log_{10} \left(1 + \frac{f}{657.6} \right) \quad (\text{A.1})$$

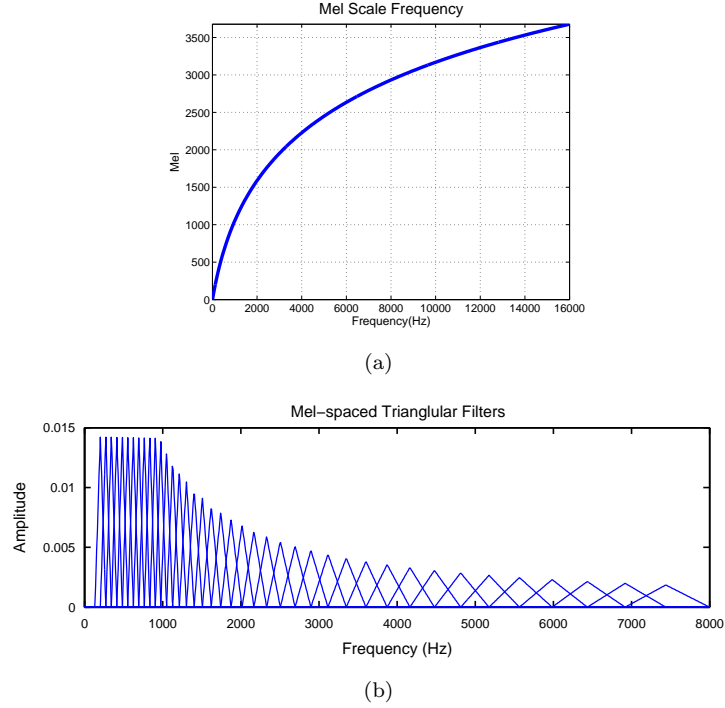


Figure A.1: The mel frequency scale (a) and mel filterbank (b).

The energy in the mel-scaled spectrum is then computed for a number of sub-bands as

$$E_{mel}(n, l) = \frac{1}{A_l} \sum_{k=L_l}^{U_l} |V_l(\omega_k) X(n, \omega_k)|^2 \quad (\text{A.2})$$

where $V_l(\omega)$ is the l th mel scale filter and $X(n, \omega_k)$ is the spectrum where n indicates the audio frame in time. The discrete cosine transform (DCT) of the log of the filter-bank outputs is calculated

yielding the MFCCs [68].

$$C_{mel}[n, m] = \frac{1}{R} \sum_{l=0}^{R-1} \log\{E_{mel}(n, l)\} \cos\left(\frac{2\pi}{R} lm\right) \quad (\text{A.3})$$

A diagram of this procedure is illustrated in Figure A.2.

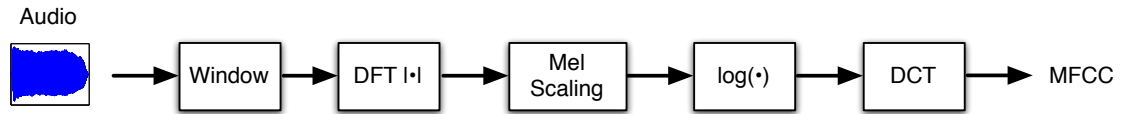


Figure A.2: MFCC calculation.

Appendix B. EM Algorithm

The mixture density for a weighted linear combination of Gaussians is given as

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (\text{B.1})$$

and the corresponding log likelihood function is given by

$$\ln p(\mathbf{x}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \quad (\text{B.2})$$

Taking the derivative with respect to $\boldsymbol{\mu}_k$ and equating it to zero gives

$$0 = \sum_{n=1}^N \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) \quad (\text{B.3})$$

where

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \quad (\text{B.4})$$

are the responsibilities (posterior probabilities) of the mixture model. Multiplying both sides by $\boldsymbol{\Sigma}_k$ and solving for the mean yields

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n, \quad N_k = \sum_{n=1}^N \gamma(z_{nk}) \quad (\text{B.5})$$

Similarly differentiating the log likelihood function with respect to $\boldsymbol{\Sigma}_k$ and following a similar line of reasoning yields

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \quad (\text{B.6})$$

The algorithm alternates between expectation and maximization steps until convergence. Initial parameters are often obtained through the k-means algorithm described in Section II. The E step maximizes the responsibilities, or posterior probabilities $\gamma(z_{nk})$. These values are used in the M step where the means, covariance matrices and mixture coefficients are calculated using the new posteriors.

Appendix C. Album Effect on Feature Data

Section II explained that the ground truth data used to develop the timbre models was derived based on a “same artist - same album” approach to similarity. An adverse effect of this approach has been observed in artist identification systems. When songs from the same album are used for testing and training data, the performance of these systems increases [43]. When the testing and training sets are mutually exclusive with respect to a given album, a significant performance degradation occurs.

The effect is attributed to post processing and mastering applied across all songs on an album. When equalization is applied to all songs, the frequency domain characteristics (i.e. the spectral envelope) are modified in the same manner resulting in a global change in spectral shape. Recalling that MFCCs are an approximation of the spectral envelope, it follows logically that a global change to the spectrum of the audio would have a normalizing effect and cause songs to be easily classified by a system trained on data from the same album.

To investigate the consequences this may have on the timbre model outlined in Section II, a brief experiment was performed. Note that some of the same problems of subjectivity in conducting this experiment could be alleviated by the type of study outlined in Section V. The steps performed are outlined below:

- Select albums that have a consistent timbre excepting at least one song that is drastically different.
- Select five tracks from the albums - four that are consistent in timbre and one that is markedly dissimilar.
- Take a 30 second clip from each song and compute MFCCs on a frame-by-frame basis.
- Generate a plot showing a projection of the data into a 3D space.
- Find a song that sounds similar to the consistent timbre of the album and plot it in the same feature space.

The albums used in this experiment were *Pork Soda* by Primus, *Pressure Chief* by Cake and *The War on Errorism* by NOFX. In each plot in Figure C.1, the dark blue is the song that was selected

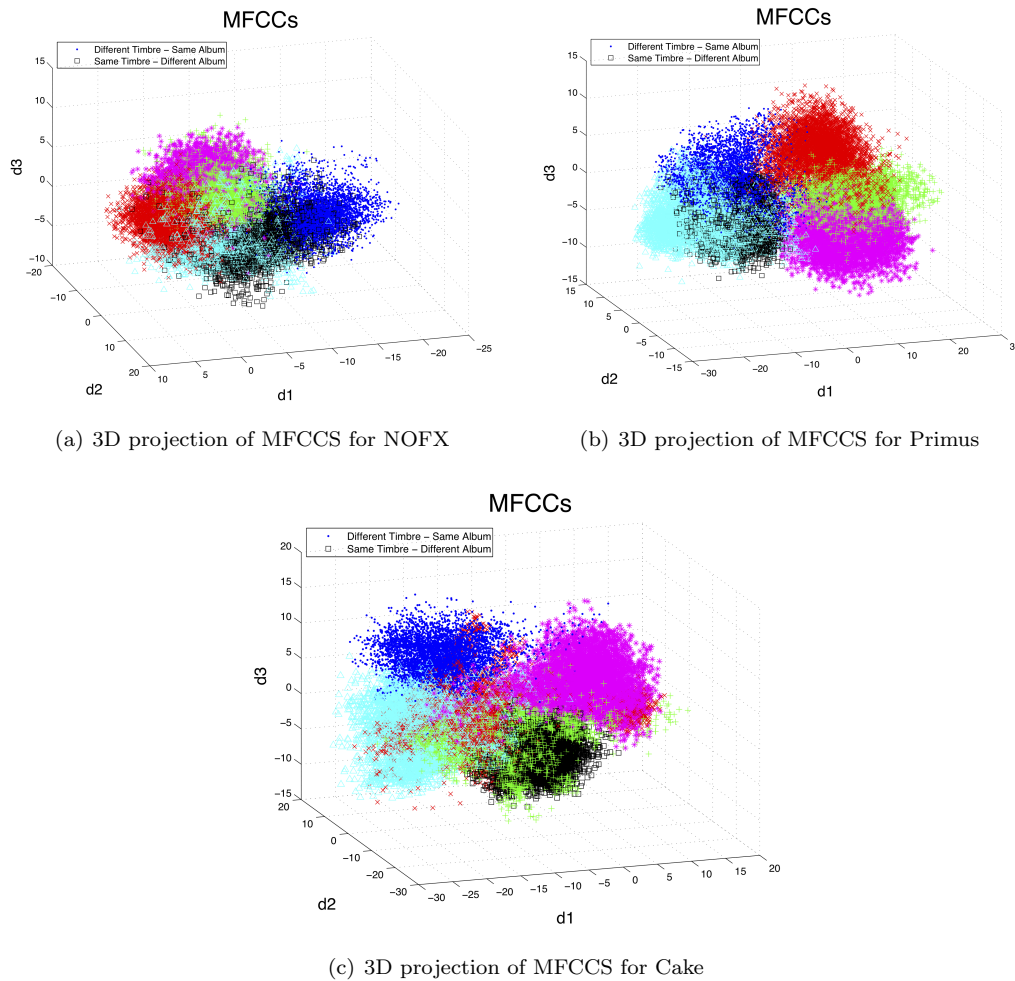


Figure C.1: MFCCs for 30 seconds of audio for several songs per album.

because it is significantly different in timbre compared to the other songs on the album. The black song is the test song that was selected from another artist and album deemed to have similar timbre. In the three plots, the song that is a different timbre from the same album is not separated very far from the remaining songs on the album. This may indicate that the post production does shift a dissimilar timbre towards the remainder of the tracks on the album. The tracks selected from the test album do not show much separation from the training samples as would be expected from a song selected to sound the same.

This short experiment would benefit greatly from data gathered from many individuals regarding the similarity of timbre in the ground truth data. It is possible that in selecting the songs for the experiment the researchers bias is heavily influenced by musical taste and other factors. A much

more convincing model can be developed if a majority of participants in a study rank the songs by similarity. This is a much more acceptable measure of ground truth compared to the opinion of one or several individuals working on the project.

Bibliography

- [1] American Standards Association and Acoustical Society of America. *American Standard Acoustical Terminology*. ASA Standard. American Standards Association, 1960.
- [2] J-J Aucouturier and Francois Pachet. Improving timbre similarity : How high's the sky ? *Journal of Negative Results in Speech and Audio Sciences*, pages 1–13, Apr 2004.
- [3] Daniele Barchiesi and Joshua Reiss. Reverse engineering of a mix. *Journal of the Audio Engineering Society*, 58(7):563–576, 2010.
- [4] L. Barrington, A.B. Chan, and G. Lanckriet. Modeling music as a dynamic texture. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(3):602–612, march 2010.
- [5] Kenneth W. Berger. Some factors in the recognition of timbre. *The Journal of the Acoustical Society of America*, 36(10):1888–1891, 1964.
- [6] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition, October 2007.
- [7] Marina Bosi and Richard E. Goldberg. *Introduction to Digital Audio Coding and Standards*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.
- [8] Juan José Burred, Axel Röbel, and Thomas Sikora. Dynamic spectral envelope modeling for timbre analysis of musical instrument sounds. *Trans. Audio, Speech and Lang. Proc.*, 18(3):663–674, 2010.
- [9] Anne Caclin, Stephen Mcadams, Bennett K Smith, and Suzanne Winsberg. Acoustic correlates of timbre space dimensions: A confirmatory study using synthetic tones. *The Journal of the Acoustical Society of America*, 118(1):471–482, 2005.
- [10] Warren C Campbell and Jack J Heller. The contribution of the legato transient to instrument identification. In *Proceedings of the Research Symposium on the Psychology and Acoustics of Music*, pages 30–44, 1978.
- [11] Warren C. Campbell and Jack J. Heller. Convergence procedures for investigating music listening tasks. *Bulletin of the Council for Research in Music Education*, (59):pp. 18–23, 1979.
- [12] Antoni B. Chan and Nuno Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE transactions on pattern analysis and machine intelligence*, 30(5):909–926, May 2008.
- [13] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.
- [14] Melville Clark, Jr., David Luce, Robert Abrams, Howard Schlossberg, and James Rome. Preliminary experiments on the aural significance of parts of tones of orchestral instruments and on choral tones. *J. Audio Eng. Soc*, 11(1):45–54, 1963.
- [15] Alice Clifford and Joshua Reiss. Calculating time delays of multiple active sources in live sound. In *Audio Engineering Society Convention 129*, Nov 2010.
- [16] Perry R. Cook. *Music, Cognition, and Computerized Sound: An Introduction to Psychoacoustics*. The MIT Press, March 2001.
- [17] S. Dalla Bella, I. Peretz, L. Rousseau, and N. Gosselin. A developmental study of the affective value of tempo and mode in music. *Cognition*, 80(3), July 2001.

- [18] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357 – 366, aug 1980.
- [19] Gianfranco Doretto, Prabhakar Pundir, Ý Prabhakar, Stefano Soatto, Pundir Ý Ying, Ying Nian Wu, Stefano, and Soatto Ý. Dynamic textures. In *International Journal of Computer Vision*, pages 439–446, 2001.
- [20] Dan Dugan. Automatic microphone mixing. *J. Audio Eng. Soc.*, 23(6):442–449, 1975.
- [21] Charles A. Elliott. Attacks and releases as factors in instrument identification. *Journal of Research in Music Education*, 23(1):35–40, 04 1975.
- [22] S. Essid, G. Richard, and B. David. Musical instrument recognition by pairwise classification strategies. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1401–1412, 2006.
- [23] Peter Flach. *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*. Cambridge University Press, New York, NY, USA, 2012.
- [24] Harvey Fletcher and W. A. Munson. Loudness, its definition, measurement and calculation. *Bell System Technical Journal*, 12(4):377–430, 1933.
- [25] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A survey of audio-based music classification and annotation. *IEEE Transactions on Multimedia*, 13(2):303–319, 2011.
- [26] E. Glenn Schellenberg Gabriela Husain, William Thompson. Effects of musical tempo and mode on arousal, mood, and spatial abilities. *Music Perception*, 20(2):151–171, 2002.
- [27] Lise Gagnon and Isabelle Peretz. Mode and tempo relative contributions to happy-sad judgements in equitone melodies. *Cognition & Emotion*, 17(1):25–40, 2003.
- [28] Gina Gerardi and Louann Gerken. The development of affective responses to modality and melodic contour. *Music Perception*, 12(3):279–290, 1995.
- [29] John M Hajda. *The Effect of Dynamic Acoustical Features on Musical Timbre*. 2007.
- [30] Stephen Handel and Molly L Erickson. Sound Source Identification: The Possible Role of Timbre Transformations. *Music Perception: An Interdisciplinary Journal*, 21(4):pp. 587–610, 2004.
- [31] S. Hargreaves, A Klapuri, and M Sandler. Structural Segmentation of Multitrack Audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(10):2637–2647, 2012.
- [32] Perfecto Herrera-Boyer, Geoffroy Peeters, and Shlomo Dubnov. Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1):3–21, 2003.
- [33] Kate Hevner. Experimental studies of the elements of expression in music. *American Journal of Psychology*, (48):246–268, 1936.
- [34] William E. Hodgetts, Jana M. Rieger, and Ryan A. Szarko. The effects of listening environment and earphone style on preferred listening levels of normal hearing adults using an mp3 player. *Ear and Hearing*, 28(3):290–297, 2007.
- [35] International Standards Organization. *ISO226: Normal equal-loudness-level contours*, 2003.
- [36] International Telecommunication Union. Algorithms to measure audio programme loudness and true-peak audio level. *Recommendation ITU-R BS.1770-3*, 2012.

- [37] Roey Izhaki. *Mixing Audio: Concepts, Practices and Tools*. Elsevier Ltd., 1 edition, 2008.
- [38] Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. Music type classification by spectral contrast feature. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 113–116, Lusanne, Switzerland, 2002.
- [39] P. N. Juslin, J. Karlsson, E. Lindström, A. Friberg, and E. Schoonderwaldt. Play it again with feeling: Computer feedback in musical communication of emotions. *Journal of Experimental Psychology: Applied*, 12(2):79–95, 2006.
- [40] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.
- [41] Roger A. Kendall. The role of acoustic signal partitions in listener categorization of musical phrases. *Music Perception*, 4(2):185–213, 1986.
- [42] Roger A Kendall and Edward C Carterette. Perceptual Scaling of Simultaneous Wind Instrument Timbres. *Music Perception: An Interdisciplinary Journal*, 8(4):pp. 369–404, 1991.
- [43] Youngmoo E. Kim, Donald S. Williamson, and Sridhar Pilli. Towards quantifying the album-effect in artist classification. In *In Proceedings of the International Symposium on Music Information Retrieval*, 2006.
- [44] C. L. Lawson and R. J. Hanson. *Solving least squares problems*. Prentice-Hall, 3 edition, 1995.
- [45] Jin Ha Lee. Crowdsourcing music similarity judgments using mechanical turk. In *ISMIR*, Utrecht, Netherlands, 2010.
- [46] Jacob A. Maddams, Saoirse Finn, and Joshua D. Reiss. An autonomous method for multi-track dynamic range compression. In *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx-12)*, York, UK, 2012.
- [47] Michael I Mandel, Douglas Eck, and Yoshua Bengio. Learning tags that vary within a song. In *ISMIR*, Utrecht, Netherlands, 2010.
- [48] Stuart Mansbridge, Saoirse Finn, and Joshua D. Reiss. Implementation and evaluation of autonomous multi-track fader control. In *132nd AES Convention*, 2012.
- [49] Stuart Mansbridge, Saorise Finn, and Joshua D. Reiss. An autonomous system for multitrack stereo pan positioning. In *133rd AES Convention*, 2012.
- [50] Stephen Mcadams, Suzanne Winsberg, Sophie Donnadieu, Geert de Soete, and Jochen Krimphoff. Perceptual scaling of synthesized musical timbres: common dimensions, specificities, and latent subject classes. *Psychological Research*, (58):177–192, 1995.
- [51] Luca Mion and Giovanni De Poli. Score-independent audio features for description of music expression. *IEEE Transactions on Audio, Speech & Language Processing*, 16(2):458–466, 2008.
- [52] Brian C. J. Moore. *An Introduction to the Psychology of Hearing*. Academic Press, fifth edition, April 2003.
- [53] Brian C. J. Moore, Brian R. Glasberg, and Thomas Baer. A model for the prediction of thresholds, loudness, and partial loudness. *J. Audio Eng. Soc*, 45(4):224–240, 1997.
- [54] P. V. Overschee and B. D. Moor. *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Kluwer Academic Publishers, 1996.

- [55] Bryan Pardo, David Little, and Darren Gergle. Building a personalized audio equalizer interface with transfer learning and active learning. In *Proceedings of the 2nd International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies*, pages 13–18, New York, USA, 2012. ACM.
- [56] R. D. Patterson, K. Robinson, J. Holdsworth, D. Mckeown, C. Zhang, and M. Allerhand. Complex sounds and auditory images. In Y. Cazals, L. Demany, and K. Honer, editors, *Auditory Physiology and Perception*, pages 429–443. Pergamon, Pergamon, Oxford, 1992.
- [57] J. Paulus. Acoustic modelling of drum sounds with hidden markov models for music transcription. In *International Conference on Acoustics, Speech and Signal Processing*, volume 5. IEEE, 2006.
- [58] E. Perez-Gonzalez and J. D. Reiss. Determination and correction of individual channel time offsets for signals involved in an audio mixture. In *125th AES Convention*, 10 2008.
- [59] E. Perez-Gonzalez and J. D. Reiss. Automatic mixing. In Udo Zölzer, editor, *DAFX: Digital Audio Effects*, pages 523–549. John Wiley & Sons, Ltd, second edition, 2011.
- [60] Enrique Perez-Gonzalez. *Advanced Automatic Mixing Tools for Music*. PhD thesis, 2010.
- [61] Enrique Perez-Gonzalez and Joshua D. Reiss. Automatic equalization of multichannel audio using cross-adaptive methods. In *127th AES Convention*, 2009.
- [62] Enrique Perez-Gonzalez and Joshua D. Reiss. Automatic gain and fader control for live mixing. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2009.
- [63] Enrique Perez-Gonzalez and Joshua D. Reiss. A real-time semiautonomous audio panning system for music mixing. *EURASIP Journal on Advances in Signal Processing*, 2010.
- [64] Pedro Pestana. *Automatic Mixing Systems Using Adaptive Digital Audio Effects*. PhD thesis, Universidade Católica Portuguesa, 2013.
- [65] Pedro Pestana and Joshua Reiss. Intelligent audio production strategies informed by best practices. In *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*, Jan 2014.
- [66] Pedro Duarte Pestana, Zheng Ma, Joshua D. Reiss, Alvaro Barbosa, and Dawn A. A. Black. Spectral characteristics of popular commercial recordings 1950-2010. In *Audio Engineering Society Convention 135*, Oct 2013.
- [67] Thomas Quatieri. *Discrete-time speech signal processing: principles and practice*. Prentice Hall Press, Upper Saddle River, NJ, USA, 2001.
- [68] Thomas Quatieri. *Discrete-time speech signal processing: principles and practice*. Prentice Hall Press, Upper Saddle River, NJ, USA, 2001.
- [69] Zafar Rafii and Bryan Pardo. Learning to control a reverberator using subjective perceptual descriptors. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pages 285–290, Kobe, Japan, October 26-30 2009.
- [70] Don Michael. Randel. *The Harvard dictionary of music / edited by Don Michael Randel*. Belknap Press of Harvard University Press, Cambridge, MA :, 4th ed. edition, 2003.
- [71] Dale Reed. A perceptual assistant to do sound equalization. In *Proceedings of the 5th international conference on Intelligent user interfaces*, IUI '00, pages 212–218, New York, NY, USA, 2000. ACM.

- [72] Joshua D. Reiss. Intelligent systems for mixing multichannel audio. In *Proceedings of the 17th International Conference on Digital Signal Processing (DSP)*, pages 1–6, July 2011.
- [73] M. G. Rigg. Speed as a determiner of musical mood. *Journal of Experimental Psychology*, 27:566–571, 1940.
- [74] Andrew T. Sabin and Bryan Pardo. 2DEQ: An intuitive audio equalizer. In *Proceedings of the seventh ACM conference on Creativity and cognition*, pages 435–436, New York, NY, USA, 2009. ACM.
- [75] Andrew T. Sabin and Bryan Pardo. A method for rapid personalization of audio equalization parameters. *Proceedings of ACM Multimedia*, pages 769–772, 2009.
- [76] Mihir Sarkar, Cyril Lan, and Joe Diaz. Words that Describe Timbre: A Study of Auditory Perception Through Language. In *Language and Music as Cognitive Systems Conference*, 2007.
- [77] E. M. Schmidt, M. Prockup, J. Scott, B. Dolhansky, B. Morton, and Y. E. Kim. Relating perceptual and feature space invariances in music emotion recognition. In *Proceedings of the International Symposium on Computer Music Modeling and Retrieval (CMMR)*, London, UK, June 2012.
- [78] E.M. Schmidt, R.V. Migneco, J.J. Scott, and Y.E. Kim. Modeling musical instrument tones as dynamic textures. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on*, pages 329–332, Oct 2011.
- [79] ErikM. Schmidt, Matthew Prockup, Jeffrey Scott, Brian Dolhansky, BrandonG. Morton, and YoungmooE. Kim. Analyzing the perceptual salience of audio features for musical emotion recognition. In Mitsuko Aramaki, Mathieu Barthet, Richard Kronland-Martinet, and Sølvi Ystad, editors, *From Sounds to Music and Emotions*, volume 7900 of *Lecture Notes in Computer Science*, pages 278–300. Springer Berlin Heidelberg, 2013.
- [80] S. Scholler and H. Purwins. Sparse approximations for drum sound classification. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):933–940, 2011.
- [81] Jeffrey Scott and Youngmoo E. Kim. Analysis of acoustic features for automated multi-track mixing. In *International Society for Music Information Retrieval Conference*, Miami, Florida, 2011.
- [82] Jeffrey Scott, Matthew Prockup, Erik M. Schmidt, and Youngmoo E. Kim. Automatic multi-track mixing using linear dynamical systems. In *Proceedings of the 8th Sound and Music Computing Conference*, Padova, Italy, 2011.
- [83] Mike Senior. *Mixing Secrets for the Small Studio*. Focal Press, 1 edition, 2011.
- [84] Sajid Siddiqi, Byron Boots, and Geoffrey Gordon. A constraint generation approach to learning stable linear dynamical systems. In *Advances in Neural Information Processing Systems 20*, pages 1329–1336. MIT Press, Cambridge, MA, 2008.
- [85] Umut Simsekli, Antti Jylhä, Cumhuri Erkut, and Ali-Taylan Cemgil. Real-time recognition of percussive sounds by a model-based method. *EURASIP J. Adv. Sig. Proc.*, 2011.
- [86] P Smaragdis, B Raj, and M Shashanka. Sparse and shift-invariant feature extraction from non-negative data. *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 2069–2072, 2008.
- [87] Paris Smaragdis and Bhiksha Raj. Shift-invariant probabilistic latent component analysis. *Journal of Machine Learning Research*, 2007.

- [88] J. A. Speck, E. M. Schmidt, B. G. Morton, and Y. E. Kim. A comparative study of collaborative vs. traditional annotation methods. In *ISMIR*, Miami, Florida, 2011.
- [89] H Terasawa, M Slaney, and J Berger. A statistical model of timbre perception. In *ISCA Tutorial and Research Workshop on Statistical And Perceptual Audition (SAPA2006)*, pages 18–23, Pittsburgh, 2006.
- [90] Michael J; Reiss Terrell. Automatic monitor mixing for live musical performance. *J. Audio Eng. Soc*, 57(11):927–936, 2009.
- [91] R. E. Thayer. *The Biopsychology of Mood and Arousal*. Oxford Univ. Press, Oxford, U.K., 1989.
- [92] A. Tindale, A. Kapur, G. Tzanetakis, and I. Fujinaga. Retrieval of percussion gestures using timbre classification techniques. In *Proceedings of the 5th International Society for Music Information Retrieval Conference*, 2004.
- [93] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293 – 302, jul 2002.
- [94] S. Umesh, L. Cohen, and D. Nelson. Fitting the mel scale. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 217–220, 1999.
- [95] Dominic Ward, Joshua D. Reiss, and Cham Athwal. Multitrack mixing using a model of loudness and partial loudness. In *133rd AES Convention*, October 2012.
- [96] Gregory D. Webster and Catherine G. Weir. Emotional responses to music: Interactive effects of mode, texture, and tempo. *Motivation and Emotion*, 29:19–39, 2005.
- [97] David Wessel. Timbre Space as a Musical Control Structure. *Computer Music Journal*, 3(2):45–52, 1979.
- [98] Asterios Zacharakis, Konstantinos Pasiadis, Joshua D Reiss, and George Papadelis. Analysis of Musical Timbre Semantics through Metric and Non-Metric Data Reduction Techniques. In *Proceedings of the 12th International Conference on Music Perception and Cognition*, 2012.