Evaluation of Interpolation Strategies for the Morphing of Musical Sound Objects

Federico O'Reilly Regueiro

Music Technology Area
Department of Music Research McGill University
Montreal, Canada

September 2010

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Arts in Music Technology.

2010/09/17

Abstract

Audio morphing is a timbre-transformation technique that produces timbres which lie in between those of two or more given tones. It can thus be seen as the interpolation of timbre descriptors or features. Morphing is most convincing when the features are perceptually relevant and the interpolation is perceived to be smooth and linear. Our research aims at producing practical guidelines for morphing musical sound objects. We define a set of features aimed at representing timbre in a quantifiable fashion, as completely and with as little redundancies as possible. We then report the interpolation of each single feature imposed on an otherwise neutral synthetic sound, exploring strategies to obtain smooth-sounding interpolations. Chosen strategies are then evaluated by morphing recorded acoustic instrumental sounds. All of the scripts and the resulting sounds are available through the *www* to the reader.

Sommaire

Le morphing audio est une transformation sonore produisant des timbres intermédiaires entre ceux de sons donnés. On peut considérer qu'il s'agit d'une interpolation des descripteurs du timbre. Le morphing est plus convaincant lorsque les descripteurs choisis sont pertinents perceptivement et quand l'interpolation est perçue comme étant linéaire. Le but de nos recherches est de constituer un guide pratique pour le morphing des objets musicaux. Nous définissons une collection de descripteurs qui décrivent le timbre d'une façon complète et non redondante. Nous nous livrons ensuite à une étude systématique ayant pour objectif de déterminer les meilleures stratégies d'interpolation, pour chaque descripteur sur des sons synthétiques simples. Les stratégies adaptées au traitement des signaux synthétiques sont ensuite évaluées pour la modification de sons d'instruments acoustiques. Toutes les routines et les fichiers audio sont disponibles sur un site internet.

Acknowledgments

I would like to acknowledge Mexico's National Funding for Arts and Culture (FONCA) for the generous support which they have granted me during my studies. By the same token, I'm grateful to Mexico's Council for Science and Technology (CONACyT) for they have also funded this endeavor. Merci a lot, Philippe for guiding me through this wonderful adventure. Thanks to fellow students at the SPCL for many fruitful exchanges. My gratitude goes out to Steve for ploughing through parts of this text, enhancing the clarity while on the lookout for *ferner's gaffes*. To my parents for many years of unwaivering support along the winding path that my studies have taken me. A Karen, merci pour être le soleil quand il fait noir et pour ta patience infinie.

Contents

List of Figures

List of Acronyms

SMS	Spectral Modeling Synthesis
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
STFT	Short-Term Fourier Transform
ADSR	Attack Decay Sustain and Release envelope
SE	Spectral Envelope
FM	Frequency Modulation
AM	Amplitude Modulation
SEM	Spectral Envelope Modulation
OSC	Onset Spectral Centroid
PAT	Partial Attack Times
PRT	Partial Release Times
EOR	Even to Odd Energy Ratio
MDS	Multi-Dimensional Scaling
LPC	Linear Prediction Coefficients
SPWD	Smoothed Pseudo Wigner Distribution
FOF	Fonctions d'Ondes Formantiques
OLA	Overlap Add
LSP	Line Spectral Pairs
DTW	Dynamic Time Warping
DFW	Dynamic Frequency Warping
SDIF	Sound Description Interchange Format
SPCL	Sound Processing and Control Laboratory

Chapter 1

Introduction

Transformations are central to music: inflections, modulations, rhythmic transformations, motivic variations, all serve to shape musical discourse and elicit a reaction in the listener.

Although timbral transformations are already a part of traditional composition through the use of techniques such as *Klangfarbenmelodie*, digital signal processing reveals an enormous potential for the exploration of this avenue by allowing the creation of hybrid sounds through sound morphing.Developments in audio morphing are readily applicable to electro-acoustic music, where we find examples such as Jean-Baptiste Barrière's *Chreode I* or Trevor Wishart's *Vox-5*.

The following thesis evaluates strategies for achieving feature-based morphing of musical sound objects via a stochastic-plus-deterministic additive model. This evaluation aims to contribute to the establishment of a series of practical guidelines for musicians who wish to delve into this fascinating terrain.

1.1 Feature-based morphing of musical sound-objects explained

As analog studio techniques gained in sophistication, artists began exploring the possibility of transforming timbre in ways that would have been very difficult or even impossible to achieve with purely acoustical means. An example of smooth timbral transitioning during this analog era was *Red Bird*[1] by Trevor Wishart; a musical piece rife with morphing that was achieved exclusively through analog treatments.

The advent of digital recording and sound processing techniques created the possibility of achieving timbral transformations with a much greater degree of control. The idea of

smoothly transitioning from one timbre to another then became more accessible. Research and commercial applications that drew on this technology began to make their appearances as early on as the late 1970's.

The term audio morphing refers precisely to the migration from one timbre toward another. Thus, morphing requires at least two input sounds to generate a new one; a hybrid is created which merges timbral characteristics from the original inputs. Given the temporal nature of sound, audio morphing can be used to generate either a single sound which migrates from one timbre to another or a discrete series of sounds between two known sounds[2].

Although it's simple to perform a cross-fading between two sounds, the most likely outcome is that intermediate sounds would not be perceived as a hybrid, but as two sources mixed together. If we seek to produce timbres that lie in between two known timbres, it seems natural to think that we need to find a set of timbre descriptors that allow hybridization by means of their interpolation. Descriptors or features can be temporal cues such as attack or release; or atemporal characteristics that describe the sound and its timbre. Some examples of the latter are spectral shape, fundamental frequency or f_0, even to odd partial amplitude ratio, inharmonicity and spectral flux.

Many documented audio morphing procedures [3, 4, 2, 5, 6, 7, 8] use some form of the additive stochastic plus deterministic model, which enables the establishment of a correspondence between partials and the direct interpolation of their frequencies and amplitudes. Such a direct interpolation is effective, as it is based on an acoustically-motivated abstraction of data. However, direct interpolation may still yield morphs which are perceived as mixed units and not true hybrids. In response to this shortcoming, we find some works in the literature which take steps toward descriptor-based morphing[4, 7], the current work seeks to build upon these efforts.

As with most studies of audio morphing, we will use a stochastic-plus-deterministic additive model. We have chosen this model both for the flexibility that it affords the user and the potential it offers for extracting timbral descriptors by analyzing the additive representation of a signal. The model divides a signal into two components: a deterministic one which is best modelled by a relatively small set of partials and a stochastic one, which is best viewed as filtered noise. Figure 1.1 represents this component separation performed on the sound of a clarinet.

The scope of the present work solely encompasses morphing in the context of musi-

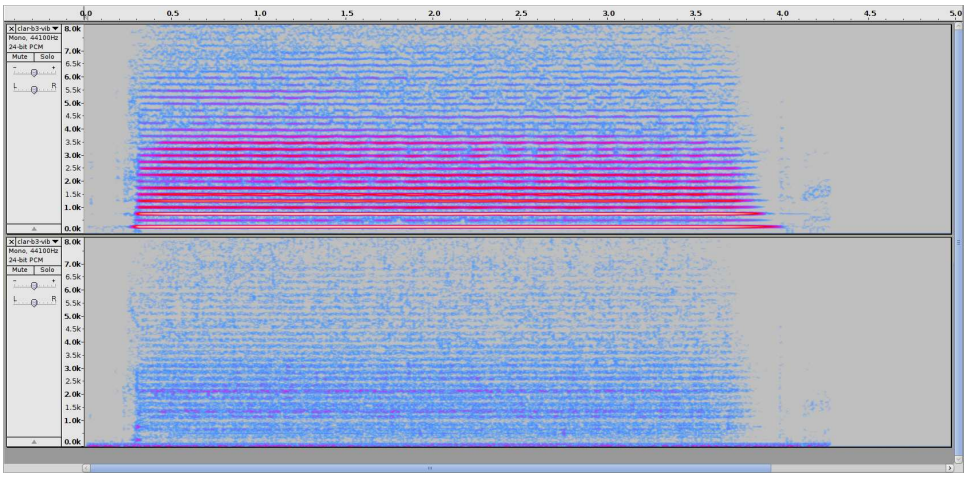


Fig. 1.1 Deterministic and stochastic component separation for a clarinet note. On the top we can see the $STFT$ of the deterministic component, with a well defined series of partials. On the bottom, we see the remaining part of the signal, which can be best modeled as coloured noise.

cal sound objects or single-event musical sounds, i.e. sounds which are relatively stable throughout their lifespan, such as those commonly found in music. In other words, the set of sounds for which we wish to evaluate strategies is that of sounds which constitute a single event flanked by transients or silence. The clarinet note from figure 1.1 is a good characterization of this type of sound.

1.2 Aims of the current study

Research on audio morphing has gradually spawned a body of literature. Additionally, a large set of articles have been written on image morphing, speech morphing and speech transformation. The principles and techniques outlined in these articles can be easily transferred to the context of audio morphing. One of the goals of the present work is to compile pertinent ideas from prior publications.

Since we intend on exploring feature-based audio morphing, another one of our goals is to choose a set of descriptors that fulfill certain conditions which we hereby detail. They should be compatible with the chosen representational model, ie stochastic-plus-deterministic additive model. The set of descriptors should also be tailored to fit timbral characteristics which are frequent within musical sound objects as they have been defined.

Furthermore, in order to effect convincing morphs, the chosen features should be meaningful, i.e. correlated to perceptually relevant characteristics of timbre. Lastly, the choice of the set of descriptors should ideally be such that, it allows the interpolation of each feature separately without modifying other features.

Another one of the goals of our research stems from the last condition: we will set out to explore isolated feature interpolation on a series of synthetic sounds, building up toward the morphing of real-world sounds. Along the way, we will produce audio files to exemplify our findings and the final morphs of real-world sounds will serve as a proof-of-concept for results found during the process of single-feature interpolations.

1.3 Premises of the current work

There are three main ideas fueling this research. First, we argue that in order to achieve a convincing morph, it is essential to define a set of meaningful features–i.e. correlated to perceptually meaningful characteristics. Second, we argue that the interpolation of isolated features will more readily reveal potential problems and their corresponding solutions than the concurrent interpolation of all features. Third, we are convinced that morphing implementation examples hold significant pedagogical value for composers and sound artists who wish to implement their own morphing routines.

1.4 Structuring of the current work

The work is organized into six chapters and one appendix. After the brief introduction presented in the current chapter, the reader shall find a literary review on chapter number 2. The review contains an overview of publications related to morphing in the context of music, speech processing and image processing; we present some compositions with noteworthy examples of morphing and mention several related software implementation efforts. Chapter 3, formally establishes the framework in which the current research is circumscribed. It presents some important terminology and the set of features to be used during interpolation. Chapter 4 reports on the realization of each single-feature interpolation and the final real-world sound interpolations; detailing the different strategies used to counter problems that became evident during implementation. Subsequently, chapter 5 is based on the application of the defined feature-set and interpolation strategies, serving somewhat as

an implementation guide. Audio files to accompany the realizations of chapters 4 and 5 can be found on the project's home page[9]. Conclusions drawn from the results are presented in chapter 6. Lastly, the appendix presents a note in regards to the implementation scripts, which can also be found on the project's home page[9].

Chapter 2

State of the Art

Besides being of interest for the audio processing community, for a long time now, morphing has received significant attention in the image processing community and to a certain degree within the speech processing and audio community. In this chapter we present a broad overview of some works on the subject. The overview is broken up by field of study: audio morphing with musical applications is presented first, findings from speech morphing are presented later and relevant image morphing ideas, which enrich or enforce the topic, are presented at the end of the chapter.

2.1 Survey of morphing in the musical context

Advances in timbral interpolation are found in varied source materials, some are found in the form of articles reporting on realizations, others make their appearance in the form of music which is accompanied by a written presentation of the underlying principles and some are found mainly as software packages which can be accompanied by relevant documentation. The presentation in the section is broadly separated by these three somewhat arbitrary categories. Papers are not reviewed extensively but an attempt has been made to retain key ideas from each of them.

2.1.1 Literary review

Perceptual effects of spectral modifications on musical timbres

After Grey[10] published a study of timbre spaces through Multi-Dimensional Scaling in 1977, Grey and Gordon[11] carried out what are generally referred to as the first sound morphings present in the literature. Grey and Gordon re-synthesized 16 sounds of different instruments with an equalized loudness and pitch. The representation from which the sounds were re-synthesized was based on a simplified additive model. The experiment was performed in order to confirm their hypothesis in regards to the role of the spectral envelope in the perception of timbre. For this purpose, from each one of the four pairs of sounds, they exchanged the peak amplitudes of their corresponding harmonics. In doing so, they effectively exchanged the sounds' spectral envelopes while retaining all other characteristics for each sound.

Dynamic timbre control for real-time digital synthesis

Schindler[12] presents a data reduction strategy which was based on a hierarchical-tree representation of envelopes which could be used for either amplitude or spectral envelopes. He also described a two dimensional state transition scheme. The aim of the data reduction technique and of the state change scheme was to facilitate real-time control for the additive synthesis of instruments with a dynamic timbre control[1]. By seeking to produce intermediate timbres from discrete points sampled from a single instrument's timbre space, Schindler was also performing morphing.

Adventures in musique concrete at CARL

In a paper published in the proceedings of the 1985 ICMC, Mark Dolson[13] reports the usage of a technique known as cross-synthesis. Similarly to a vocoder the technique imprints one sound's spectrum on another sound, the later preferably being a spectrally rich one, in order to heighten the effect. Let's refer to *sound 1* as the source for the spectrum to be imprinted on *sound 2*. Cross synthesis is achieved by performing an $STFT$ analysis on *sound 1* to extract it's envelope by smoothing the $STFT$ along the frequency axis.

[1] In Schindler's article, namely the ratio between the partial's frequencies and amplitudes.

Subsequently, $STFT$ analysis-synthesis is performed on *sound 2* with an intermediate multiplication of its spectra by the envelopes extracted from *sound 1*.

Sound hybridization techniques based on a deterministic plus stochastic decomposition model

Serra[3] proposes effecting hybridization through a stochastic plus deterministic additive model and contrasts it to cross synthesis or, as he calls it, hybridization through the $STFT$. In Serra's presentation of morphing through an additive model, there is an implicit hierarchy of the two original sounds, where sound 1 retains its temporal and pitch characteristics and sound 2 is warped to match it. Serra presents a scheme that grants independent interpolation of f_0, partials' ratio to f_0, overall deterministic amplitude, partial's amplitude ratios, stochastic part amplitude, stochastic interpolation factor and time warping factor for sound 2. Morphing through SMS (Serra's implementation of an additive model) and cross synthesis are presented as musically complementary tools given the vast differences between the effects that they achieve. In this regard, cross-synthesis is adequate when the spectrum of one of the sounds to be hybridized has well defined formants, such as is the case of speech, and the second has a rich and relatively flat spectrum, such as the roaring of the sea. Sound hybridization by additive model, on the other hand, is likely to offer better results when the deterministic components of both sounds can be modeled with a similar amount of partials, irrespective of their spectral contour.

Timbre morphing of sounds with unequal numbers of features

Tellman, Haken and Holloway[4] from the CERL sound group at the University of Illinois, addressed the topic of morphing in an article published by the Journal of the Audio Engineering Society in 1995. In their article, they present a generalized strategy for morphing which utilizes the Lemur representation; an implementation of McAulay-Quatieri's sinusoidal representation[14]. *A grosso modo*, their approach to morphing implies the description of a given sound, which occupies a timbre space, by a series of features. A feature is broadly defined to be a temporal portion of the sound that is important to the morphing process. Features are taken to be either unique or repeatable.

Unique features are those which occur once, and only once, for the duration of each one of the input sounds and therefore have a one to one correspondence between sounds.

Clear examples of unique features are the onset or the end of a note. On the other hand, repeatable features are those features which can be omitted or used in the morphing as is seen fit. An example of a repeatable feature is a vibrato amplitude peak. The definition of feature used within the article raises an interesting point. It's important to note that in the current text and other prior writings[7] the expression *feature* is used as a synonym of the word *descriptor*, as opposed to the usage given to it in Tellman, Haken and Holloway's publication.

Morphing is thus seen as a process which includes time-warping, partial matching and repeatable feature stepping. Time warping is used to align unique temporal features such as maxima or minima within the amplitude envelope. Amplitude weighted partials are matched between inputs. They offer more than one solution to partial coupling issues: firstly, partials for which frequencies do not have a closely corresponding partial frequency in the complementary input are paired with a zero-amplitude partial at the same frequency; on the other hand, given that the analysis can yield erratic frequencies for low amplitude partials, those with an amplitude lesser than a given threshold are corrected to the closest frequency which is a multiple of the fundamental frequency. In regards to the interpolation of repeatable features found between unique features, they are matched, warped and omitted if necessary so that the end product contains an interpolated number of repeatable features between each set of unique features. In this article, Tellman Haken and Holloway also propose the idea of interpolating the log_2 frequencies rather than a linear interpolation:

$$f_{(1+\alpha,k)} = 2^{(1-\alpha)log_2(f_{(1,k)})+\alpha log_2(f_{(2,k)})} \tag{2.1}$$

where α is the interpolation factor, and $f_{(1,k)}$, $f_{(2,k)}$ and $f_{(1+\alpha,k)}$ are respectively the k^{th} partials' frequencies for input 1, input 2 and interpolated output. This can be shown to be equivalent to

$$f_{1+\alpha} = f_1 \left(\frac{f_2}{f_1}\right)^{\alpha}$$

which is another form of writing formula 3.5 which will be presented in chapter 3[2].

[2]This, of course is valid as long as neither interpoland is 0.

Automatic audio morphing

Slaney, Covell and Lassiter[2] present a method for automatically morphing audio. Interpolated sounds are found by extracting smooth spectral envelope information from the targets; warping along the time and frequency axes to align their transients, attacks and pitch, and then interpolating matched partial amplitudes and spectral envelopes. One of the ideas presented in the paper is the possibility of three different types of morphing: a static or stationary morph, finding one given sound between two targets with a constant interpolation factor; a dynamic morph, producing a sound for which the interpolation factor changes in the course of its duration; and a cyclo-stationary morph, which consists of finding a series of repeated sounds, each with a constant interpolation factor, which smoothly evolve from one of the target sounds to the other.

Timbre morphing of synthesised transients using the Wigner time-frequency distribution

In a paper by Lysaght and Vernon[15] the authors ponder the interpolation of short-duration transient sounds. Because of the duration of such sounds, it is difficult to perform a Fourier-based analysis with sufficient resolution for morphing. Thus, the authors propose using a Smoothed Pseudo Wigner Distribution for the analysis stage. Since the SPWD is a concentrated time-frequency representation, it allows for an increased resolution over that afforded by the Fourier derived techniques generally used for morphing. Additionally, they propose using subgraph isomorphism as a pattern matching technique in order to find feature-correspondence for the two known sounds. The latter idea is more fully explained in a subsequent publication by Lysaght, Vernon and Timoney[16].

Morphing for karaoke applications

While some of the articles found on morphing are extensible to real-time performance, most applications are best suited for offline use. This is, of course, not possible when we think of a karaoke application, the very nature of which requires a real-time algorithm. In a paper presented at the ICMC in 2000 by the audiovisual group from Pompeu Fabra University[5], interpolation is performed between a pre-analysed recording of a song and the user's performance of the same song with user-specified interpolation coefficients for different features.

The underlying model for the data is that of SMS[17]. For the purpose of the application, SMS analysis was fine-tuned to perform with low latency for the singing voice. In addition to the low-level SMS model, Cano, Loscos et al propose using some higher-level attributes as per a prior article by Serra and Bonada[18]. The attributes or features that are interpolated are spectral shape, fundamental frequency, amplitude, residual signal, pitch micro variations, vibrato, spectral tilt and harmonicity.

Only relatively steady-state sections of the input are morphed; transients are left untouched. Because of this constraint and because of the fact that corresponding vowel sounds need to be matched for the morphing, the audio is separated into what the authors call morphing units, i.e. a relatively steady state signal flanked by silence and/or transients.

An interesting finding reported by the authors is that for the interpolation of both spectral envelopes, which is performed on a bin-by-bin basis, or by cross-fading spectral envelopes, interpolation factors close to 0.5 yield a relatively flat spectrum. This corresponds to the fact that envelopes with non-correspondent peaks will tend to average out.

Sound timbre interpolation based on physical modeling

Hikichi and Osaka[19] propose morphing by means of a physical model as an alternative method to morphing through an additive model. In order to do this, it is necessary to create a unified model of the two known sounds. In their paper, they present a three-part physical model consisting of an exciter, a vibrator and a resonator. The unified model is useful for synthesizing both guitar and piano. It is through the interpolation of parameters in such a model that timbre morphing is achieved. Albeit the difficulties in fitting a model to the known sounds, the primary advantage of such a method is the substantial reduction of parameters that need to be interpolated. Hikichi and Osaka also state three main areas of exploration to follow: the evaluation of interpolation strategies different from linear interpolation, which is the strategy that they used; the expansion of the model to include other target timbres, and the qualitative comparison between the results obtained by the use of their method and the results obtained through the interpolation of an additive models' parameters.

Sound morphing using Loris and the reassigned bandwidth-enhanced additive sound model: practice and applications

Because morphing is essentially data interpolation, it's greatly impacted by the underlying choice of a representational model. While many articles on the subject converge toward a sinusoidal representation, there are differences even among the additive models. Haken, Fitz, Lefvert and O'Donnell[6] proposed morphing with the Loris representation which stands apart from other sinusoidal models in two respects: that it consolidates stochastic and deterministic parts by assigning a bandwidth to each partial and that it uses a time reassignment method for resolution improvement. The argument put forth in favor of enhanced-bandwidth partials is the resulting compactness of data and the convenience of dealing with a single data stream for interpolation, as opposed to two data sets in representations such as SMS.

In their method, data pairing for partials is governed by a principle which they call *channelizing* that consists in establishing corresponding frequency regions in the inputs and allowing for a single bandwidth-enhanced partial in each region. The conflicting coexistence of partials within regions can be solved either by eliminating additional partials (*sifting*) or by broadening the remaining partial's bandwidth according to the amount of energy in the removed partial (*energy redistribution*). Each partial is seen to be a set of three envelope streams, one for frequency, one for amplitude and one for bandwidth; interpolation is then performed on each one of the three envelope pairs for every set of corresponding partials.

The same ideas were discussed with much more detail by Lippold Haken, Kelly Fitz and Paul Christensen in the third chapter, *Beyond Traditional Sampling Synthesis: Real-Time Timbre Morphing Using Additive Synthesis*, of Beauchamp's book *Sound of Music: Analysis, Synthesis, and Perception*[20].

High-level audio morphing strategies

Hatch's thesis[7], as can be expected, contains a large amount of information on the subject of morphing. It would therefore be inappropriate to cover all of the aspects presented in his thesis but we will limit our presentation to some of the most relevant concepts therein. The underlying representational model used for morphing is LORIS and the morphing strategies that are explored are meant to satisfy a wide range of inputs, not a particular set of sounds. The features that are defined to be interpolated are pitch, spectral envelope, harmonicity

and spectral centroid. Hatch uses linear time warping for amplitude envelope matching.

F_0 is interpolated on a logarithmic scale and a modulo 2 is applied to the frequency multiplier, such that glissandi of more than an octave are avoided. The spectral centroid is interpolated logarithmically and imposed onto the morphed sound in one of two ways: either by multiplying the frequencies of partials when a sound does not have a harmonic partial structure, or by modifying the partial's amplitudes when there is a harmonic structure. Because the strategies tested throughout the work are intended for sounds that have no constraints in regards to having a relatively static amplitude or static frequency content[3], Hatch proposes two different strategies for partial matching through time: one which is based on a sliding window, where partials are matched again at the end of each window length, and another in which partial matching is locked throughout the duration of the sound. The later strategy is more geared toward sounds with relatively stationary states throughout their duration.

SSynth: a real time additive synthesizer with flexible control

In this article, Verfaille, Boissinot, Depalle and Wanderley[21] present a synthesis framework geared for real-time additive synthesis based off of a three-dimensional database of sounds. The database dimensions are pitch, dynamic level and instrument. Additionally, a series of spectral envelopes per instrument are also part of the database and the selection or interpolation of these envelopes is dependent on parameters which are independent from pitch and dynamic level. The production of a note is thus an interpolation between the closest four points, two closest in pitch and two closest in dynamic, for the particular instrument. In the paper, there is no explicit mention of morphing between instruments but it is an obvious application which would imply interpolating between the closest eight points in the database–four for each instrument.

Because Ssynth is intended for real-time synthesis, morphing is performed in a different way for transients than that of locally stationary parts of the sound. Stable-state parts of the sound are looped through their additive parameters for as long as the note is being played; requiring pitch warping for alignment, partial frequency matching and amplitude interpolation as well as spectral envelope interpolation. Transients on the other hand, involve an additional time warping as well as involving more stringent precautions to avoid

[3]In other words, it is geared toward the morphing of *any* two snippets of audio.

gliding artifacts due to the fact that partials' relations are in flux during the transient stage.

In the article, particular attention is given to the fact that the spectral envelope $E(f)$ can be obtained and represented in many different ways. Because each of these representations might be better suited to a given context, the authors have included a table of implemented exact and inexact conversions between different representations of $E(f)$.

Evolutionary spectral envelope morphing by spectral shape descriptors

Caetano and Rodet[22] propose finding intermediate spectral envelopes by means of the interpolation of the envelopes' statistical descriptors: centroid, spread, kurtosis, skewness and slope. In order to properly achieve this interpolation, they propose the usage of an evolutionary algorithm applied to the trajectories of the poles and zeros between the two spectra. The results compare very favorably to the naive point-by-point interpolation and they represent a slight improvement over the interpolation of the LPC coefficients. Although the method is somewhat cumbersome, it has the advantage of offering control over each independent statistical descriptor. The results of this endeavor may be consulted on Marcelo Caetano's home page[23].

Spectral tools for dynamic tonality and audio morphing

In an article published in the Computer Music Journal, Sethares, Milne, Tiedje, Prechtl and Plamondon[8] present the Spectral Toolbox; a series of Java classes and Max/MSP routines created for dynamic tonality and audio morphing. Morphing is proposed as a manipulation which is related to dynamic changes in tuning and temperament, where the series of partials in a given sound is made to match the present tuning and temperament.

In the article, morphing is presented in a very similar fashion than that of several prior publications; it is achieved through partial matching and interpolation performed on the deterministic part of an additive model [4]. In terms of the morphing aspect of it, it is noteworthy that the paper explores strategies for matching partials which are not necessarily harmonic. They propose to achieve this through the choice of one out of three differing criteria: matching for nearest frequency, matching for corresponding component number[5] or matching according to the order of each partial series' amplitudes. They also

[4] Of course, if morphing is viewed as a partial alignment technique, the stochastic part of the model should remain unmodified.

[5] In either ascending or descending order.

state that frequency interpolation should be done on a logarithmic scale but that amplitude interpolation should be done in a linear fashion. Both the toolbox and results obtained through its use can be consulted at the dynamic tonality website[24].

Presently submitted for review

At the time of writing this thesis, we have received news about the submission an article on the topic on descriptor-based morphing by Marcelo Caetano and Xavier Rodet from IRCAM's analysis/synthesis team. The paper was submitted to the International Computer Music Conference for consideration for the 2010 conference. Since the paper would be published shortly after this thesis, the reader is directed to consult the ICMC 2010 proceedings for further information.

2.1.2 Software implementations review

From the previous list of publications, there are two publicly available software implementations created with the purpose of morphing audio: LORIS[25], and the Spectral Toolbox[24]. There is also mention of two software packages that are not currently available. One of them is Ssynth, also mentioned in the article review section, which is still under a stage of development and the other one is Oberheim's G-WIZ, which is mentioned in Tellman, Haken and Holloway's article[4] and of which there is no further appearance in the literature.

Additionally there are some cases of audio morphing in commercial applications. Early examples include embedded timbral transformation algorithms from the Fairlight CMI, the PPG audio synthesizer or more recent digital musical instruments such as the Emu Emulator 3 hybrid sampler. On such instruments, both wave-form and spectral shape interpolation can be performed.

More recent software-only implementations also exist. Camel Audio's Cameleon5000[26] performs audio morphing according to a user-specified trajectory on a *morph square* that contains a target sound on each one of its corners. The Cameleon uses an additive model to represent audio snippets which are to be morphed.

Ircam's Diphone Studio[27] decomposes input sounds into chunks or diphones and generates a dictionary with them. Once the dictionary has been assembled, the user can specify a sequence of chunks and the software morphs from each diphone to the next.

The Composer's Desktop Project[28], or CDP for short, is a collection of routines for the treatment of audio which has been spawned by a large international community for over 20 years. Within this large collection there are a number of routines intended for morphing.

2.1.3 Music applications review

Red Bird, Vox-5

Trevor Wishart, one of the founders and main contributors of the CDP, has dealt extensively with sound transformations and sound morphing, which he has documented[29]. In one of Wishart's early works[1], composed during 1973 to 1977, Wishart explores morphing by means of analog techniques. The amount of transformations obtained throughout the piece is astonishing; utterances and cries morph to birdsong, barks, gunshots, slamming doors and all sorts of animal and mechanical sounds. Some of the transformations present in the piece are documented on the CD liner notes[30].

Years later Wishart composed a piece while in residence at IRCAM[31]. Vox-5 was created in 1986 with extensive use of the phase vocoder. The piece revolves around the transformation of utterances to other real-world sounds, alluding to the voice of Shiva[32]. The techniques used in the creation of the piece include formant preservation during spectral manipulation; warping for spectral matching; spectral stretching and shifting; as well as spectral interpolation. Wishart wrote an article in the Computer Music Journal[33] which documents the compositional process as well as the techniques used therein.

Chreode I

Jean Baptiste Barrière[31, 32, 34] composed Chreode I on IRCAM's PDP-10 during 1983. Barrière used FOF synthesis to generate seamless transitions between different timbres that represent four general characters; vocal, instrumental, acoustic and synthesized.

Farinelli

Corbiau's biographical film on Carlo Maria Broschi's life as a virtuosic castrato singer was one of the first realizations of musical morphing presented to wider audiences[35, 36].

Generally speaking, the voice of Castrati covered a more ample register than that of any other, ehem... un-modified tessitura. Given the parts written for Farinelli, his register can be inferred to have encompassed both the ranges of a Soprano and of a Tenor. Since no singer alive these days is known to possess the combination of such a wide register and extensive musical training, it was necessary to craft the music from parts performed by two singers of the aforementioned tessiturae. The problem of merely mixing these two parts was that the timbral characteristics of each voice were quite distinct and would have yielded a composite register with a very clear timbral discontinuity somewhere in the middle. Morphing and interpolation were key elements for the riddance of such a discontinuity.

Generally speaking, at each intersecting pitch and for each voiced sound, the two timbral extremes for each of the two voices were established and timbres were interpolated at different rates along the intersecting range, the interpolation being weighted by their proximity to each of the two tessitura's midpoints.

Sheep

One notable realization of morphing in the realm of Progressive Rock can be found in Pink Floyd's song Sheep, from the album Animals[37]. At the end of each verse, the voice smoothly morphs into an synthesizer tone. Unfortunately, we have not come across any sources describing the process by means of which this was achieved.

2.2 Survey of morphing in the context of speech processing

Within speech synthesis, there are some applications of morphing and interpolation which are used areas such as generation of emotions in synthetic speech, unit articulation in concatenative speech, speaker conversion and speech morphing.

Imprinting emotional content on synthesized speech can possibly be achieved by means of morphing or warping some features of the synthesis to match characteristic features of a given emotion.

Concatenative speech synthesis is built upon the notion that speech can be reproduced by a relatively small number of sampled spoken units. Units vary in size, depending on the type of synthesis but all types of concatenative synthesis share the need for articulating or *stitching* pre-recorded bits. It is at these junctions that interpolation becomes of interest, since it makes it possible to eliminate or diminish the discontinuities therein.

Also useful for concatenative-based synthesis is data reduction. Phonetic databases for each individual speaker make use of large storage spaces for the recorded samples or *codebooks*. By means of speaker transformation, more speakers can be generated than the amount of *codebooks* that the system contains. Thus, morphing is a means of reducing the needed storage space for a given speaker diversity.

Speech morphing refers to the interpolation of two different recorded instances of similar[6] speech. Applications in speech morphing are much less general in nature than those sought in speaker transformation but allow for a much higher quality on each particular realization.

In speech manipulation and synthesis literature, we find two main tendencies: pitch-synchronous spectrum modifications and manipulations through source filter decomposition.

A considerable number of articles have been written on all of these subjects and it is well beyond the scope of this thesis to present an extensive review of the literature. We hereby present a few relevant ideas contained therein; ideas which can be carried over to the morphing of musical sounds.

2.2.1 Generation of emotional content in speech synthesis

In a review of the subject written in 2001, Marc Schröder[38] presents several solutions which have been proposed to this problem. Approaches are dependent on the type of synthesis that they address. This is, of course, due to the fact that affecting certain features of speech may prove to be easier or harder, depending on the speech synthesis technique in use. Nevertheless, all approaches share the goal of imprinting on synthesized speech some form of prosody, according to a set of rules. Additionally, voice quality is sought to be modified in speech synthesis models which allow for this. Prosodic modifications involve time-warping, f_0-warping (the rhythmic quality and melodic contour of speech), as well as amplitude modifications.

In a paper by Mareüil, Célérier and Toen[39] we can find the presentation of some rules regarding f_0 contour manipulation, time warping, amplitude manipulation and the repetition of some phonemes[7]. These rules were derived separately for English, French and

[6] Similar in length and content, since correspondence between the two instances needs to be established for morphing to take place.

[7] e.g. Stuttering for emulation of fear.

Spanish from actors who were recorded as they portrayed the set of emotions being studied.

2.2.2 Phonetic unit-articulation smoothing

As an example of work done for smoothing transitions between phonetic units used in concatenative speech, we take an article by Stylianou, Dutoit and Schroeter[40] on diphone concatenation. Diphone speech synthesis requires a library of diphones, which are small speech units with transitions from one phoneme to another. Diphone libraries are among the slimmest libraries for concatenative speech but, since they generally rely on time-based overlap-add methods, artefacts can easily arise due to discontinuities in the transition from one diphone to the next. The paper proposes modelling diphones with harmonic plus noise additive parameters and smoothly interpolating from one diphone's parametrization to the next during a short *mixing* region. One thing to notice in the results presented in the paper is that, although transitions from one diphone to the next are much smoother than those obtained via a time-based overlap add method, formants are broadened during the transitions.

Pfitzinger[41] presents a spectral morphing technique based on the derivative of the LPC-given spectrum. Where the spectral derivatives of the two spectra to be interpolated are matched through dynamic programming. The process of matching and warping the spectrum is called dynamic frequency warping (DFW). Using a function's derivative for locating peaks or valleys is a standard technique. The advantage of using this technique is that zero-crossings in the derivative necessarily equate to local maxima or minima in the spectrum and that the derivative slope is related to resonance bandwidth. Once the peaks have been found, a dynamic programming algorithm is used to find the best correspondence between the peaks of both interpoland spectra. Then, interpolation includes warping both spectral envelopes in order according to peak matches.

Compared to the magnitude interpolation of two spectra performed as a cross-fade, this method yields interpolation of spectra for which not only peak magnitudes are interpolated, but also peak frequencies. In other words, peaks do not just rise and fall at the same frequency during interpolation; rather, they slide toward their matching peaks on the target spectrum. Thus Pfitzinger claims that, with this technique, there is no broadening of formant regions at interpolation factors close to 0.5 and that formant frequency, amplitude and bandwidth are interpolated in a phonetically meaningful way.

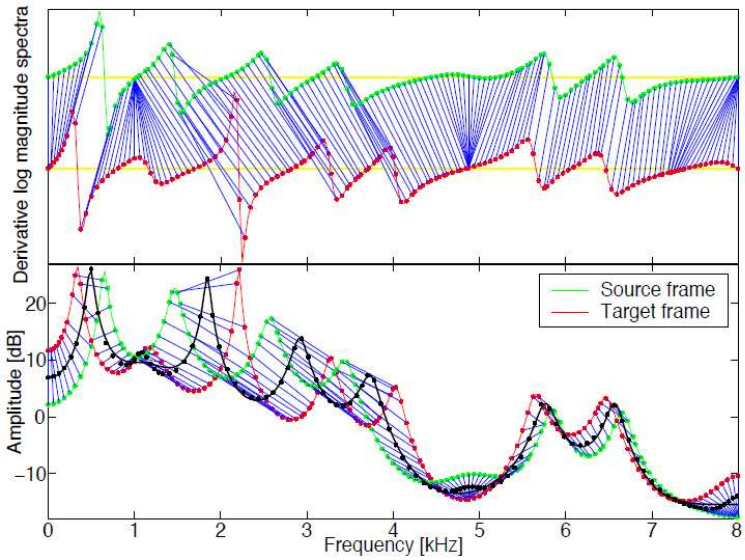


Fig. 2.1 An illustration of DFW proposed by Pfitzinger. At the top, both spectra's derivatives are matched through dynamic programming, the yellow line that crosses each derivative represents each one of their corresponding zeros. At the bottom, the corresponding spectra are interpolated along the matching lines.

2.2.3 Voice conversion

Voice conversion seeks to imprint the identity of one speaker onto the utterances of a second speaker. In the context of speech synthesis a *speaker* refers to a series of rules coupled with a codebook which is used by a particular synthesizer, and not necessarily to a human speaker.

Several speech synthesis methods rely on a data reduction technique called vector quantization which yields codebooks or sparse code. This technique extracts a discrete set of spectra (vectors) which make up a sampled or discretized version the whole spectral palette (or vector space) observed during a training period. The collection of sampled spectra is called a codebook. Several articles[42, 43, 44] have proposed a technique for voice conversion between two speakers in such a system. The proposed technique consists of a training period with matched utterances from two speakers which is used to generate a transformation codebook which maps the spectrally partitioned codebook of one speaker onto the second speaker's spectrally partitioned codebook. While this is useful for a complete transformation from $speaker_1$ to $speaker_2$, it does not allow smooth transitions from

one speaker to the next. Another notable limitation, is that this technique is only useful in cases involving no more and no less than two speakers.

The same sort of idea has been explored in later works by Iwasashi and Sagisaka[45, 46] with some modifications which make it relevant in the context of morphing and interpolation. The known speakers can be more than two. During the training period, the target speaker's utterances' spectral characteristics are matched by dynamic time warping of those from the known speakers. After the training period, the optimal weighting factor for the known speakers is found so as to more closely match the target speaker. Iwasashi and Sagisaka propose the use of cepstral coefficients and log-area ratios for spectrum interpolations.

It is worth noting that these and other works with similar approach aim at either finding correspondence between spectral characteristics or directly interpolating the spectra.

2.2.4 Speech morphing

Abe[47] proposed a time-domain pitch-synchronous-overlap-add (TD_PSOLA) based approach which consists of finding corresponding pitch marks (periods) for two given sources and interpolating the FFT of each period, afterwards, an $IFFT$ is performed on each period which is overlap-added to obtain the resulting morphed speech signal. Interpolation of the pitch period results in f_0 interpolation and the weighted sum of each corresponding *grains'* FFT yields a spectral envelope interpolation.

Ye and Young[48] propose morphing with a pitch-synchronous sinusoidal model which helps avoid the phase incoherence at each pitch period that results from TD_PSOLA-based strategies. Spectra are represented as Line Spectral Frequencies or Line Spectral Pairs (LSP) since they behave better than LPC coefficients during interpolation. Ye and Young found that, even while interpolating through LSPs, formant peaks became flattened out and spectra lost details toward an interpolation factor of 0.5 . Thus they propose to train the system to recognize the correspondence between the interpolated spectra–those which have lost detail–and transitional spectra observed during the training; once the system is trained, it can regain some of the lost spectral detail by combining the flattened-out spectra with their corresponding learnt spectra. Residual or transient spectra are matched between speakers in a training stage and the criteria for choosing correspondence between speakers' transient spectra is to retain the longest matching sequences from recorded transients.

In addition to the application of dynamic frequency warping to speech morphing, Pfitzinger[49] proposes the interpolation of the source signal through a similar time domain process, matching separated source signals in short (20 ms) windows by dynamic time warping and interpolating on a per-period basis. The results can be heard on Pfitzinger's home page[50], for which you may find a link here in the electronic version.

2.3 Some considerations from the context of image processing

The use of morphing in films has been well established and dates back to the very dawn of this industry. Experimentation began early on, at the beginning of the twentieth century with Georges Melies' transformations based on careful compositing[51]. Later films, such as the 1931 realization of *The strange case of Dr. Jekyll and Mr. Hyde* further advanced this effect. Later in the twentieth century, image morphing has yielded effects now memorable to pop culture, such as those found in Michael Jackson's music video for *Black and White* or those in the film *Willow*. In the last couple of decades, convincing visual morphs have become relatively common in film.

Writing a complete review of all that has taken place in this domain would be material enough for another thesis, and most likely not one for a degree in Music Technology. However, because we are dealing with morphing, it is relevant to invoke some aspects of image morphing for which we can find analogies in the realm of audio morphing.

2.3.1 Feature selection and image warping

In image morphing, feature correspondence between two or more images is often found by user specification, image warping is then performed and finally color interpolation is effected to obtain the morph. Color interpolation has several variants, such as the interpolation of raw RGB values; or the interpolation of hue, saturation and luminance. Even so, much of the emphasis of literature regarding image morphing seems to be placed on the partition of space given feature selection and the subsequent surface warping strategies; in their article on feature-based image metamorphosis, Beier and Neely[52] note that after warping the image, color interpolation is the simpler part of the process.

This is of interest to us as it reinforces the notion that an important part of the audio morphing process lies both in the careful selection of features that we wish to morph as well as in the mindful analysis and extraction of these features for each given realization.

The reader should note that in the present work we refer to time and frequency alignment when we talk about warping audio; so, these observations in regards to the major effort being invested into warping do not hold true for audio morphing. Yet, if we include general sonic feature selection and interpolation in the analogy, we can similarly observe that the interpolation of already matching features is the simpler part of the process. Thus, in many ways we can think of observations made in regards to feature selection and image warping as equivalent to the general feature selection and morphing process of an audio signal.

In his survey of image morphing, Wolberg[53] gives a detailed explanation of several feature selection and geometrical deformation strategies. The trend in feature selection tends to go toward less constrained strategies, where initial strategies were based on generating meshes and later strategies rely on simple lines and points with a variable degree of deformation *influence* from each feature to its surrounding areas. Another interesting idea presented by Wolberg in his survey is that of more significant morphs being achieved by allowing different rates of change for each one of the involved features[8].

In an article on the deformation of n-dimensional objects, Borrel and Bechmann[54] state that a simple, flexible and efficient procedure for achieving the deformation of an n-dimensional object is to map it to an m-dimensional object, where $m > n$, in order to perform a series of simple linear transformation on the m-dimensional object and then project it back onto an n-dimensional space.

We can extrapolate this idea of warping onto sound morphing in two ways: firstly, in some cases the selection of features might very well be viewed as having a higher dimensionality than the original sample-wise representation. Secondly, in order to avoid certain undesirable effects once we are interpolating a feature, we might have little option but to choose between interpolation paths that change some characteristic or another from both our source and target sounds[9].

2.3.2 Contrast loss along interpolation midpoints

Another finding from the image processing community that can be of interest for the purpose of audio morphing is mentioned in Grundland, Vohra, Williams and Dodgson's[55]

[8] Wolberg refers to this as *transition control*.

[9] As we will see in the corresponding section, this is the case of f_0 interpolation, where in order to avoid overbearing partial glissandi we must either choose to create phantom partials or to have an inharmonic sound resulting from the interpolation of two harmonic sounds.

article, *Cross Dissolve without Cross Fade: Preserving Contrast, Color and Salience in Image Compositing*. In this publication, the authors propose a way of solving the frequent loss of dynamic range which is likely to result from the interpolation of two or more data sets. Their solution is to weight the interpolation by salience masks applied to each one of the images. In the case of audio interpolation, we must bear in mind the loss of dynamic range and look for solution involving the interpolation of salient or otherwise perceptually meaningful representation of features whenever possible[10].

[10] As we will see in the corresponding section, this loss of contrast can be found in point by point spectral envelope interpolation. It can be avoided by recurring to the interpolation of an alternate representation of the spectral envelope, one which is more meaningful or alludes to perceptually salient features, as is the case of reflection coefficients.

Chapter 3

A Few Necessary Definitions

For the current explorations to be of any use to the reader, it's essential to clarify certain aspects that limit the context in which the strategies have been reviewed. As should be fairly evident from the title of the thesis, the current work presents ideas and results of an evaluation of spectral interpolation strategies for the morphing of musical sound objects; the synthesis of sounds that contain features from two or more sources within the musical context. Thus, it seems necessary to talk about what defines a musical sound object or a single-event musical sound, to briefly expose some concepts regarding interpolation and to explain the criteria for choosing the features across which interpolation will occur. The features or descriptors themselves will also be defined in the current chapter.

3.1 Musical sound objects

The present exploration has a well defined scope which is to perform a broad evaluation of strategies for morphing single-event musical sounds. In order to avoid confusion, it's necessary to state what the term *single-event musical sounds* implies. The terms *single-event* or sound object refer to sounds having a clear delimitation in the temporal domain: sounds which are flanked by either transients or silence. The reference to them being *musical* is not a reference to their subjective *musicality* but rather refers to their relatively stationary nature, as opposed to the extremely dynamic nature of other sounds such as speech utterances or general everyday sounds. This means that we are interested in sounds which possess relatively stable characteristics throughout their lifespan; sounds which are commonly found in music as we traditionally know it, such as a note from an instrument

or a percussion hit.

3.2 Dynamic time-warping

We recall Dynamic time warping was used for speech morphing and that surface warping was necessary for image morphing. The purpose of these warping processes is to align features of the morph targets, in time or on the $x-y$ plane respectively. Once this alignment has been found, we can either warp one of the targets to match the other or warp both according to the interpolation factor, so that both targets' features match. By the same token, during the interpolation of sound objects' temporal features, such as the amplitude envelope or vibrato, dynamic time-warping will be used extensively. Its application in this context aims at producing different time-scaling coefficients for each section between key features to be aligned; ie one time-scaling coefficient for the attack and another for the stable section and another one for the release.

3.3 About meaningful features

We need to define the parameters or features with which we represent sounds so that we can perform interpolation along the set of features and not along the raw data. The simplest form of interpolation between known data of inputs $x, y...z$, would be to interpolate between each of the samples $x[n], y[n]...z[n]$ at time nT_s, where T_s is the sampling interval[1]. Yet it is likely that this interpolation, or *cross-fade*, would result in an unconvincing morph since it would yield a perceptually distinguishable mix of multiple inputs rather than a unique hybrid containing features from all inputs[4]. Thus, we will define a feature set that describes the inputs and for which it is possible to interpolate each feature. This set will contain descriptors used to warp for alignment along the temporal and frequency domains; descriptors that define the event's amplitude through time; descriptors that define the relationship and behaviour between the deterministic and stochastic components of the additive representation as well as descriptors which define the relationship between partials of the sound.

[1] provided, of course that T_s is a constant for all signals at all times.

3.3.1 Why do we need them?

In order to achieve a convincing morph, it is crucial to note we are not limited to directly interpolating samples from signals themselves, since it is possible to approximate sufficiently well the known data in parametric fashion by means of analysis performed on the signal or on its transformations. One such example of parametrization is the additive sinusoidal representation, which unveils the possibility of achieving interpolation of features such as the frequencies or amplitudes of each partial. Further analysis steps might help represent the data with higher-level parameters which will prove useful for obtaining features that are true hybrids.

As an example of why it is important to establish meaningful features let us consider two morphings effected from two input signals which are represented in two ways–the later being more meaningful. We shall take our signals to be two notes, each having a full harmonic spectrum, produced by the same instrument with differing f_0 and vibrato rates. For simplicity's sake, let f_0 and $f_{vibrato}$ be 100 Hz and 3 Hz respectively for the first sound and 125 Hz and 2 Hz for the second. Let's also suppose the modulator for the vibrato to be a pure sine and the interpolation factor to be 0.5.

Case 1 - Linear interpolation of the raw data.

We would obtain a hybrid containing energy at 100 Hz, 125 Hz, 200 Hz, 250 Hz, 300 Hz, 375 Hz, 400 Hz, 500 Hz and so forth. Where each partial that is a multiple of 100 Hz would be modulated at 3 Hz and each partial that is a multiple of 125 Hz would be modulated at 2 Hz–the partial at 500 Hz would display modulations at both of these frequencies. Modulations aside, if we simply think of f_0 as the maximum common denominator for all present partials, we would arguably perceive a note with an f_0 of 25 Hz and a very irregular spectrum which is missing, among others, the first three partials. But then again, if we do consider modulations and we consider the cues that synchronous modulations give for source separation[56] we can actually argue that the interpolation will yield an event which can be perceived as the mix of two distinct signals.

Case 2 - Linear interpolation of the additive model.

If we were to match partials and linearly interpolate their frequencies and amplitudes, with an interpolation coefficient of 0.5, we would obtain a harmonic structure with an f_0 of

112.5 Hz. In this regard, we would now perceive a true hybrid. It is when we think of the modulations of the partials that we realize that the features used for morphing are still problematic since the frequency modulation itself will contain components at both 2 and 3 Hz as opposed to a pure sine at 2.5 Hz.

In essence, this short example serves to show that as long as we don't precisely define the set of features–such as vibrato frequency or f_0–that we wish to interpolate from the inputs and as long as we don't interpolate these features directly we will keep facing the same effect at one level or another. It then becomes clear that in order to achieve a convincing morph, we must define the set of features which we will use to represent sounds and over which interpolation will occur. These features will generally be derived from at least one, and most often more than one, step of analysis performed on the original data. Thus, because the vast majority of features that interest us are obtained from transforming and analyzing the sample-wise or lower-level representation of audio, they are commonly referred to as *higher-level features*[57].

Wesley Hatch has already done some work in terms of defining a set of *higher-level features* for the purpose of morphing[7]. However, in the present work, we are interested in obtaining a larger set of features which describe sounds within the musical context in as meaningful a way as possible; thus we will diverge from this definition and allude to *meaningful features*.

3.3.2 How do we go about defining them?

There are a few things we can establish before embarking on the quest for meaningful features:

- We are interested in quantifiable features
- Features should be chosen to be as independent as possible
- Features should be perceptually meaningful
- Entire feature-set should describe the sound as completely as possible

Features should be perceptually meaningful

Ideally, there should be a link between a feature and some audible quality of the sound. Features such as f_0 have much more correlation with what we hear than perceptually

irrelevant and arbitrary characteristics such as if f_0 is prime or not. Features with a stronger correlation to what we can hear are more perceptually meaningful.

We are interested in quantifiable features

Since interpolation consists in finding an intermediate value from surrounding values, it follows that a necessary condition for interpolation of a given feature is that it be quantifiable.

Features should be chosen to be as independent as possible

It's quite likely that our extracted features will not be completely independent. For example, the frequencies of the extracted partials in our previous examples contain both constant partial frequencies (i.e. a carrier frequencies) and vibrato (i.e. a modulator) and would require an additional analysis step, such as that proposed by Marchand and Raspaud[58], to separate these two features. The more we can enforce independence in terms of perceptually meaningful features in our representation, the easier it will be to use our feature set for interpolation.

Entire feature-set should describe the sound as completely as possible

We've previously stated that it is desirable to avoid morphs which contain elements from each one of its inputs which can be perceived as separate and distinct elements (such as the prior example with a composite vibrato). It then follows that we should choose a set of features that describes sounds as completely as possible to avoid non-interpolated outcomes of isolated features.

3.4 Presentation of meaningful features

After establishing what it is that we seek from a set of descriptors, we are ready to choose the features that will conform the set which we'll use for interpolation. The following features will be used wherever applicable: f_0, amplitude envelope, spectral centroid during attack, spectral shape, vibrato[2], peak-amplitude time, odd to even partial amplitude ratio, deterministic to stochastic energy ratio and inharmonicity. Whenever possible, the

[2]We will use an extended vibrato definition which includes FM, AM, and SEM as defined by[59].

descriptors have been grouped according to whether they are only useful for sounds with an underlying harmonic structure or they are useful for warping and morphing sounds regardless of them having an underlying harmonic structure.

As Peeters[60] has noted, for some features such as the number of partials or the amplitude envelope, analysis will yield a single result for every input sound[3]; while for others such as vibrato (seen as a modulation) or spectral envelope, analysis will most likely yield one or more results per analysis frame. Optionally, some of these results might be consolidated by averaging them over time, which gives two possible representation and interpolation strategies.

Since we will be using an additive model with stochastic and deterministic part decomposition, we must take into account that each one of these parts should have descriptors in its own right, where the stochastic part will not contain descriptors that are useful for harmonic structures. Furthermore, interpolations for the stochastic and the deterministic parts should be independent.

The rest of the chapter explains each one of these features in detail and table 3.1 condenses important information regarding the set of descriptors.

3.4.1 Features used for morphing regardless of a harmonic structure

We will first review descriptors which can be used regardless of whether a harmonic structure is present or not. They are the amplitude envelope, the onset spectral centroid and the spectral envelope. Strictly speaking, we will use the amplitude envelope for temporal warping for alignment and not for morphing.

Amplitude Envelope

The function of amplitude vs time for any given signal. This function can be obtained by calculating the RMS of windowed portions. For the envelope extraction, Peeters[60] proposes a window with a size of 100 milliseconds.

$$A[k] = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1}\left(x[n+k]w[n]\right)^2} \tag{3.1}$$

[3] A single ADSR can describe the envelope or a constant number of partials will be valid throughout the duration of the sound

Where $A[k]$ is the amplitude at analysis frame k. Once the function has been extracted, the envelope can be approximated by an ADSR for most instrumental sounds. This is achieved by fitting points of inflection within $A[k]$ to the ADSR model.

Onset spectral centroid

As Grey has shown[10, 11] in his well known studies of timbre, the distribution of energy content during the attack is an important factor in our perception of timbre. The element that separates this from any other spectral distribution feature is that it is only the ratio at the event's onset[4] that interests us. Grey does not, however, give a clear-cut definition for this and we will thus define it as:

$$OSC[m] = \frac{\sum_{n=0}^{N-1} f_n \cdot a_{mn}}{\sum_{n=0}^{N-1} a_{mn}} \tag{3.2}$$

where $OSC[m]$ is the onset spectral centroid at frame number m from 0 to $M-1$ and where M is the number of frames corresponding to the attack. N represents the number of bins, f_n represents the frequency at bin n, a_{nm} represents the amplitude at bin n during frame m.

Although the onset spectral centroid can be useful to describe any given sound, imposing a given OSC upon re-synthesis can be achieved through a variety of spectral modifications. When we have a smaller set of partials, we propose to keep track of partial peak amplitude times and interpolate them instead. This will be described with more detail in 3.4.2

Spectral shape

The spectral shape or spectral envelope of a particular sequence $x[n]$ is a curve that fits over the peaks of its spectrum. The spectral shape is typically characterized by how many peaks it tries to fit and by its smoothness in some cases. Alternatively, from a source-filter perspective, it can also be seen as the frequency response of the filter that we would need to apply to a flat-spectrum signal in order to obtain a similar spectral distribution to that of the sequence from which we obtained the envelope in the first place. Additionally, since the spectral envelope can be viewed as the frequency response of a filter, it can be parametrized

[4] And possibly during the release portion.

as a series of reflection coefficients, which guarantee it's stability during the interpolation process[61].

3.4.2 Features found only in pitched sounds

From the set of descriptors that we have defined, those that will be of interest to us only in the case of sounds with an underlying harmonic structure are vibrato, inharmonicity, even-to-odd partial energy ratio and Partials' peak attack times. As in the case of the amplitude envelope we will actually use f_0 for spectral warping for alignment and not really for morphing.

Fundamental frequency: f_0

For any sound containing a harmonic series of partials, f_0 is a frequency such that f_0 is the greatest common divisor for the frequencies of all partials. f_0 is correlated to the notion of pitch in the perceptual domain. As has been mentioned before, it will be important to warp pitched sounds along this feature to avoid a hybrid that has an f_0 that is not necessarily between that of the sources or a hybrid that can be perceived as having two distinct f_0s.

Vibrato

While it's classically understood to mean a Frequency Modulation at a sub-audio rate, whenever we talk about vibrato in the current study we will be referring to a generalized vibrato as the one defined by Verfaille, Guastavino and Depalle in their *Perceptual Evaluation of Vibrato Models*[59].

The generalized model of a vibrato is not only defined as Frequency Modulation but also comprises sub-audio rate Amplitude and Spectral Envelope Modulations. AM, when present at a sub-audio rate is typically referred to as tremolo, and SEM can fall under the general category of Spectral Flux. These three modulations generally occur at correlated frequencies, where for any particular type of sound, there is an almost constant relationship between modulation phases[5] and each modulation has it's own amplitude or depth[6]. The

[5]It is an almost constant relationship, since we will find hysteresis given the most common of cases, as has been shown in the aforementioned paper[59].

[6]Of course, amplitude modulation depth, frequency modulation depth and the n-dimensional SE maximum-difference vector are like apples, oranges and pineapples, in other words, units for each depth will be unrelated to other modulation depths.

classical approach to vibrato or tremolo was to ascribe a single frequency of modulation throughout the event, but it's preferable to extract these modulations by performing a second order sinusoidal analysis, as has been shown by Marchand and Raspaud's DAFX article on time stretching[58]. This way, we can extract the three sets of modulation partial *tracks*—equivalent to the partials in regular additive modelling—by performing an $STFT$ on the up-sampled original partial track frequency and amplitude fluctuations. Spectral envelope modulations are implicit if we perform amplitude modulation extraction for each harmonic as opposed to doing so only on the first partial.

Inharmonicity

Generally speaking, when we talk about pitched sounds or harmonic series of partials, there is a slight deviation from purely harmonic relations. This is caused by specific characteristics of the physical principle for sound generation. For example, string geometry for an oscillating string or bore irregularities in a wind instrument slightly affect the frequencies of different partials. Thus, for each different sound, we will often find a particular series of deviations from purely harmonic relations between each harmonic and the fundamental frequency.

Inharmonicity refers to this deviation of partial frequencies from their expected frequencies as per $h \cdot f_0$. Peeters proposes inharmonicity to be:

$$inharmonicity = \frac{2\sum_h |f_h - h \cdot f_0| a_h^2}{f_0 \sum_h a_h^2 \cdot h}$$

With a value of $[0, 1]$. We propose to extend the range to $[-1, 1]$, where negative values correspond to generally compressed harmonic spectra and positive values correspond to stretched harmonic spectra. We can avoid the use of absolute value from the previous equation to know if the trend of the inharmonicity corresponds to compression or stretching. So we would have:

$$inharmonicity = \frac{2\sum_h (f_h - h \cdot f_0) a_h^2}{f_0 \sum_h a_h^2 \cdot h} \tag{3.3}$$

The prior definition allows us to consider a single coefficient that measures the overall inharmonicity of a given sound this is a broad characterization which is generally insufficient if we intend to reconstruct the sound from such a characterization. It is only in certain cases where inharmonicity relations have been studied, such as is the case of piano sounds [62]

that a single measure of inharmonicity would suffice to approximate individual coefficients for each partial during re-synthesis. We will find that, in most cases, a complete vector of inharmonicity coefficients will be needed in order to reconstruct each partial's frequency given an f_0. Such a vector could be denoted as follows:

$$Inharmonicity = i_h = \frac{2(f_h - h \cdot f_0)}{h \cdot f_0} \tag{3.4}$$

Where h is only relevant for partials above the one with a frequency which coincides with f_0, if such a partial were present[7].

Even to odd partial energy ratio

The ratio between these two energies is correlated with spectral smoothness and plays an important part in our perception of timbre[63]. A typical example of a very low even to odd harmonic energy ratio is a stopped pipe, e.g. the clarinet, played *piano* at the lower end of its register. An example of an instrument which has a ratio of approximately 1 is the trumpet. High EOR values will tend to sound an octave above the fundamental pitch. Even to odd partial energy ratio or EOR can be calculated as follows[60]:

$$EOR = \frac{\sum_{h=1}^{\lfloor H/2 \rfloor} a_{2h}^2}{\sum_{h=1}^{\lfloor H/2 \rfloor} a_{2h-1}^2} \tag{3.5}$$

Partial attack times

We recall that the spectral fluctuation during attack has been described as a correlate of instrument families[10, 11] and that in 3.4.1 we proposed an alternative to the onset spectral centroid can be useful in cases where a relatively low number of partials are present[8]. The proposed alternative is to have a vector of times at which each partial attains its peak amplitude, which can be imprinted on the attack of a series of partials with less ambiguity than the onset spectral centroid. Similarly to the onset spectral centroid, partial attack times can describe the spectral fluctuation of the attack.

[7] Allowing for pitched sounds with a phantom fundamental component.

[8] Since we can consider the deterministic plus stochastic representation as a data-reduction technique from the $STFT$, we can generally say that a number of partials describing a sound is significantly smaller than the number of bins required for the same purpose.

In regards to the amount of data needed for describing onset spectral fluctuations, we have a few observations. Keeping a vector of OSC values across all frames for long attacks[9] is not too far from keeping a single attack time for each partial. Also, if there were a need to reduce the data in the representation, we could approximate the time of reaching peak amplitude as a function of partial number.

Once we have created an additive deterministic and stochastic model through analysis, we can easily keep track of the times at which peak values are attained for each partial. We can do so with a minimum error equivalent to half of the partial frequency sampling time which is given by the analysis hop size[10].

3.4.3 Other considerations

Once additive model separation has been done, we are dealing with (at least) two entities for each sound; its deterministic and stochastic components. It's possible to extract some characteristics, such as spectral envelope or amplitude envelope from the sound before performing deterministic and stochastic component separation as well as doing so after performing the separation. Depending on the application, it is important to weigh the qualitative difference between the outcomes of these two procedures against their added cost in terms of processing and storage.

Harmonic-part to noise-part ratio

After having performed the separation of harmonic and noise components of a signal, a global value may be obtained by calculating the ratio of the energy of each component. Alternatively we may obtain a sequence of power values at each frame, for both the deterministic and stochastic components, and then store a vector of power ratios across all frames. Peeters proposes harmonic part energy and noise part energy as two separate features[60], but consolidating the two into a ratio diminishes the number of features and avoids the overlap with the amplitude envelope.

[9] If we consider a $0.5s$ attack with an analysis framerate of $86Hz$, we would contemplate 43 onset spectral centroid values.

[10] At a sampling rate of 44,100 Hz, a hop size of 256 samples would thus be accurate to within 2.9 milliseconds which should prove to be sufficient.

3.5 On interpolation

Interpolation is a means to find an intermediate value between known samples. Mathematically this is achieved by fitting a function to pass through a set of known points[11]. Many common interpolation schemes, such as cubic interpolation, require more than two known points through which a function must be fitted. In general we will interpolate between two given sounds, which means that our data-set for each feature will consist of two samples. This leaves a choice from three common interpolation strategies: nearest-neighbour, linear interpolation and logarithmic interpolation–or linear interpolation performed on the logarithm of the known points.

Seeking the value $y_{n+\alpha}$ by these three different types of interpolation, where $0 \leq \alpha \leq 1$ is given by the following expressions:[51, 7]:

nearest-neighbour interpolation

$$y_{n+\alpha} = \begin{cases} y_n & \text{if } \alpha < 0.5 \\ y_{n+1} & \text{if } \alpha \geq 0.5 \end{cases}$$

linear interpolation

$$y_{n+\alpha} = (1-\alpha)\, y_n + (\alpha)\, y_{n+1}$$

logarithmic interpolation

$$y_{n+\alpha} = y_n \left(\frac{y_{n+1}}{y_n} \right)^{\alpha}$$

Although nearest neighbour interpolation can be considered as more of a decision-making strategy than interpolation, we have decided to mention it, as does Wolberg in his presentation on the subject[51]. This strategy would be inappropriate in most scenarios which aim at morphing, since it does not really find intermediate values between known points but merely opts for one or the other. As for linear and logarithmic interpolations, choosing between them shall be part of evaluating strategies for different features. We hereby hypothesize that in some features logarithmic interpolation will be preferable since it better corresponds to our perception of the evolution of the parameters themselves. For example,

[11] For many types of interpolation, we can alternatively see this as convolving each point with a given kernel function[51]. This is not so for polynomial interpolations.

it is well known that the linear evolution of the log_2 of frequency is strongly correlated with the perception of pitch, therefore it is likely that interpolation of f_0 for two given inputs might seem preferable when performed with logarithmic interpolation than when performed with linear interpolation.[12]

Table 3.1 Perceptually meaningful descriptors: features can be affected during time-warping, spectral-warping or during the morphing process. Features can be exclusive to the deterministic component and they can be seen as one measure—or an average measure—per event or can be seen as a time-series of measures per event.

Descriptor	Used during warping(w) or morphing(m)	Found only in deterministic component	Single measure or average	Time series
Amplitude envelope	w			*
Spectral centroid during attack	m		*	*
Spectral shape	m		*	*
f_0	w	*		*
Vibrato	m	*	*	*
Inharmonicity	m/w	*	*	*
Odd/even partial amplitude ratio	m	*	*	*
Partial attack time	m	*	*	
Harmonic/noise amplitude ratio	m	*	*	*

[12]Ideally, interpolation would be best performed within all perceptually validated scales, but since this is not contemplated within the scope of the present work, the extrapolation of the results herein presented into the perceptual domain is left as future work.

Chapter 4

Experimental Observations and Results for synthetic sounds

The stated goal of the present study, is to evaluate several different interpolation strategies which can be used for the purpose of morphing between single-event musical sounds. This chapter presents an evaluation of isolated interpolation strategies for features which we defined in chapter 3. These strategies were tested on pairs of synthesized sounds which vary exclusively and heavily in terms of the feature being tested. We present the features in the same order as we did in chapter 3. In the presentation and comparison of interpolation strategies, we've commented in regards to the desirability of using one strategy versus another. In some cases, more than one strategy may produce viable results. In these cases, we discuss the usefulness of each strategy under different circumstances.

We have also posted sound files on the project's website[9] so as to provide an *audible illustration* of the results. These sound files are intended to be complementary to the observations and figures presented herein to illustrate these comparisons. The reader is encouraged to refer to these samples.

4.1 Amplitude envelope warping and alignment

An important first step for audio morphing consists in finding the time-alignment of the involved sounds, this will facilitate the subsequent performance of dynamic time-warping. The warping and alignment of the amplitude envelope or $A[n]$ involves dealing with both amplitude and time. For each one of these axes, there are two obvious strategies which are

logarithmic interpolation and linear interpolation.

Having two possible interpolation strategies for two different aspects of the temporal envelope yields four possible strategies for interpolation: linear time and linear amplitude; linear time and logarithmic amplitude; logarithmic time and linear amplitude, and finally, logarithmic time and logarithmic amplitude. For all strategies, the envelope was interpolated in 5 steps, where the interpolation factors were: 0.0, 0.25, 0.5, 0.75 and 1.0.

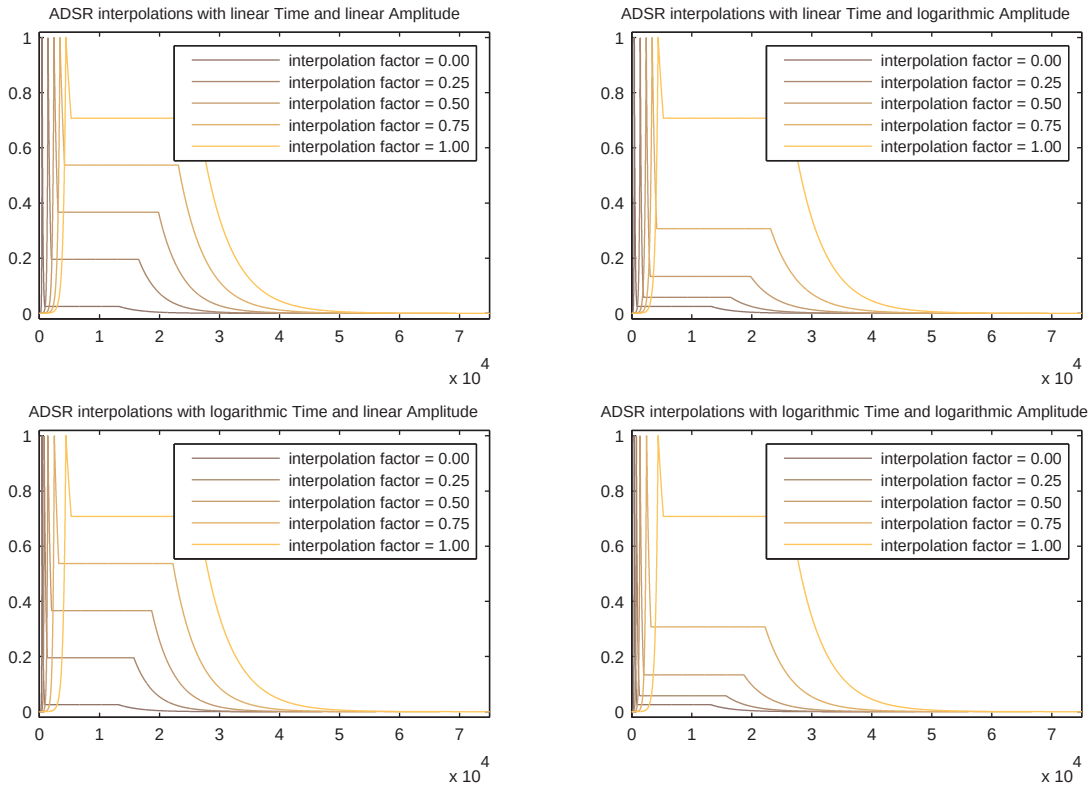


Fig. 4.1 Comparison of amplitude envelope interpolation strategies, the corresponding audio files can be found in the the project's web page.

We generated the sinusoidal model for a band-limited sawtooth-wave at 441 Hz in Matlab and then used Ircam's *pm*2 to synthesize the wave. Two arbitrary contrasting amplitude envelopes were created. Then the envelopes were interpolated with the different combinations of strategies. Afterwards, the sawtooth-wave's gain was scaled by the set of interpolated envelopes. Lastly, we compared the results of the four different interpolation strategies in an informal listening test.

The listening test showed the choice of logarithmic interpolation along both, time and

amplitude, dimensions to be superior to linear interpolation. The envelope interpolations are depicted in figure 4.1 and the resulting audio samples can be accessed on the project's website[9]. The following list contains links to playlists in order to provide easy access for readers of the electronic version:

- linear time, linear amplitude,
- linear time, logarithmic amplitude,
- logarithmic time, linear amplitude and
- logarithmic time, logarithmic amplitude.

4.2 Morphing spectral envelopes

The interpolation of spectral envelopes or $E(f)$ will generally be performed separately on the deterministic and stochastic component. This would imply that our choice of strategy for spectral envelope morphing is perhaps of greater importance than that of other features. The comparison that we carried out uses deterministic component. However, since the component is a spectrally-rich source, the results should be readily applicable to stochastic-part representations.

Spectral envelopes can be obtained through various analysis techniques[64] and they can also be represented in several exactly equivalent or approximately equivalent ways[21]. Because the combination of spectral envelope analysis techniques, spectral envelope representations, and types of sounds which can be analyzed is rather large, we have chosen the true envelope estimation, which seems to be accurate and well-behaved in most circumstances, albeit computationally expensive. We mention different analysis techniques, leading up to the one we have chosen. Afterwards presenting possible representations of spectral envelope information and finally reporting on the implementation of the comparison of spectral envelope interpolation strategies. Note that we can find equivalences between $E(f)$ representations, some exact and some approximate, regardless of the method by which the estimation was performed[21].

4.2.1 On spectral envelope estimation

A spectral envelope obtained from spectral analysis of a given signal is an approximation that relies on smoothing spectral information as a function of frequency[1] in one way or another[2]. Therefore, the way in which the spectral envelope is estimated will greatly impact the sort of envelope we obtain. Schwarz and Rodet[64] wrote an article which compares different spectral envelope estimation techniques. The following descriptions are extremely simplified but are sufficient for giving the reader a broad idea in regards to spectral estimation.

The simplest forms of approximating spectral envelopes are given by the line-segment approximation of peak values[3] selected from each section of a partitioned frequency domain.

Linear predictive coding gives a good approximation of spectral envelopes but requires significant tuning of the model order. This tuning calls for either prior knowledge or for making assumptions about the signal to be analyzed.

Another way of estimating the spectral envelope is to low-pass filter a given signal's spectrum. This is the way in which cepstral approximations of the spectral envelope are obtained. By choosing only the low frequency bins of a FT of the logarithm of the FT of the signal, we eliminate abrupt changes (higher *quefrencies*[4])in the spectrum. The problem with this approach is that the estimated spectral envelope generally turns out to be significantly lower in magnitude than the actual peaks in the spectrum.

The discrete cepstrum yields a spectral approximation that, like the cepstrum, is also a sum of sinusoidal functions. The difference is that the discrete cepstrum is an estimation of the spectral envelope comprised by a sum of sinusoids which must pass through a discrete number of points taken from the original spectrum. This method gives a better approximation than regular cepstral estimation when we have adequate information in regards to the spectral peaks and the order is chosen appropriately. Yet, despite the improvement in accuracy, discrete cepstral estimation can produce inaccurate envelope estimations when the constraints are too tight for the order that has been chosen[5].

[1] In other words, smoothing each spectrum.

[2] This is not necessarily the case for envelopes obtained otherwise, for example those that derive from knowledge of a particular physical model.

[3] whether we are referring to a single peak or some sort of peak averaging.

[4] In cepstral analysis we obtain the FT of the logarithm of the FT of a signal–*quefrencies* are to the *cepstrum* what frequencies are to the standard spectrum.

[5] e.g. When two points with very different magnitudes are too close in the frequency domain.

An alternate way to counter the low magnitude yielded by cepstral analysis is to iteratively compute the spectral envelope for the peaks that remain larger in magnitude than the spectrum. This is called a true envelope estimation.

4.2.2 On spectral envelope representations

We now focus on the comparison of the interpolations performed via different representations of the spectral envelope. All of the strategies that we tested relied on representations which were derived from a true envelope estimation.

True envelope estimation yields either cepstral coefficients or an equivalent magnitude spectrum. Based on the Wiener-Khinchine theorem, we can convert a magnitude spectrum into an autocorrelation sequence by transforming it into a power spectrum and applying an $IFFT$ to it. Having an autocorrelation sequence is an intermediate step toward being able to represent the spectrum as an auto-regressive model. From the transversal coefficients, we can easily obtain a series of reflection coefficients, log-area ratios or line spectral pairs.

4.2.3 Comparison of $E(f)$ interpolation strategies

We created a pair of arbitrary spectral envelopes with very different characteristics. The first arbitrary spectral envelope has two resonances: the first resonance, with a peak amplitude of 1, is at $500Hz$, with a bandwidth of $25Hz$; the peak of the second resonance is at $1150Hz$ with a magnitude of 0.5 and a bandwidth of $50Hz$. The target arbitrary envelope contains a single broad resonance at $1500Hz$ with a peak magnitude of 1 and a bandwidth of $100Hz$. A 90-partial sawtooth-wave with $f_0 = 85Hz$ was generated again through pm2. Then, for each strategy, the magnitudes were scaled by a series of 32 spectra obtained by interpolating the two arbitrary envelopes. The choice of a low fundamental frequency was made in order to have closely-spaced harmonics, revealing a good degree of detail in the spectral envelope. This was deemed important since we did not include frequency modulations in the audio examples.

We compared four interpolation strategies: naive linear and naive logarithmic cepstral coefficient interpolation; reflection coefficient interpolation and log-area ratio interpolation. These four strategies can be clearly grouped into two types: the *cross-fade* or bin-by-bin interpolation of spectra and the interpolation of a representation with a stronger correlation to a physical representation. Both naive interpolations belong to the former category and

the remaining interpolations pertain to the latter. The resulting synthesized interpolations can be heard on the project's website[9] and each strategy is presented in the following paragraphs.

Naive cepstral coefficient interpolation, linear and logarithmic interpolations

The naive interpolation of cepstral coefficients tends to give poor results with interpolation factors nearing 0.5. Resonant frequencies of the peaks of the spectral envelopes remain constant throughout the interpolation. Thus, toward the mid-point in the interpolation we tend to have as many spectral maxima as the sum of the number of resonances[6] in both spectra. However, the interpolated peaks would all present a lower amplitude than their original counterparts. This tends to produce a relatively flat spectrum which parallels the loss of contrast addressed in image morphing in subsection 2.3.2. This relative *flatness* can be observed in figure 4.2. Both audio examples can be found on the project's website[9]. They have also been linked in the electronic version, click here for the linear interpolation and here for the logarithmic interpolation.

Interpolation of reflection coefficients and log-area ratios

Naive interpolations did not prove to be convincing given that there was no sweeping of resonant frequencies and that it was found that interpolations with an interpolation coefficient close to 0.5 could produce relatively flat spectra. Therefore, we then tried two interpolation strategies that address these problems: The interpolation of reflection coefficients and of log-area ratios. The former offer two important advantages over other spectral interpolation techniques. Firstly, if the magnitudes of all coefficients for both source and target spectra are limited to 1 then system stability is guaranteed throughout the interpolation[61] since no interpolated coefficients will have a magnitude larger than 1. Secondly, their interpolation produces more meaningful spectral changes because they can be seen as being derived from the ratios between cross-sectional areas of a series of cylindrical sections which constitute a propagation path. That is to say, they are meaningful changes since they reflect the acoustic behavior of interpolating the sampled diameters for propagation paths: resonant frequencies move as well as magnitudes and bandwidths.

[6] If the spectral maxima or resonances from the two original spectra do not overlap during the interpolation, otherwise we would have less peaks than the total number of peaks.

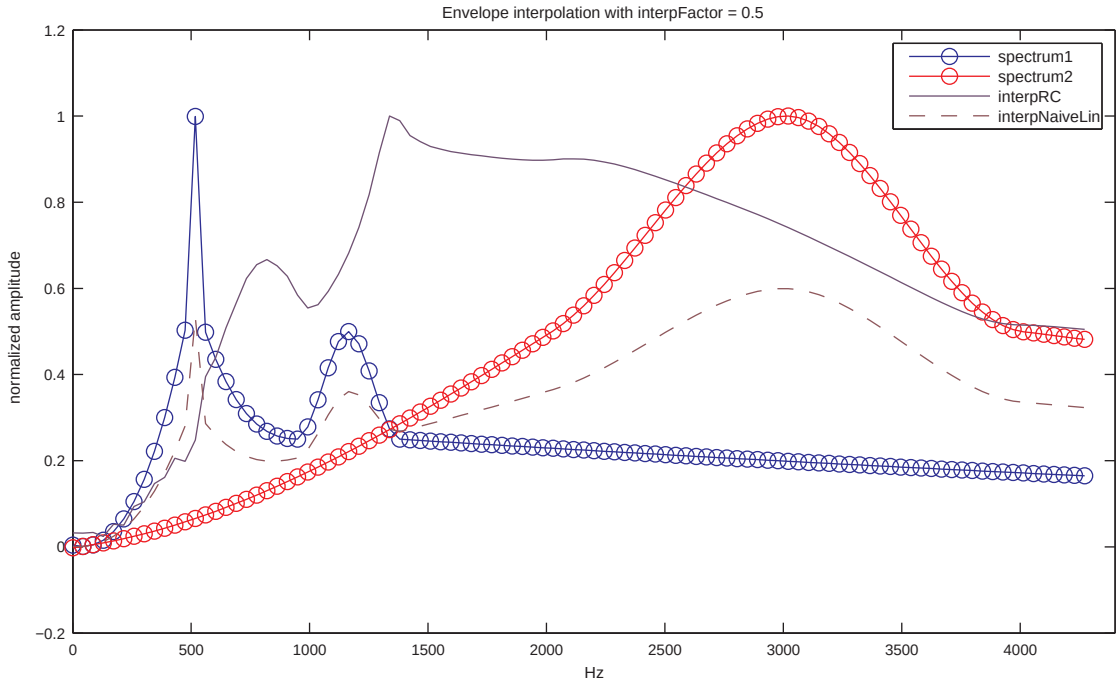


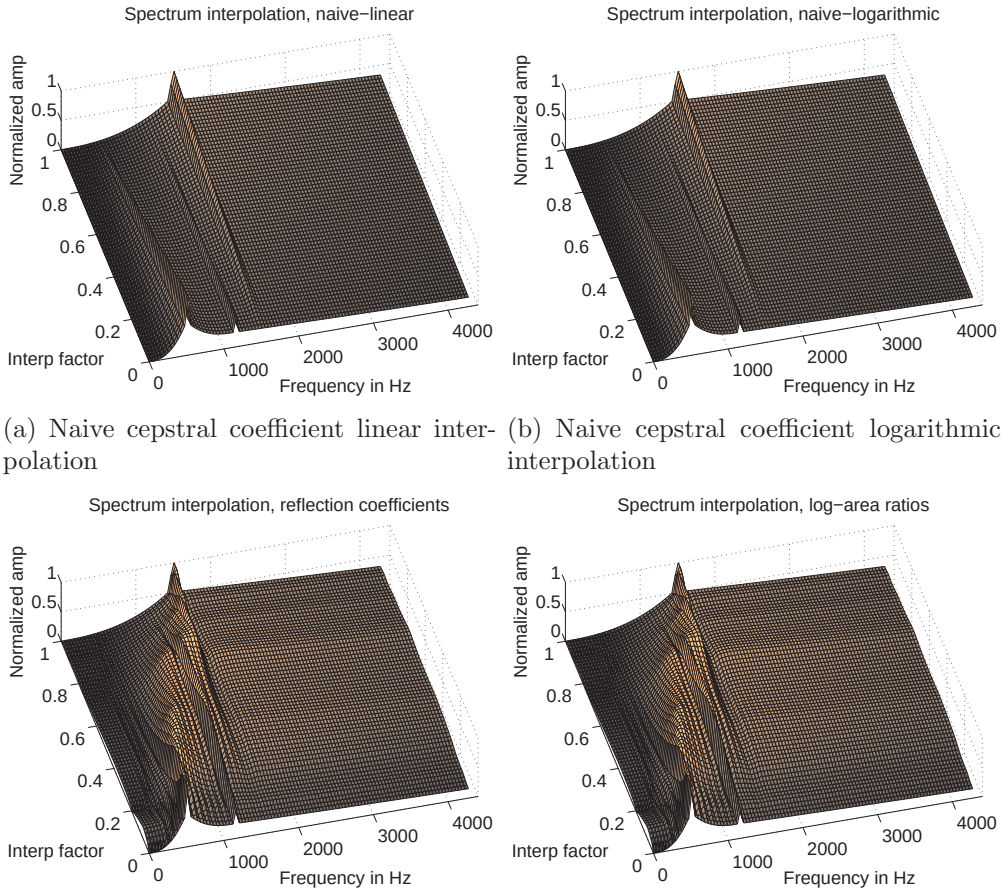
Fig. 4.2 Flattening of spectral envelope obtained by naive interpolation of the cepstral coefficients at an interpolation factor of 0.5 vs. that of a spectral envelope obtained by the interpolation of reflection coefficients with the same interpolation factor.

Log-area ratios are similar to reflection coefficients but are often preferred in the speech processing community since they are more directly related to the physical properties of the vocal tract. Log-area ratios are related to reflection coefficients in that $lar_k = log\left(\frac{1-r_k}{1+r_k}\right)$ where r_k is the k^{th} reflection coefficient[65].

In figure 4.3 we can observe the difference between naive and non-naive interpolation strategies. It is plain to see in this figure that not only do the spectral envelopes flatten out toward the middle of the interpolation on both naive cases whereas *peakiness* is maintained throughout both reflection coefficient and log-area ratio interpolations. Another difference which is apparent to the naked eye is that the resonant frequencies do not move during naive interpolation, whereas figures 4.3(c) and 4.3(c) present a frequency sweep along interpolation paths. The frequency sweep is, perhaps, ideally smooth but the results are a considerable improvement over the results from naive interpolation.

Both audio examples can be found on the project's website[9]. They have also been linked in the electronic version, click here for the reflection coefficient interpolation and

here for the log area ratio interpolation.



(a) Naive cepstral coefficient linear interpolation

(b) Naive cepstral coefficient logarithmic interpolation

(c) Reflection coefficient interpolation

(d) Log-area ratio interpolation

Fig. 4.3 Four spectral interpolation strategies, two bin-per-bin interpolation strategies, reflection-coefficient interpolation and log-area ratios interpolation applied to the same two arbitrary spectral envelopes. Meshes were obtained by performing 32 interpolation steps between two 512-point spectra.

Line Spectral Pairs or Frequencies

An additional strategy was tested but gave very poor results in the present context. In speech processing, it is common to use line spectral pairs, also called line spectral frequencies, for envelope representation[48]. But in this context, we found that as the order of our representations increased, interpolation of LSP introduced considerable inaccuracies and

discontinuities. Figure 4.4 represents an interpolation of the same two spectra which were used for previous examples. In this case, the order is reduced to 64 frequency sampling points as opposed to 512 which were used in prior realizations. Already with this order we start to see some discontinuities, such as the spurious peaks throughout the higher frequencies, as well as a gap in the resonance toward interpolation factor 0.8. Higher orders produced increasingly disparate results.

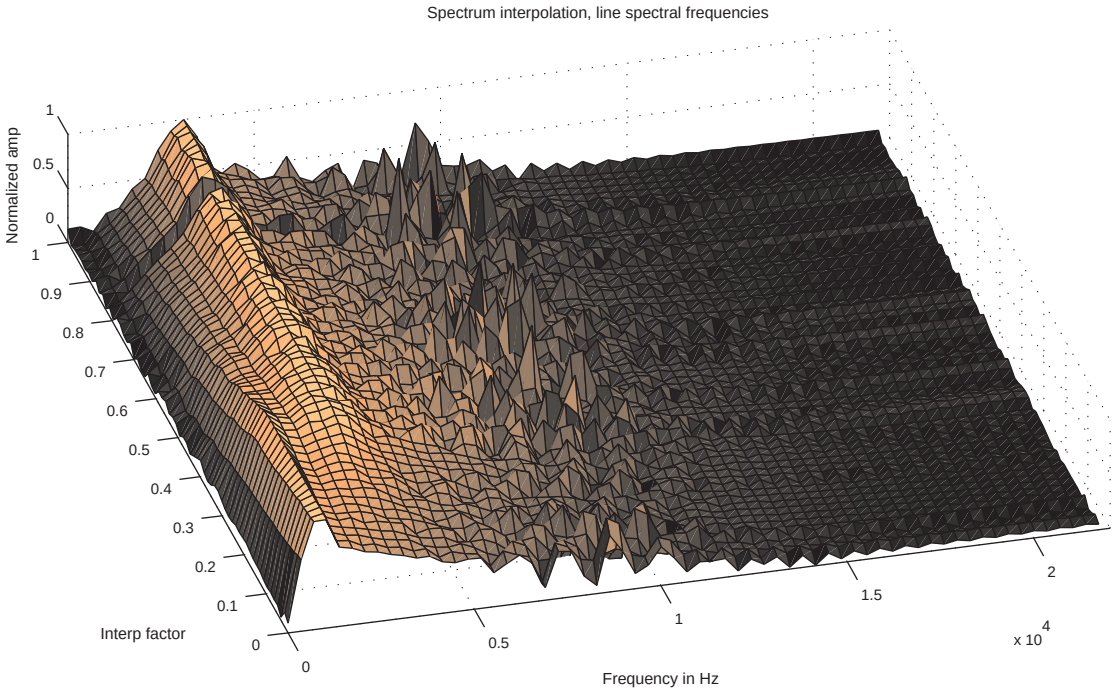


Fig. 4.4 Line spectral pairs interpolation for the same two envelopes that were used for prior interpolations. With as little as 64 sample points per envelope, spurious peaks can be observed.

4.2.4 Some considerations and proposed improvements

We have compared four interpolation strategies for spectral envelopes which, due to their generality, could be useful in most cases. Yet there are several improvements which can be implemented. Foremost, if we were to be dealing with spectra having well-defined formants, optimal results could be obtained by using a method consisting of matching formants between the involved spectra and then performing a logarithmic interpolation of resonant frequencies, Q factors and amplitudes[66]. Formant information can be acquired in

several ways. It could be derived from a physical model, estimated through LPC analysis or it could also possibly be approximated from a power spectrum as proposed by Depalle[67].

Alternatively, for peaky envelopes, which can be estimated via LPC, line spectral pairs would seem to be an appropriate choice if we rely on findings from the speech processing community.

We also recall, from subsection 2.1.1, Caetano and Rodet's[22] paper on evolutionary spectral morphing. By the same token, we recall that in section 2.2 we mentioned Pfitzinger's[41] Dynamic Frequency Warping algorithm. This seems to be a very promising method as can be heard from the results posted on his webpage[50] which is linked here in the electronic version of this document. It's worth noting that, in the implementation of DFW, the preoccupations in terms of strategy are displaced to the tuning of the Dynamic Programming algorithm for finding correspondences between spectral peaks.

4.3 Warping along the frequency axis

As we have previously stated, within the musical context we often find sounds which have a harmonic or quasi-harmonic structure in the arrangement of their partials. For these sounds we have a frequency f_0 such that it is the maximum common divisor of all partial frequencies. Sounds with such characteristics are perceived as having a definite pitch, and it is important, throughout a morph between two such sounds, to retain the harmonic structure so that as one sound evolves toward the other, fractional interpolation factors don't yield inharmonic sounds.

The first and obvious comparison that we could perform between a pair of strategies is that of logarithmic interpolation vs. linear interpolation. Yet most times that we find mention of the actual method of interpolation of partial frequencies in the literature, logarithmic interpolation is preferred[4, 7, 8]. Since we found a quick perceptual test to be in agreement with the unanimous choice of logarithmic interpolation, we have decided to forgo this comparison[7].

In what regards f_0 interpolation, a more interesting area of exploration is that of keeping a harmonic structure throughout the interpolation while avoiding glissandi covering great

[7]We have, nevertheless, investigated linear interpolation and left it as a commented out section of code in the scripts. If uncommented, this section performs linear interpolation, in case the reader is interested in executing it.

distances between different f_0s, since the gliding becomes an overpowering characteristic of the sound. Thus the comparisons performed in this area head toward proposing a viable solution to spanning greater fundamental frequency differences through the shortest possible glissandi.

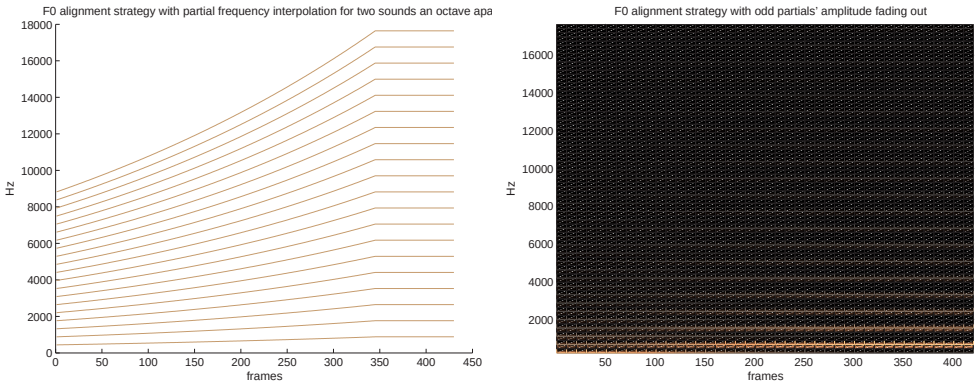
For every comparison, we chose two fundamental frequencies, f_{0_1} and f_{0_2} and performed their interpolation throughout a 5 second sawtooth-wave.

4.3.1 First strategy, one-to-one naive partial frequency interpolation

In order to illustrate the overpowering effect of glissandi, a choice of $f_{0_1} = 441Hz$ and $f_{0_2} = 882Hz$ was made; this is equivalent to an ascending octave. An sawtooth-wave with twenty partials was created in order to avoid aliasing toward f_{0_2} and the partial frequencies were logarithmically interpolated from the onset to $t = 4s$. An example of this strategy can be found on the project's webpage[9] and has been linked here in the electronic version.

4.3.2 Second strategy, closest harmonic structure partial amplitude interpolation

The same fundamental frequencies were interpolated but in this case, we took advantage of the common harmonic structure between the two sounds. Since the partials' frequencies of a sawtooth-wave at f_{0_2} are equivalent to the even partials' frequencies in an sawtooth-wave at f_{0_1}, f_{0_1}'s odd partials were faded out between $t = 1s$ and $t = 4s$. There is a very big difference with the prior strategy, since the proposed *octave fadings* produce no glissando throughout. Recalling the three possible types of audio morphing proposed by Slaney, Covell and Lassiter[2], this strategy yields a result which is most likely to be preferable particularly in the context of a dynamic morph since it seems easier to keep focused on the timbre throughout the interpolation than with the glissando. By the same token, the first strategy might be preferable for what Slaney *et al* refer to as stationary and cyclostationary morphs. The comparison of the first and second strategies can also be seen on figure 4.5. An example of this strategy can be found on the project's webpage[9] and has been linked here in the electronic version.



(a) f_0 naive warping over an octave (b) f_0 warping with *octave fading*.

Fig. 4.5 Two f_0 warping strategies, corresponding audio files can be found on the project's website. Partials obtained by performing interpolations directly on the parameter set. The representations differ since amplitude–pictured as brightness–only varies in the second strategy. 4.5(a) A frequency vs frames representation: frequency of partials gradually increases toward the target f_0, keeping a harmonic relation. 4.5(b)A frequency, frames and amplitude representation: The spectrogram shows partials fading out toward the end, where only even partials remain

4.3.3 Extending the previous strategy

For the next trial we have chosen $f_{0_2} = 3 \cdot f_{0_1}$ which is equivalent to an ascending just thirteenth or an octave plus a just fifth. Instead of a very broad glissando, we have again resorted to fading partials out throughout the event; in this case we fade out two out of every three partials.

An example of this interpolation can be found on the project's webpage[9] and has been linked here in the electronic version.

4.3.4 Third strategy, closest neighbor and closest harmonic structure partial amplitude interpolation

In order to illustrate this strategy, we have chosen $f_{0_1} = 441Hz$ and f_{0_2} to be $\frac{8}{5}f_{0_1}$ which is equivalent to an ascending minor sixth in just tuning. Following the same principle of fading partials, we have performed a glissando toward $\frac{4}{5}f_{0_1}$ coupled with fading odd partials. We have also performed a naive interpolation of f_{0_1} and f_{0_2}. The f_0 ratio from

the interpolation with the fading partials is equivalent to a descending major third in just tuning. The rationale behind this choice is to take the *shortest distance* approach in order to avoid extensive glissandi.

The reader is encouraged to listen and compare the results; both interpolations are available from the project's website[9]. They have been linked in the electronic version: the naive interpolation has been linked here and the fading-partials interpolation here.

Given any two f_0s, f_{0_1} and f_{0_2}, a good procedure for obtaining the shortest glissando along with phantom partials would be to follow an algorithm which seeks to minimize the interval of the glissando and at the same time retains as much as possible from the target harmonic structures. We have sketched out a routine to achieve this goal in algorithm 4.1 where we study several possible routes[8] to get from the lower f_0 to the higher f_0. We then store values for each one of these possible routes in an array and in the end we opt for the route c with the lesser glissando. Note that minor adjustments should be made at the end for slight deviation of partial frequencies from the underlying harmonic model due to their inharmonicity coefficients.

The possibility of combining partial fading with glissandi can prove useful for the fulfillment of certain musical constraints. One example of this could be the placement of a different constraint for choosing c, such that it yields a $cent2gliss$ value as close as possible to forcing a certain intervalic content on a morph; where the sought interval is in accordance with a harmonic or melodic context.

It would be advisable to exercise caution, avoiding too many of the partials to fade in or out during the process since we can easily approach results which are closer to cross-fading than to morphing. Thus it could also be of interest to place constraints on c in regards to the resulting $finalPartial2phantom$. Another consideration to take into account is that in performing this type of strategy we might impose changes on the spectral envelope as well as on even-to-odd ratio.

4.4 Morphing vibrato

As we have seen in chapter 3, a generalized form of vibrato can be understood to be comprised of the modulations of an event's frequency and amplitude, and the implicit

[8]There is a potentially redundant number of them, but it seems to be sound enough for a first approximation to the issue at hand.

Algorithm 4.1 is a way of interpolating between f_{0_1} and f_{0_2} getting the shortest glissando with *partial fading*. Several paths, c, between harmonic structures that in the case fading partials would have no more than $\frac{1}{b}$ of their partials in common are explored and the shortest glissando, is chosen to be $finalCent2gliss$, giving a final number of common partials $partial2phantom$.

if $\frac{f_{0_2}}{f_{0_1}} \geq 1$ **then**
$\frac{p}{q} \leftarrow \frac{f_{0_2}}{f_{0_1}}$;
else
$\frac{p}{q} \leftarrow \frac{f_{0_1}}{f_{0_2}}$;
end if

for $b = 2$ to $\lceil \frac{p}{q} \rceil$ **do**

$exponent_{[b-2]} \leftarrow log_b \left(\frac{p}{q}\right)$;

$partial2phantom_{[b-2]} \leftarrow b^{\lfloor exponent_{[b-2]}+0.5 \rfloor}$;

$ratio_{[b-2]} \leftarrow b^{exponent_{[b-2]}-\lfloor exponent_{[b-2]}+0.5 \rfloor}$;

$cents2gliss_{[b-2]} \leftarrow log_2 \left(ratio_{[b-2]}\right) \cdot 1200$;

end for

$c \leftarrow b-2$, *s.t.* $|cent2gliss_{[b-2]}| = min(|cent2gliss_{[b-2]}|)$;

$finalCents2gliss \leftarrow cent2gliss_{[c]}$;

$finalPartial2phantom \leftarrow partial2phantom_{[c]}$;

spectral envelope modulations which result from considering the amplitude modulation of each first-order partial separately. Modulations can be seen as a series of inharmonic low-frequency partials which can be extracted from a second order sinusoidal analysis.

We have sought to extrapolate the time-stretching technique proposed by Marchand and Raspaud[58] in the context of morphing and have found the results to be adequate. In order to do so, we have performed an interpolation of two vibratos extracted from the analysis of a violin tone and a saxophone tone. The steps that were followed can be grouped into three general stages: Analysis of interpoland modulations, establishment of temporal correspondence and re-synthesis of target modulations. The stages and the steps performed therein are described in the following pages.

4.4.1 Analysis of interpoland modulations

The first step is to define how we will measure modulations in a way that is somewhat independent of f_0 and amplitude. Take for example the case of frequency deviation: since a maximum deviation of 20 Hz is perceived as a mild vibrato for a note with $f_0 = 2000Hz$ but a wild vibrato for an event with $f_0 = 200Hz$, it would be desirable to have our measure of deviation in units that can be easily ported between sounds by virtue of being independent from f_0. The same principle applies to whatever choice we make for measuring amplitude modulations. A standard procedure to achieve this is to use a relative error, which instead of measuring the total frequency deviation Δf it measures the variation relative to the mean $e_{rel} = \frac{\Delta f}{f}$.

We have chosen a particular variant to measure frequency deviations in such a portable fashion: doing so in cents. This can be done in the following manner:

$$fm_{cents}[k] = log_2 \frac{f_{0inst}[k]}{f_{0avg}} \cdot 1200$$

where $f_{0inst}[k]$ is the value of the first partial's frequency at frame k, f_{0avg} is the mean value of f_0 across all the event's analysis frames and $fm_{cents}[k]$, for the K available frames is the frequency-independent representation of the modulated signal.

Amplitude modulation or the instant amplitude deviation from the amplitude envelope,

on the other hand, can be measured in decibels. Thus we have:

$$am_dB[k] = 20 \cdot log_{10} \frac{amp_{inst}[k]}{amp_{env}[k]}$$

where $amp_{inst}[k]$ is taken to be the amplitude at analysis frame k and $amp_{env}[k]$ is the value of the amplitude envelope at frame k.

During the analysis, stage, we perform a first-order sinusoidal analysis of our target sounds and then extract the modulation signals in cents and dB. Once we have the target modulation signals, we extrapolate them in preparation for a second-order sinusoidal analysis which we perform at the end.

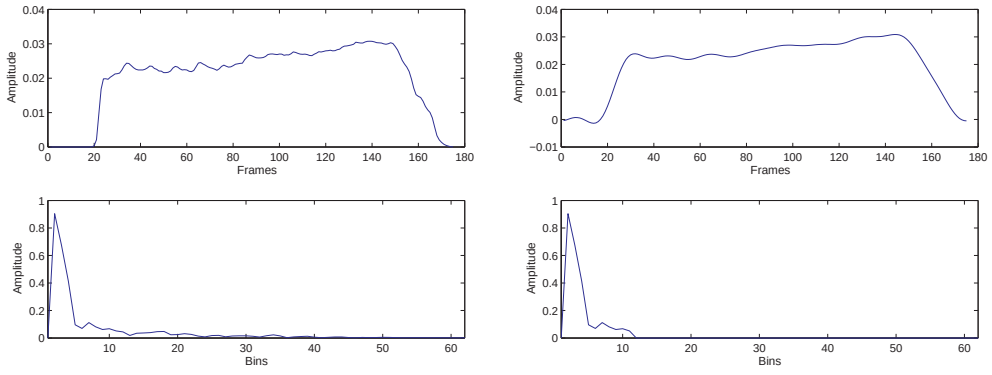
Vibrato extraction

In order to evaluate the proposed strategy on two realistic vibratos, we prepared for the interpolation by performing second order sinusoidal analysis of two notes played on a violin and a saxophone respectively. For the most part, we followed the procedure proposed by Marchand and Raspaud[58]. The analysis was performed on the harmonic track information which was obtained from a *pm2* analysis. The hop size for the harmonic analysis was 1024 samples and the audio files were originally sampled at 44100 Hz. Thus the resulting analysis sampling rate for the extracted first-level sinusoidal data was roughly 43 Hz, which allowed us to extract modulation information[9] up to 21.5 Hz. This upper bound effectively limited the information that we could obtain from a second order sinusoidal analysis to the sub-audio frequency range, which is precisely the range in which we are interested.

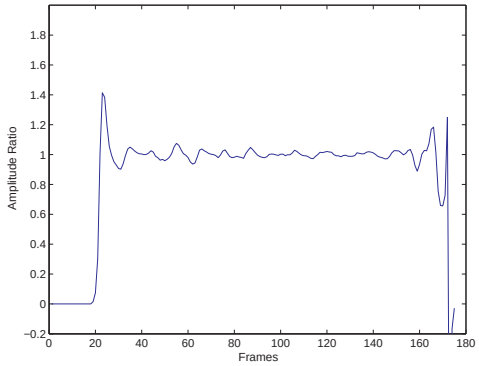
Frequency modulation was extracted only from the first partial, as a deviation given in cents, allowing the information to be independent of the notes' f_0.

For the analysis of amplitude modulations, we needed to define a method for estimating the amplitude envelope. We opted for performing a frequency-domain low-pass filtering of the partial's amplitude evolution. Filtering in the frequency-domain, yields no group delay and eliminates all energy at frequencies above the cutoff frequency. We chose a cutoff frequency of 6 Hz and obtained adequate results. Yet as with many analysis methods, some tuning should be performed on the cutoff frequency for each realization in order to obtain a smooth amplitude envelope.

[9] ie, second order partials.



(a) Amplitude envelope (b) Low-passed amplitude envelope



(c) Amplitude ratio

Fig. 4.6 Figure 4.6(a) shows the raw amplitude values per frame and their Fourier transform, excluding dc offset. Figure 4.6(b) shows an ideal low pass filtering of the amplitude envelope, obtained by zeroing out higher frequencies in the spectral domain. Figure 4.6(c) shows the ratio between these two versions of the amplitude envelope.

For an implicit spectral envelope modulation, amplitude modulation analysis should be performed on each partial; in this reductionist trial, we only used a single amplitude modulator obtained from the first partial.

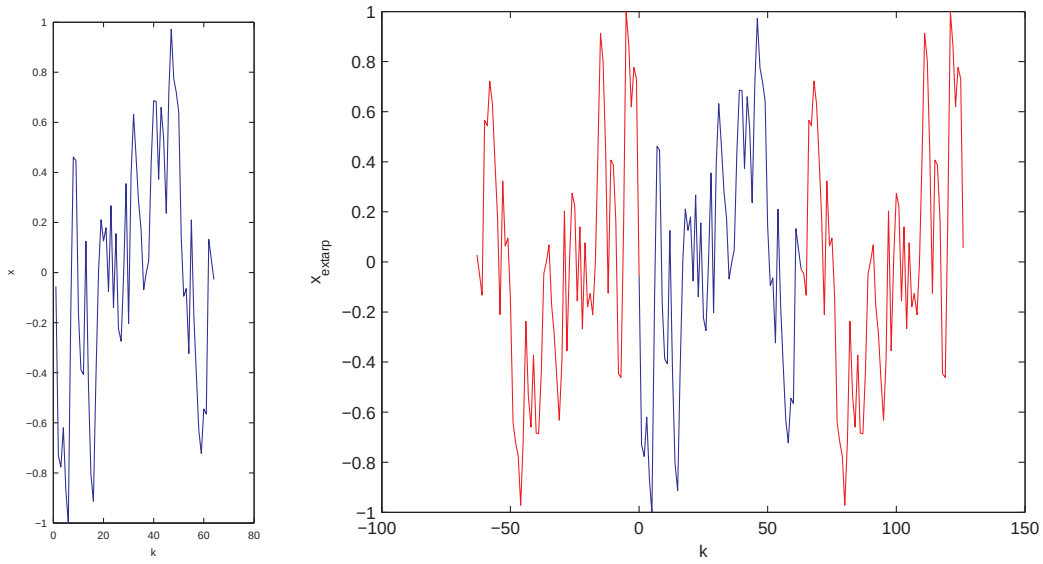
Signal extrapolation

Following the procedure for vibrato preservation during time-stretching proposed by Marchand and Raspaud[58], the frequency and magnitude-independent modulations were extrapolated, or extended, so as to increase the accuracy of the analysis for the first frames of our signal. In this case, differing from the method that they propose, a standard method

for extrapolation was used; by placing a symmetric inversion of the signals at their start and end-points:

$$x_{extrap}[k] = \begin{cases} -x[-k] & \text{if } -K < k < 0 \\ x[k] & \text{if } 0 \leq k < K \\ -x[K-k] & \text{if } K \leq k < 2K \end{cases}$$

where $x[k]$ is defined for $k \in \mathbb{Z}$ such that $0 \leq k < K$ and x_{extrap} is defined for $k \in \mathbb{Z}$ such that $-K < k < 2K$. The result of extending a signal with this extrapolation strategy can be seen on figure 4.7[10].



(a) Original signal. (b) Extrapolated signal

Fig. 4.7 Extrapolation increases the accuracy of the $STFT$ at the extrapolated region by diminishing the impact of discontinuities. Here we have an illustration of the extrapolation procedure described above.

Second order sinusoidal analysis

After the signals were extrapolated, a partial tracking analysis was performed on the frequency and amplitude information obtained from the first-order sinusoidal analysis. The

[10]Although through this type of extrapolation we eliminate discontinuities at the boundaries of our signal, we do not necessarily eliminate discontinuities in its first derivative, thus the spectrum cannot be steeper than 12 dB per octave.

analysis was performed via *pm*2. Attention was paid both to the upper and lower frequency bounds of the second-order partial analysis.

The upper frequency bound is important since the goal of the second-order partial analysis is to model the sub-audio rate partials that can represent modulations[11]. We satisfied this condition by choosing the first-order partial analysis frame rate to be 43 Hz; effectively placing an upper bound of 21.5 Hz on the frequency content that could be extracted by the second-order partial analysis.

In regards to the lower bound, vibrato can contain very low frequencies which should not be lost during the second-order partial analysis; therefore, the $STFT$ parameters for second-order partial analysis should include a large window size. But with an already low sample rate, as was the case with our first-order analysis frame-rate, a large window implies a very poor time resolution; we are faced with the ubiquitous time-frequency resolution constraint. Fortunately, the loss in time resolution can be palliated by the use of a small hop size. Weighting trade-offs, we chose to use a 64 point window[12] with a hop-size of 4 samples.

4.4.2 Time-warping, re-sampling and matching of modulation information

Once we had extracted a second-order partial model of the modulations of each target sound, we were faced with the need to warp these models. The need stems from the discrepancy between both sounds' durations. Thus, each morph requires warping target sounds to an interpolated duration. Consequently, the second-order partial models must also be warped. The approach taken was to over-sample these models until a common number of second-order frames was attained. Oversampling the second-order models' amplitude and frequency information required a different approach from oversampling their phase information.

Time-warping

Once the duration of the interpolated sound was decided, and for each target sound a time-warping factor was found. Each frame's *timestamp* was modified according to the

[11] Vibrato is, by definition, concerned only with the modulations that occur below the frequency threshold of audibility.

[12] More than two periods of frequencies higher than 1.34 Hz.

time-warping factor, effectively changing the frame rate so as to attain the target duration without modifying the amount of frames. Differently from frequency and amplitude information this time-warping required some sort of compensation to be performed on phase information; each partial's phase vectors were unwrapped and then multiplied by the time-warping value in order to preserve phase evolution.

It is important to remember that in this simplistic case we had a single time-warping factor, yet for most sounds, time-warping would imply at least two different time-warping coefficients in order to be able to align the attack independently from the overall duration.

Oversampling

We then oversampled second-order partial information from both target sounds to a least common multiple of each one of the resulting frame rates. In their article, Marchand and Raspaud propose oversampling by means of convolving with a *sinc* function. We oversampled our data by a process which is equivalent to this convolution: by adding zeros between known samples[13]; then performing a FT on the resulting signal; eliminating all frequencies which are higher than the original Nyquist frequency; scaling all magnitudes by the upsampling factor, and then performing an IFT on the rescaled low-pass filtered information.

The procedure for the phase information of each partial was different than the one used for magnitude and frequency. The previous procedure, actually retains the frequency information of the signal up to the original Nyquist frequency and does not add any frequency content above it, eliminating contributions from the frequency range between the original Nyquist frequency and the higher Nyquist frequency resulting from oversampling. Since linear or quasi-linear functions can be generally seen to have a relatively flat spectrum and a partial's phase tends to present a quasi-linear evolution, it follows that phase information is not well represented as a sum of lower-frequency sinusoids. Thus the previously described oversampling procedure generally yields a fair amount of ripples. If instead of oversampling by the frequency-domain equivalent of a *sinc* convolution, we perform a cubic interpolation for the known phase values at the times given by the new sampling intervals, we obtain a phase evolution which is much closer to the original in terms of being quasi-linear.

[13]This is Matlabs standard upsample() routine.

4.4.3 Interpolation and target modulations

What remained to be done was the interpolation of the second-order partial models, re-synthesizing modulations from the resulting model and applying these modulations to the otherwise morphed first-order partial set. It is important to note that interpolation of the second-order partial sets required finding a strategy for matching partials between models.

Interpolation

Once our second-order sinusoidal analysis had a common duration and a common frame-rate, partial correspondence was sought between each one of the interpolands. The first strategy was to establish a correspondence based on the proximity of each second-order partials' frequency. This was seen to yield poor results and so we tried establishing correspondence based on partials' magnitudes; the latter strategy yielded far superior results. Since partial sets differed in regards to number of partials, the unmatched partials of the larger set were matched with phantom partials. Their frequencies and phases were equalized to the matching set's frequencies and phases. The phantom partial's magnitude, on the other hand, was set to be zero.

Subsequently, corresponding partials' frames were interpolated. Having observed logarithmic interpolation to be best suited for frequency and magnitude, it was seen as an obvious choice for these two partial characteristics, while phase interpolations were performed linearly on the unwrapped phases.

re-synthesis of modulator

After performing interpolation, we performed a re-synthesis of the frequency and amplitude-independent modulator signals via *pm*2. It was observed that performing such a re-synthesis through the oversampled partial information produced unexpected results, containing higher frequencies than the original modulators, this is perhaps a phase evolution problem or a bug in our sample-rate changing routines but we have not yet discovered it. The solution was found to be to downsample back to a lower sampling rate, in the vicinity of either one of the original samplerates, before re-synthesis.

Different resulting modulator signals are shown in figures 4.8 and 4.9.

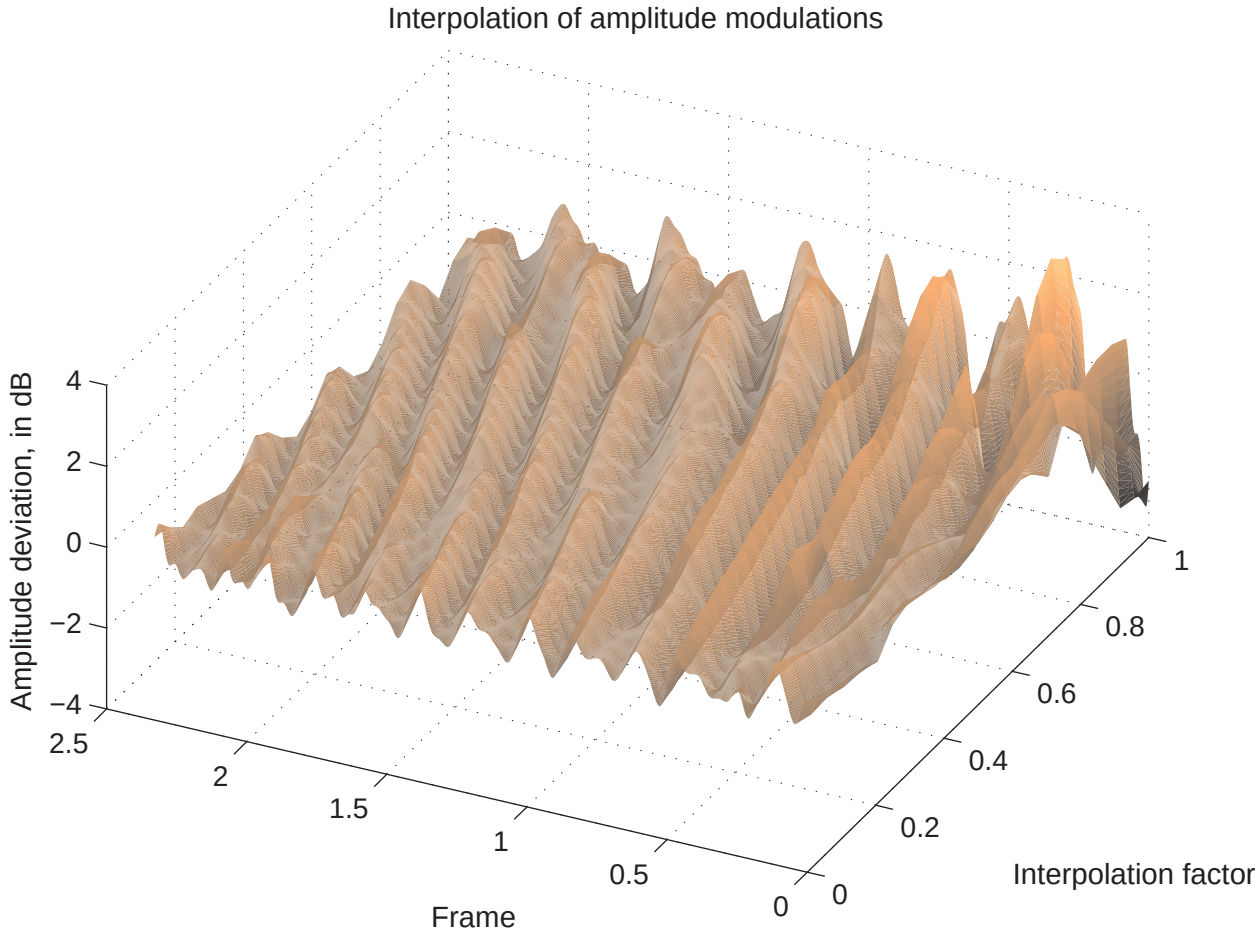


Fig. 4.8 Amplitude modulation interpolation, the extracted AM is represented in terms of dB, so as to be useful for modulating any partial despite its magnitude .

Modulation

The last step was to impose the resulting modulators on a sawtooth-wave for perceptual validation[14]. This was achieved by importing the modulator tracks into Matlab and using them to modulate an arbitrary set of partials which were then re-synthesized through *pm*2– similarly to those of other prior tests. The results can be found on the author's webpage[9] and have been linked here in the electronic version.

[14]During this particular realization, the saxophone vibrato, was perceived to be somewhat faint and was thus exaggerated by multiplying the magnitude of both it's AM and FM parts by a factor of 4.

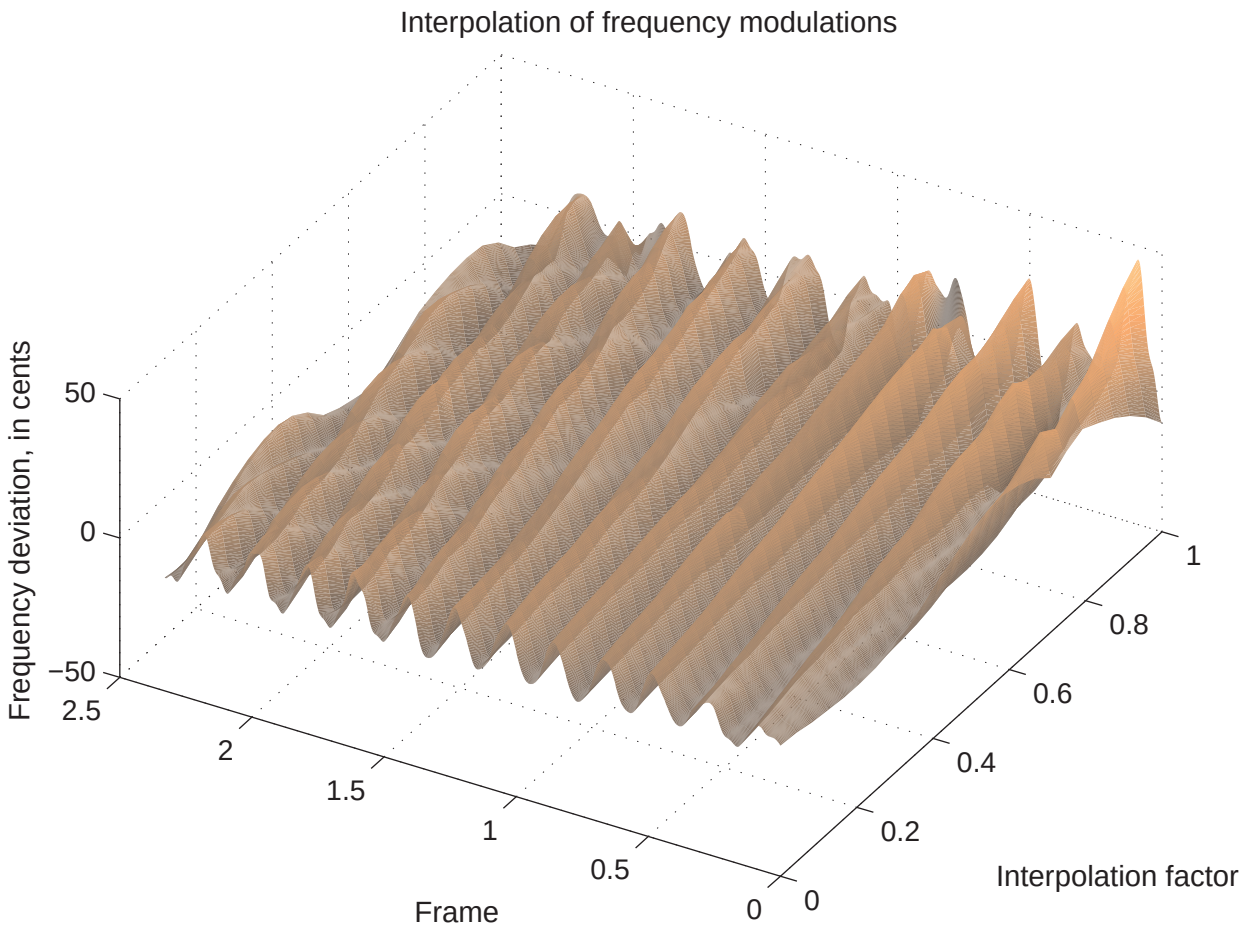


Fig. 4.9 Frequency modulation interpolation, the extracted FM is represented in terms of deviation in cents, so as to be useful for modulating any partial despite its frequency.

4.5 Inharmonicity

As we have seen in chapter 3, inharmonicity is a deviation from purely harmonic relations. In physically produced sounds, inharmonicity tends to occur due to specific characteristics of the resonator; characteristics such as string geometry for an oscillating string or bore irregularities in a wind instrument.

The way that we have defined inharmonicity allows us to consider a single coefficient that measures the overall inharmonicity of a given sound or to characterize the relations between partials' frequencies by means of a vector of inharmonicity coefficients corresponding to each partial above the first.

The proposed inharmonicity vector allows us to interpolate between harmonic or quasi-harmonic series. Moreover, such a definition grants us the necessary tools to interpolate completely inharmonic sounds with a harmonic series. Since inharmonicity relies on the assumption that at least one of the interpolands has an underlying harmonic structure, we have not contemplated the interpolation of two completely inharmonic signals, where we should be faced with a partial matching problem that doesn't require a harmonic structure for it's solution.

We have written a few scripts for testing different possibilities of interpolation of inharmonicity coefficients on harmonic, quasi-harmonic and inharmonic partial series. For harmonic and quasi-harmonic partial series of n components, all of the first n components are present and have an -6 dB per-octave roll off; thus, we could talk about sawtooth-waves and *quasi sawtooth-waves*.

When dealing with inharmonic series, we have contemplated two cases. In order to deal with the two categories that we have envisioned for fully inharmonic sounds let us define a full-partial, inharmonic set as one for which each partial could be matched with an underlying harmonic structure. By the same token, let us define a sparse set of inharmonic partials as one for which there are fewer partials than those which would be contained in an underlying harmonic structure spanning the same frequency space. For the first case, or the full-partial inharmonic sound, all partials present in our tests have equal magnitudes. In the second case, or sparse set of inharmonic partials, we used varying magnitudes as well as phantom partials to match with the underlying harmonic structure.

We have first set out to compare linear and logarithmic interpolation. Furthermore, we then detail a method for interpolating sparse inharmonic sets of partials, such as those that we could extract from a bell sound, with full harmonic or quasi-harmonic series.

4.5.1 Interpolation of coefficients

The linear interpolation of inharmonicity values for any two given series of partials is a trivial routine. Yet if we want to compare it with a logarithmic interpolation we are then faced with a problem: inharmonicity coefficients include the set of numbers $[-1, 0]$ for which a logarithmic function is not defined.

One solution is to offset inharmonicity values to a range such as $[1, 3]$, interpolate them and then remove the offset. This is not an optimal solution, since interpolation curves will

vary depending on how large of an offset we use and our original intention was to have the resulting frequencies behave as if they had been interpolated in a logarithmic fashion.

Thus, another solution is to obtain target frequency values according to each partials' inharmonicity coefficients and then to interpolate partial frequencies as opposed to the actual inharmonicity coefficients. This is, in a way, *cheating* since we are not really interpolating the inharmonicity coefficients but the actual frequencies that they yield. Nevertheless, it should give us accurate sound examples by which we can evaluate logarithmic interpolation.

One comparison trial was realized for a full-partial, inharmonic set to harmonic series interpolation and another one was carried out for an interpolation between two quasi-harmonic series. Finding a sound with a full-partial, equal-magnitude, inharmonic set is unlikely, yet this realization was used to test what could be considered an extreme case; thus helping to perform a qualitative evaluation of the method via the resulting sound file.

For each comparison, a random vector of inharmonicity coefficients was created. The complete $[-1, 1]$ range was used for inharmonic sounds. Whereas an arbitrary constraint was placed on the inharmonicity coefficients in order to consider the sound quasi-harmonic, limiting them to the $[-0.25, 0.25]$ range.

Although logarithmic interpolation did sound more natural to our ears, we found the difference between linear and logarithmic interpolation to be rather small.

A closer look at the behavior of partials during interpolation is more eloquent in regards to the almost negligible effect of the choice of interpolation type. Let us remember that the limit case is when we interpolate the inharmonicity coefficient of -1 from the source sound's partial with an inharmonicity coefficient of 1 for the target partial. In this limit case, the largest differences between linear and logarithmic interpolation occur around an interpolation coefficient of 0.5[15]. For the second partial[16], interpolating coefficients 1 and -1 would yield a maximum difference of approximately 55 cents between the two types of interpolation. As the harmonic number increases, this difference decreases, as can be seen in figure 4.10. Already at partial number four, we can see that the worst-case difference is a

[15] Actually the point of maximum difference is where $\frac{d}{d\alpha}$ of $h - 0.5 * (\frac{h+0.5}{h-0.5})^{\alpha} = 1$, which for a second partial is roughly $\alpha = 0.522$, for a third $\alpha = 0.511$ and it asymptotically approaches $\alpha = 0.5$ as h increases.

[16] Although the first partial may have an inharmonicity coefficient, or even be absent, we simplify by stating that f_0 will be the average frequency of our first partial.

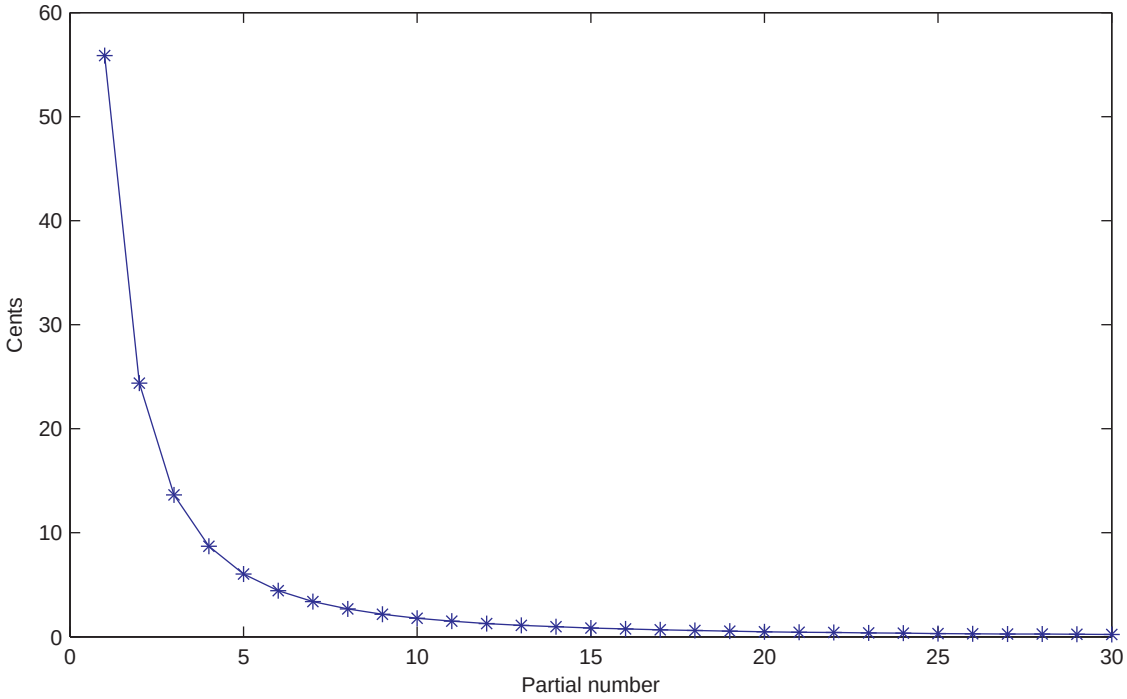


Fig. 4.10 The worst-case relative frequency difference per partial for the interpolation coefficient which gives the largest discrepancy between linear and logarithmic interpolation of inharmonicity coefficients. Differences are given as cents, dependent on the partial number and the worst case scenario corresponds to the interpolation of coefficients 1 and -1. As can be observed, the difference above the third partial becomes negligible.

negligible 13 cents. These differences can be found by evaluating the following expression:

$$\Delta Cents_h = log_2(\frac{h - 0.5 + \alpha_{\Delta_{max}}}{h - 0.5 * (\frac{h+0.5}{h-0.5})^{\alpha_{\Delta_{max}}}}) \cdot 1200 \tag{4.1}$$

Where h is the harmonic number, $\Delta Cents_h$ is the difference in cents between the logarithmic and linear interpolations for a given harmonic and $\alpha_{\Delta_{max}}$ is the interpolation coefficient at which the maximum difference between interpolation strategies is found.

The resulting sound examples can be found on the project's website[9] and the files are herein linked for the reader's convenient access.

- linear interpolation between two quasi-harmonic sounds,
- logarithmic interpolation between two quasi-harmonic sounds,

- linear interpolation between an inharmonic sound and a harmonic sound,
- logarithmic interpolation between an inharmonic sound and a harmonic sound

4.5.2 Matching partials between inharmonic sounds and underlying harmonic structures

The partial-matching strategy we chose dictates that partials from inharmonic sounds are always matched to their closest neighbour from the underlying harmonic structure. For any given partial in an inharmonic sound, correspondence to the underlying harmonic structure only results ambiguous in the limit case, when the partial's frequency falls exactly half-way between two harmonics and it could matched with partial h with inharmonicity 1 as well as with partial $h + 1$ with an inharmonicity coefficient of -1. In such a limit case, other considerations might decide in favour of one choice or another; e.g. the trends of surrounding partials. Conversely, components from the underlying harmonic structure that find no correspondence in the inharmonic sound's partials can then be paired with a zero magnitude version of themselves.[17]

Both logarithmic and linear interpolations between sparse inharmonic series of variable partial magnitudes and harmonic series can be found at the project's website[9]. The files have also been linked in the electronic version of the document for the reader's convenience:

Interpolations between an inharmonic sound with phantom partials and a harmonic sound:

- linear,
- logarithmic

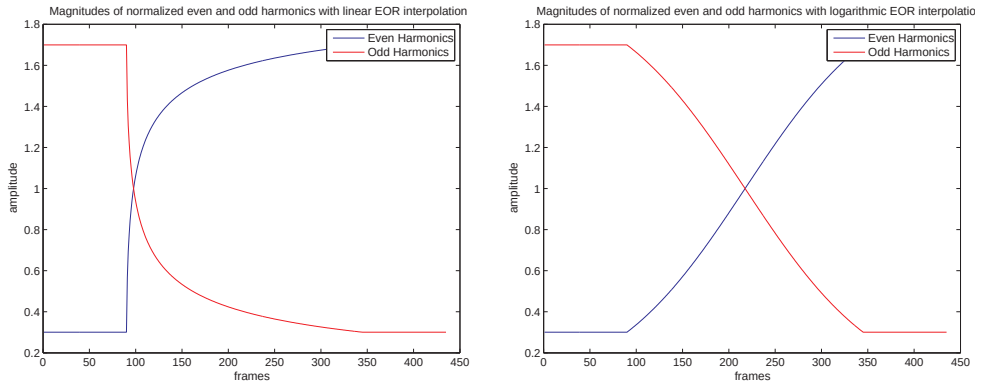
4.6 Even to odd partial energy ratio

The interpolation of the even to odd ratios is relatively straight-forward. By definition, EOR deals with energies, which is a sum of squared magnitudes and can be seen as a

[17] As in other partial matching strategies which rely on phantom partials, it is important to note that due to simultaneous masking, phantom partials become evident only toward the end of their magnitude interpolation. We thus propose researching the factors that might influence the choice of a lower threshold for phantom partial magnitude interpolation. We believe that phantom partials should start at a magnitude slightly inferior to whatever masking threshold is in play for their frequency within a given context. Yet this is subject matter for future study.

squared gain which is applied to the even set of partials[18]. From chapter 2, we recall the three types of morphs mentioned by Slaney, Covell and Lassiter[2]: stationary, cyclo-stationary and dynamic. During a stationary or cyclo-stationary morph we can interpolate the ratio of the total energy contained in each of the subsets of partials: even or odd. But for dynamic morphs we can only interpolate the ratio of the power of both sets; an interpolation which takes place on a frame-by-frame basis. Thus, for the EOR interpolation during dynamic morphs, a single EOR measure is insufficient; we must have a measure of EOR per frame.

Like most of the previous features, the interpolation can be performed linearly or logarithmically. We have carried out tests with both sorts of interpolations and found logarithmic interpolation to be much smoother. This comes as no surprise since we have already corroborated logarithmic interpolation of amplitudes to be much smoother than their linear interpolation.



(a) Linear interpolation of EOR (b) Logarithmic interpolation of EOR

Fig. 4.11 A comparison of the behaviour of amplitude given by linear and logarithmic EOR interpolation strategies.

A brief scrutiny of the differences between logarithmic and linear interpolation of EOR will serve to caution the reader against the use of linear interpolation for this feature. As previously mentioned, let us simply consider the EOR as a square of the gain applied to even amplitudes keeping in mind that this is a heuristic, since that which concerns us is the interpolation of energy, and not amplitude. With this simplification, it becomes evident that the logarithmic interpolation of squared values is equivalent to the square of

[18]This dismisses the importance of the overall final gain of a sound but results in no loss of generality.

the interpolation of the same two values.

$$g_1^2 \left(\frac{g_2^2}{g_1^2}\right)^{\alpha} = \left(g_1 \left(\frac{g_2}{g_1}\right)^{\alpha}\right)^2 \tag{4.2}$$

For linear interpolation of squared quantities, however, this identity does not hold.

$$g_1^2 (1-\alpha) + g_2^2 \alpha \neq (g_1 (1-\alpha) + g_2 \alpha)^2 \tag{4.3}$$

With the interpolation of squared gain values, the divergence in the trends of the gain is more pronounced than it would be in the case of logarithmic vs linear interpolation of the gain itself. This divergence can be seen in figure 4.11 and is bound to produce noticeable discontinuities when EOR is linearly interpolated.

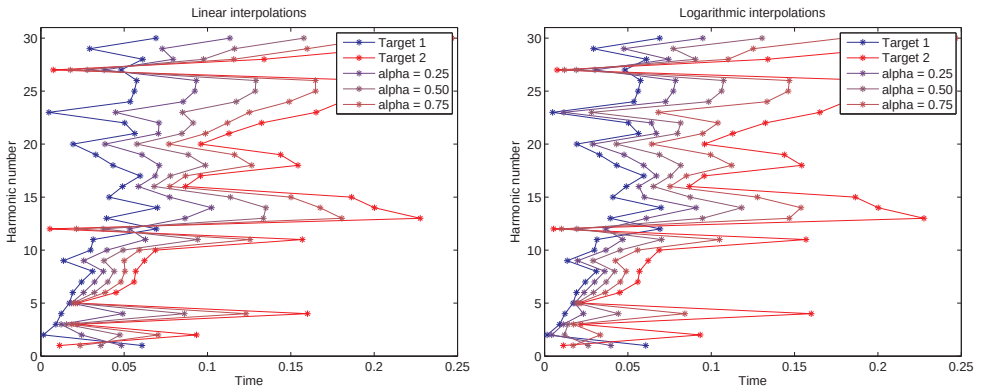
Both logarithmic and linear interpolations of EOR can be found at the project's website[9]. The files have also been linked in the electronic version of the document for the reader's convenience:

- linear interpolation of EOR values ranging from $\frac{1}{32}$ to 32,
- logarithmic interpolation of EOR values ranging from $\frac{1}{32}$ to 32,

4.7 Partial attack times

We recall from chapter 3 that the distribution of energy throughout the attack is an important feature. We also recall that it can be represented as a vector of times from the events t_0 to each partial's peak amplitude. Since the attack happens only once during a single-event sound, this feature can only be morphed in stationary or cyclo-stationary morphs.

We have written a script to compare the linear and logarithmic interpolations of this feature. The attack times for each of the partials were determined with a random generator, and some of the times were sorted in ascending order corresponding to partial numbers. Partial attack times ranged from 0 to 0.075 seconds during the first sound and from 0 to 0.25 for the second. Partials were chosen to be sorted in ascending time order according to a Bernoulli process; in other words, the equivalent of a coin toss was performed for each partial to decide if it would be part of the lot to be sorted or not. The target partial peak times are plotted on figure 4.12 as are the peak time interpolations.



(a) Linear partial attack time interpolations

(b) Logarithmic partial attack time interpolations

Fig. 4.12 A comparison of partial attack times for different partials given by linear and logarithmic interpolation strategies.

Both logarithmic and linear interpolations of partial onset time difference can be found at the project's website[9]. The files have also been linked in the electronic version of the document for the reader's convenience:

- linear interpolation of partial attack times for two target sounds,
- logarithmic interpolation of partial attack times for two target sounds,

4.8 Deterministic vs stochastic energy ratio

Once all other features have been interpolated for both deterministic and stochastic components, we will be able to synthesize them. After doing so, the last step required to effect the morph is to mix the two components. By controlling the energy ratio for the deterministic and stochastic components, we are indirectly controlling the gain that should be given to each one of the components during their mix. Since many sounds of the type that we have defined as *musical sounds* have a predominant deterministic part, we then propose that a good rule of thumb should be to adjust the stochastic component's gain in order to achieve the desired energy ratio[19].

We have carried out both linear and logarithmic cyclo-stationary interpolations so that the user may compare them. The interpolations were performed between two given broad-

[19]In dynamic morphs, this would actually alter the stochastic part's amplitude envelope.

band components with an arbitrary amplitude envelope; the two target ratios were 1 and 4. We have two reasons for arguing logarithmic interpolation to be preferable to its linear counterpart. Firstly, amplitude interpolation is generally perceived as smoother when it's carried out logarithmically than when it is done linearly. Secondly, as we have seen in the section on even to odd energy ratio and corresponding figure 4.11, logarithmic interpolation of energy equates to logarithmic interpolation of gain, whereas there is no such correspondence between linear interpolation of energy and gain.

We then performed a dynamic interpolation using the same two components and target ratios. Given previous results, the chosen strategy for the dynamic morph was logarithmic. Additionally, the interpolation was effected in both possible directions; form source to target and from target to source. We note that while cyclo-stationary and stationary morphs only require the interpolation of a single value throughout the whole morph, dynamic morphs call for the interpolation of the ratio[20] on a per-frame basis. However, since this is a time-variant ratio, it constitutes an approximation of a power ratio rather than the actual energy ratio.

It is worth pointing out an important difference between this *in vitro* interpolation test[21] and the interpolation of this feature during the cyclo-stationary morphing of two real-world sounds: in the latter, all other features, would have been interpolated and thus the two components to be mixed should be different for each different interpolation coefficient.

The reader may find all involved sounds on the projects webpage[9]. As with all other examples, files have been linked in the electronic version for the reader's convenience:

- deterministic component,
- stochastic component,
- linear cyclo-stationary interpolation,
- logarithmic cyclo-stationary interpolation,
- logarithmic dynamic interpolation source to target and
- logarithmic dynamic interpolation target to source

[20] And thus, indirectly, the stochastic part gain.
[21] With a static behaviour for all other features.

We have seen that during dynamic interpolation we are actually dealing with power and modifying the stochastic component's amplitude envelope. Thus along these lines we could also consider the possibility of deriving the stochastic component's amplitude envelope from the energy ratio between both deterministic and stochastic components. In order to do this, we need to extract the power ratio per analysis frame, up-sample the sequence of power ratios to audio-rate and take the square root of the resulting sequence to be the gain parameter of the stochastic component.

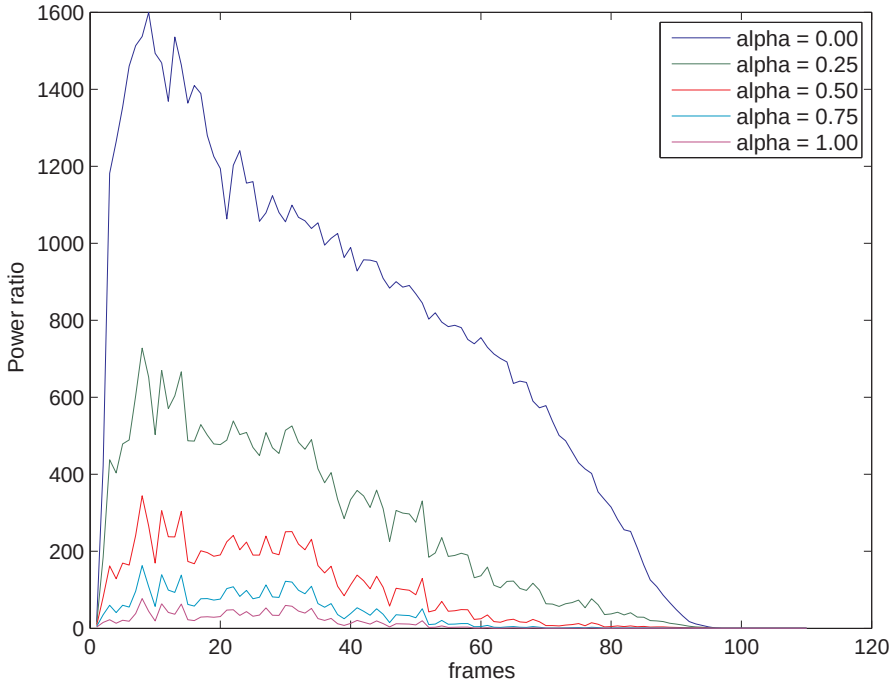


Fig. 4.13 Deterministic vs stochastic component power ratios

Extraction is done by obtaining the energy ratio at each analysis frame. We have extracted two such ratios from real world sounds, one lute tone and one english horn note and depict them along with their interpolations in figure 4.13. Having obtained power ratio sequences and their interpolations, each ratio represents the squared gain at the center of each analysis frame. Thus we up-sampled power ratios with a logarithmic strategy. Finally, in order to convert them into usable gains, we obtained the square root of each sample in the resulting sequence. The resulting series of gains is an amplitude envelope which can be applied to a constant-power stochastic component for which spectral characteristics would

generally have been interpolated[22]. The reader may find an example of this strategy on the project's website[9]. Additionally, the file has been linked here in the electronic version.

[22]Although in this trial we are simply using white noise

Chapter 5

Experimental Observations and Results for real-world sounds

The current chapter presents some simple morphings which are solely aimed at assembling the morphing framework, as well as validating proposed strategies. Morphing real-world sounds revealed challenges that we had not anticipated during the interpolation of synthetic sounds. In each case, we have at the minimum suggested a plausible solution as a future exploration. In some cases we have attempted to solve these difficulties.

Two morphs were performed between a pair of notes from a clarinet. Despite their many similarities, these two notes were different enough to allow the application and validation of previously explored concepts. Throughout the chapter, we document both the procedure that we employed in each case and the difficulties we encountered while effecting the morphs.

The arrangement of the chapter fully reflects the structure of the procedures that we followed. These procedures were mainly determined by the choice of a representational model and the selection of descriptors. The deterministic-plus-stochastic additive model forced upon us an initial component separation. The subsequent analysis, interpolation and resynthesis were almost all performed separately for each component and some descriptors had to be obtained and re-synthesized in a particular order. On the other hand, the differences between cyclo-stationary[1] or dynamic morphs called for slightly different procedures to be followed.

Thus, we first explain the procedures followed for the extraction of features obtained

[1]Or stationary morphing, which is procedurally equivalent to cyclo-stationary morphing.

from the complete musical sound object, then for the component separation, followed by the extraction of descriptors that can only be obtained after component separation has been performed. We then recount the steps taken to achieve a cyclo-stationary series of morphs and conclude by reporting on the realization of a dynamic morph.

The pitches of the two given musical sound objects were a B_3 and an $F\sharp_4$ at roughly 250 Hz and 380 Hz respectively. The dynamic level for the B_3 being *mezzo-forte* and *piano* for the $F\sharp_4$; both notes presented a noticeable vibrato. The vibrato of the first note was stronger at the start while the vibrato of the second note was more pronounced toward the end of the sound. The recordings were obtained from the RWC music database available at the SPCL. As with the previous chapter, corresponding sound files can also be found on the project's website[9] so as to provide an *audible illustration* of the ideas discussed herein; the reader is encouraged to refer to these samples. The B_3 has been linked here and the $F\sharp_4$ has been linked here in the electronic version.

5.1 Analysis

The first step taken was to fit a global amplitude envelope[2] to each sound object. The subsequent step was to separate each one of the notes into its stochastic and deterministic components. Afterwards, analysis and parametric extraction of both the deterministic and stochastic components were carried out, bringing us one step closer to the morphing of each one of the components.

5.1.1 An envelope fit for warping

The traditional model for amplitude envelopes is the ADSR, which finds its roots more in parametric synthesis than it does in analysis. Peeters[60] has rejected this model as a descriptor and after trying to employ it, we too find it ill-suited for our purposes. Morphing musical sound objects requires us to perform dynamic time-warping in order to match certain key moments. The definition of these key moments is a determining factor for designing an amplitude envelope model. Taking this into consideration, we have defined a collection of key-points that yields an alternative amplitude envelope model. This sort of amplitude envelope approximation can be seen in figure 5.1.

[2]The amplitude envelope of the whole musical sound object, as opposed to an amplitude envelope for one of the note's partials or the amplitude of either the stochastic or deterministic component.

The temporal features or key-points that we have chosen are the pre-attack, the attack, the stable part, the release and the post-release. The pre-attack and post-release sections contain ancillary noise such as the preparation of the attack or the breath release after a note. Due to their noisy nature, these two sections are more present in the stochastic component than the deterministic one. The attack and release sections are transitive stages with a heavy amount of spectral flux. It is in these sections that the spectral centroid change mentioned in chapter 3 takes place[3]. The remaining section, the stable section, is the quasi-stationary section characteristic of many musical sound objects; it is in this section, for example, that we may find a significant extended vibrato[4]. In regards to the amplitude envelope that stems from this choice, all sections except the stable stage are represented by start and end times as well as start and end amplitudes; in the case of the stable section, because of it's potential duration and changes[5] it can be modeled by a line segment approximation.

On envelope approximation methods

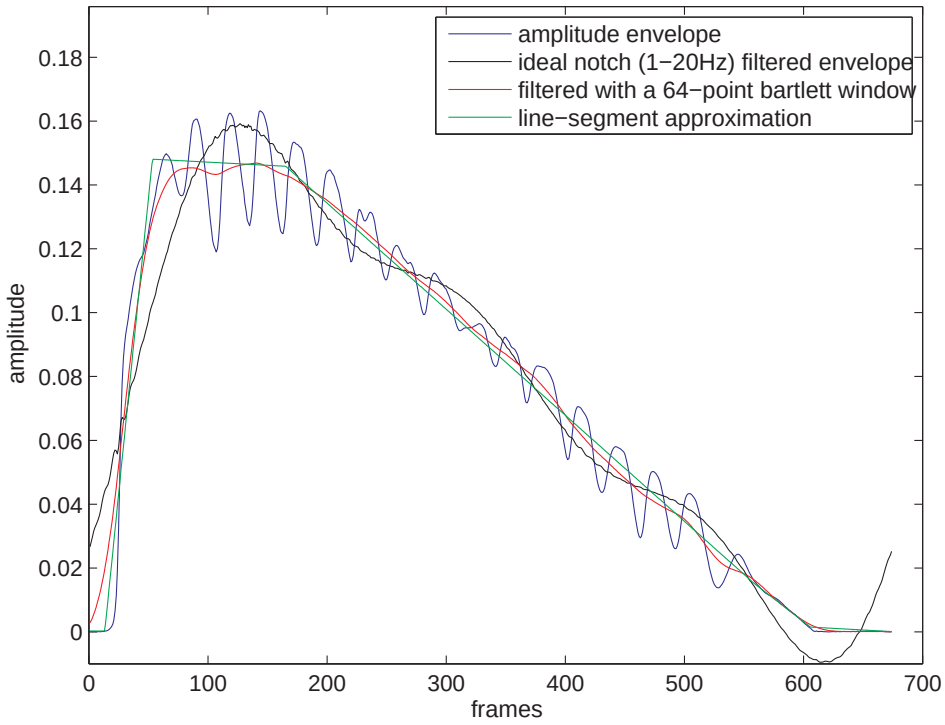
We employed an ideal notch filter for our first attempt at amplitude envelope extraction; we performed filtering in the frequency domain, completely removing all frequency content that spanned the defined notch. Separating all frequencies that are relevant for amplitude modulations was the motivation for the choice of this type of filter. We were aiming for obtaining an envelope approximation and amplitude modulation information in a single step. The reader may see in figure 5.1 that the resulting amplitude envelope is a very poor approximation, presenting an extremely slow attack and considerable oscillations. We tried tuning the filter with different upper and lower bounds for the ideal notch and were unable to obtain good results. The notch filtering performed in figure 5.1 effectively eliminates frequencies between 1 and 20 Hz.

We then tried a classic smoothing procedure: a moving average filtering. In this case, the envelope approximation behaved much better than ideal-notch filtering, particularly for the purpose of extended vibrato extraction. However, it also caused some important distortions by augmenting attack and release times.

[3]We remind the reader that this movement in the spectral centroid during attack and release has been proven to be perceptually meaningful by Grey[10, 11]

[4]Comprised of frequency and amplitude modulations.

[5]Eg, for many instruments, it's during this section that we find a continual energy input from the player.



(a) Amplitude envelope and approximations of the $F\sharp_4$ clarinet tone.

Fig. 5.1 A comparison of amplitude envelope approximation for the $F\sharp_4$ clarinet tone by means of line segments; weighted moving-average filter performed with a Bartlett window of 64 points, and an ideal notch filter–excluding all frequencies which are useful for modulation estimation, ie 1 to 20 Hz.

We thus approximated the envelope by manually chosen line-segments. This strategy proved to be unwieldy for large amounts of data, yet was found to be justifiable given that we would be working with only two musical sound objects. The advantage of this method was that it provided us with a finer degree of control. In order to manually approximate the envelope, we relied on the data as much as we relied on attentive listening. The section with the fastest rate of change in amplitude was chosen as the point of attack. We started at the first few frames, where blowing could be heard, and we ended at a point where heavy spectral fluctuations were finished[6]. Finding the release portion also relied heavily on attentive listening, particularly for the $F\sharp_4$ tone, since it decreases in amplitude throughout the duration of the note, as can be seen in figure 5.1. Thus, for finding the

[6]In other words, a point which, if taken as a starting point to play the musical sound object, would yield a sound which was perceived to have a stable timbre

period of release, we relied on several cues: steeper decrease in amplitude, increased spectral flux, and most importantly the absence of breath noise and a lack of modulations. The last two cues are relevant for wind instrument[7] sounds, signaling a halt in the excitation of the instrument. Once these two sections are identified, pre-attack, stable and post-attack stages can be picked by process of elimination.

5.1.2 Component separation

In preparation for the separation of stochastic and deterministic components, each note was carefully trimmed and extrapolated in the same manner as we presented in 4.4.1. IRCAM's *pm2* was used for the purpose of the separation. Both files had a sample rate of 44100 Hz and were mono. The analysis for separation was performed with a Blackman window of 1024 points, oversampled to 8192 points and with a hop size of 128 samples–giving an anlaysis framerate of 344.5 Hz. The spectrogram of the deterministic component resulting from this operation for the B_3 note can be seen in figure 1.1.

5.1.3 Extraction of the deterministic to stochastic energy ratios

Once the stochastic and deterministic components were separated, we were able to perform a calculation of the energy ratio between the two.

5.1.4 Extraction of descriptors from the deterministic component

Most of the features for which we tested interpolation strategies correspond to the deterministic component. We extracted these features for their subsequent interpolation. The two extracted deterministic components can also be found on the project's website [9]. The B_3 has been linked here and the $F\sharp_4$ has been linked here in the electronic version.

Even to odd ratio

The extraction of the even-partial to odd-partial energy ratio from the deterministic component can be performed in the time domain or in the frequency domain. By Parseval's identity, we know that the sum of the square of the Fourier coefficients of the FT of a signal is equivalent to the sum of the squared samples of the signal itself. Thus, given

[7] The absence of friction noise should also be relevant to bowed notes.

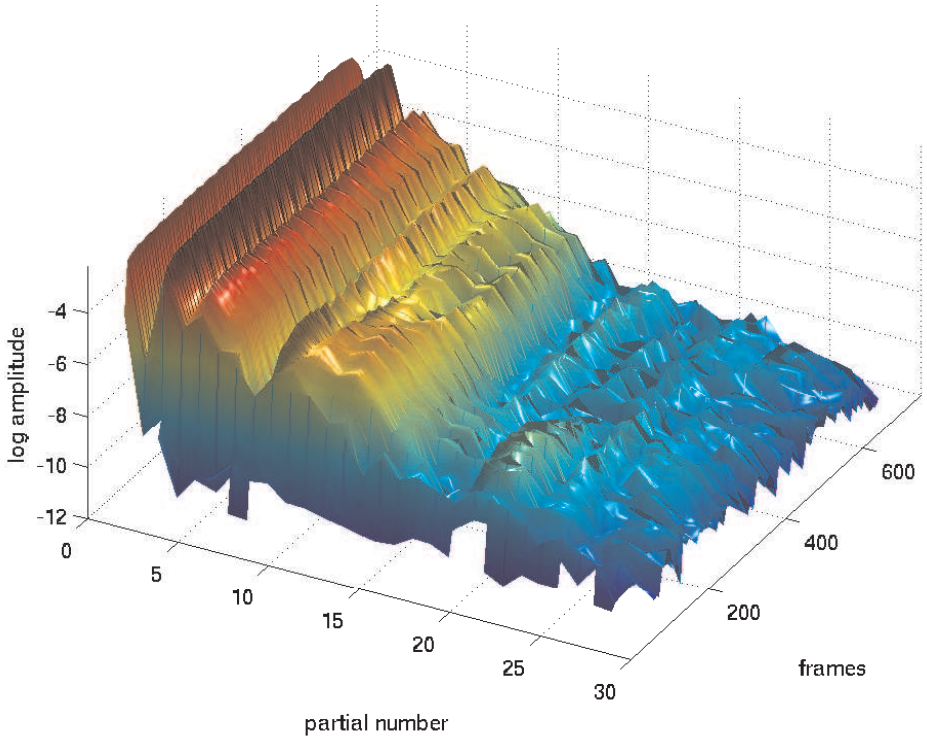


Fig. 5.2 Deterministic components of the B_3 clarinet tone, amplitude is presented in deciBels. Vibrato is clearly visible as is EOR in the first few partials.

that the information we receive from *pm2* for the deterministic component is essentially equivalent to a series of FT coefficients[8], the ratio between the sum of the square of all even-partial magnitudes and that of all odd-partial magnitudes should give us the EOR. Furthermore, since we will be performing a logarithmic interpolation of these ratios, we will obtain equivalent results if we use the ratio of the sum of all even-partial magnitudes and all odd-partial magnitudes. Performing the extraction this way reduces the operation to the quotient of two summations with no need to re-synthesize even and odd partials.

We found that by extracting the EOR and then *removing* it[9], produced an even more irregular spectrum than the original one. Additionaly, since the correction at higher frequencies makes all even partials louder than odd ones–see figure 5.4(a)–the result is perceived to be an octave higher. The resulting *jaggedness* of the spectral envelope results

[8]We obtain maximal amplitude values for the partials present in the deterministic component, where the contribution of all other bins to this component is zero.

[9]By dividing all even-partial magnitudes by it.

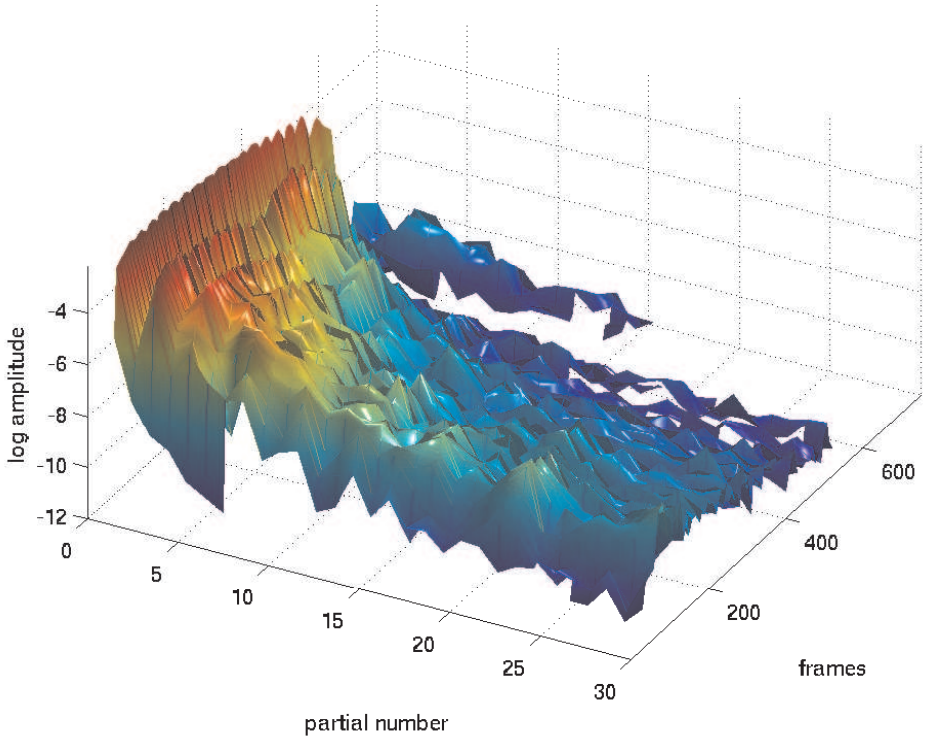
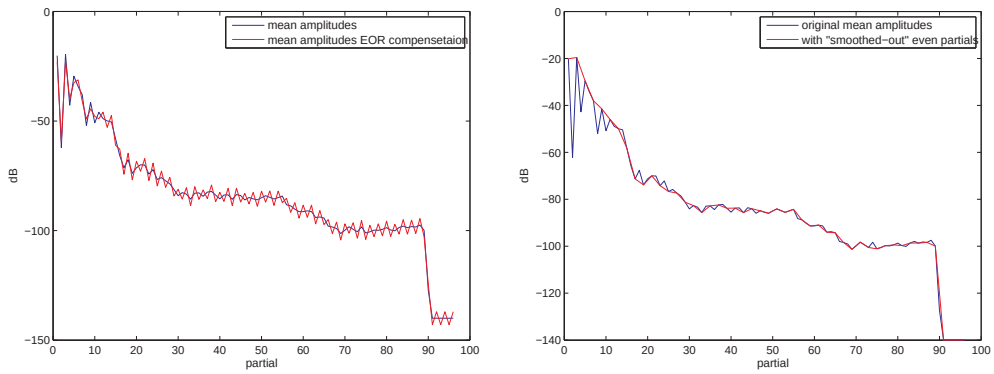


Fig. 5.3 Deterministic component, up to the partial number 30, of the $F\sharp_4$ clarinet tone. Amplitudes lower than -12 dB have been clipped.

from the low EOR of a stopped pipe being much more present in the first few partials than it is in higher partials[10], as an example, we can see a very low EOR below 6th harmonic, but closer to unity after the 8th partial on both figures 5.2 and 5.3.

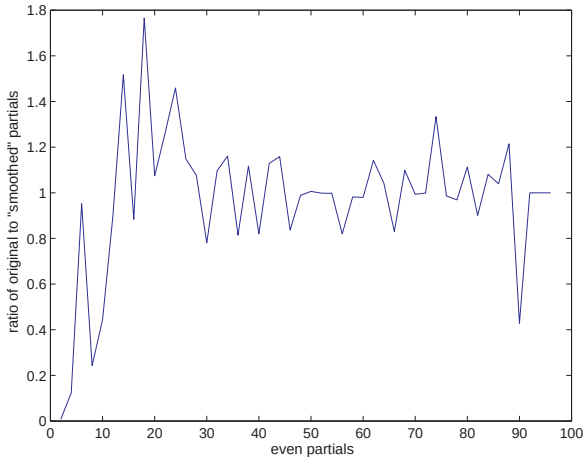
We implemented a tentative solution to this problem; we smoothed the spectral envelope by scaling the magnitude of even partials to lie at a point of log-interpolation between the magnitudes of odd ones and extracting a vector of ratios between the magnitudes of the original even partials vs the magnitudes of the logarithmically interpolated ones. This allowed us to estimate the envelope with a higher spectral smoothness than that which is characteristic of the clarinet. The importance of this spectral smoothness lies in that the loci at which the notches which characterize the spectral irregularity of the clarinet are correlated to the fundamental frequency of the pitch that is being played; making it undesirable to include these notches during spectral envelope interpolation. The steps involved in the proposed solution can be seen in figure 5.4.

[10]It is also more present in notes played *piano* than it is in notes that are played *forte*.



(a) Jagged spectrum

(b) Smooth spectrum



(c) Even-to-smooth ratios

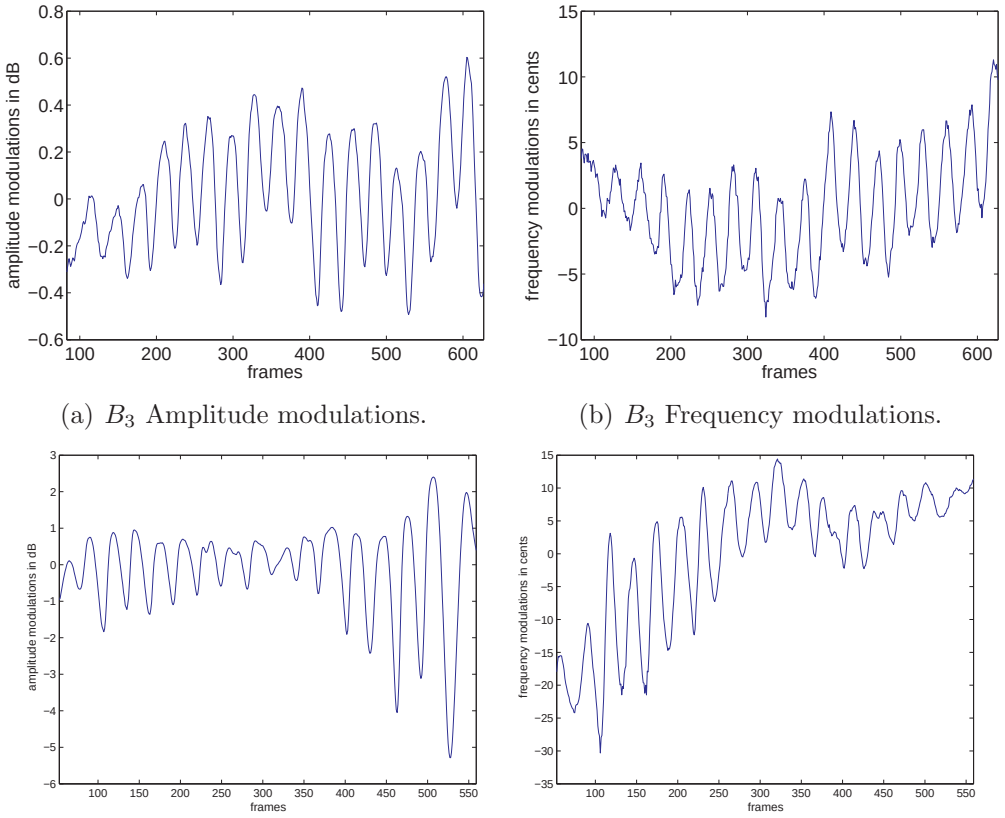
Fig. 5.4 EOR is present mostly in the lower partials of tones played with a *piano* dynamic, thus a single EOR value for all partials can be misleading. We utilize an alternative strategy for the same phenomenon which allows for different values per partial. 5.4(a) If we try to smooth out the spectral envelope by removing a unified EOR, upper partials end up jagged, where odd partials are louder than even ones, sounding an octave higher. 5.4(b) We omit all even partials and the resulting spectral envelope is much smoother. 5.4(c) Then we can extract a vector of ratios of the magnitudes of all even partials to the magnitude of the smooth envelope at equivalent loci.

Vibrato

As we have discussed in chapter 3 we will take vibrato to be the ensemble of amplitude and frequency modulations. With the intention of evaluating the quality of an approximation, we only extracted modulations of the first partial. This approximation was based on

the premise that frequency modulations of all partials are very similar and amplitude modulations of all partials are also fairly homogeneous.

Both the frequency and amplitude the modulations were extracted in the manner previously detailed in 4.4.1 and 4.4.1 with one notable exception; instead of using the ideally filtered envelope, we used the line-segment amplitude envelope approximation from 5.1.1, scaled to fit the magnitude of the first partial's amplitude envelope. After extracting the partial tracks of the extrapolated modulations of the complete envelope, we kept only the frames pertaining to the stable stage of the musical sound object and discarded all other frames.



(a) B_3 Amplitude modulations.

(b) B_3 Frequency modulations.

(c) $F\sharp_4$ Amplitude modulations.

(d) $F\sharp_4$ Frequency modulations.

Fig. 5.5 Clarinet tone modulations

PAT and PRT

Although the extraction of the global amplitude envelope is performed on the complete musical sound object, partial-specific attack and release times must be obtained from the amplitude envelope of each individual partial in order to attempt to recreate it from the global envelope during interpolation. Extraction of this value proved to be difficult due to the heavy presence of modulations. Thus, using any single criteria–such as maximum effort or maximum amplitude–for finding the peak of the attack proved to be unreliable. We resorted to a combination of the following: peak rate of change, amplitude thresholds and time constraints. Both peak effort and amplitude thresholds are suggested by Peeters[60]; the additional time constraints were added for robustness.

The estimation of the end of each partial's attack was performed by evaluating a series of candidate points. The points were chosen from local maxima[11] that were located after the first attack frame[12] and before half of the sound's duration. No local maxima under half of the sound's maximum amplitude were considered candidates. For each candidate point, the slopes of the line crossing the start of the attack and the given candidate point were taken to be it's effort. Thus, from the pool of possible points, the one having the maximum *effort* was chosen to be the end point of the partial's attack.

The end of the release was much simpler to find: we chose the first point after the release frame 0[13] having an amplitude $20dB$ lower than the start of the release.

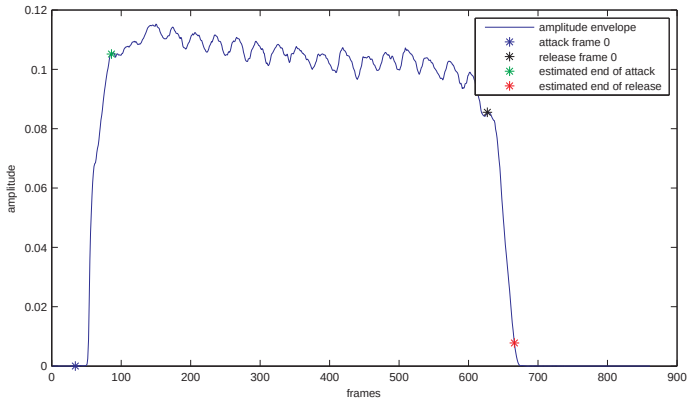


Fig. 5.6 PAT and PRT estimation for the B_3 clarinet tone

[11] found by means of the derivative method
[12] Per the global amplitude envelope extraction in 5.1.1.
[13] Once again, of those found in 5.1.1.

Spectral envelope

Although we already had a line segment approximation of the spectral envelope in 5.1.4, we were interested in transforming it into a function with a continuous derivative, such as that given by cepstral coefficients. The reason for doing this was that the interpolation of spectral envelopes via reflection coefficients is not as well-behaved when the envelopes are given by line-segments instead of a smooth function such as that given by cepstral coefficients. Thus, in order to obtain a smoother envelope, we used *pm2* to synthesize a brief audio file having the average frequencies and amplitudes[14] of partials during the stable stage of the sound object. We subsequently used *supervp* to perform a true envelope estimation on the newly synthesized file. Upon importing the resulting estimation, we kept the middle frame, discarding the effect of discontinuities at the start and end of our audio file. The resulting envelope for the B_3 tone can be seen in figure 5.7

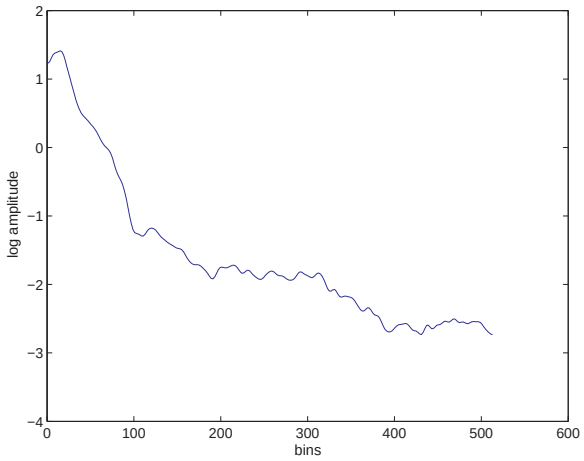


Fig. 5.7 True envelope estimation for the B_3 clarinet tone, obtaining the envelope this way yields a much smoother envelope than that given in 5.1.4 as can be seen by comparing this figure to figure 5.4(b).

[14]Having corrected the even partials as described in 5.1.4

Inharmonicity

Inharmonicity was easily found once we had the average partial frequencies from the stable region of the sound object by applying the following equation.

$$Inharm_h = 2 \cdot \frac{f_h - (f_0 \cdot h)}{f_0} \tag{5.1}$$

Where $Inharm_h$ is the inharmonicity coefficient for partial h, f_h is the average frequency of the same partial and f_0 is the sound object's fundamental frequency, which, in this case, is taken to be the average frequency of the first partial.

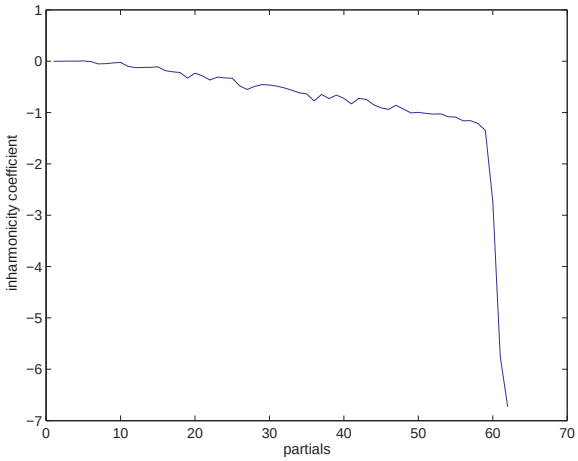


Fig. 5.8 Inharmonicity coefficients for the $F\sharp_4$ clarinet tone, a steep drop from the normal 1 to -1 range can be seen in the upper harmonics; a possible indicator of them being spurious.

At least in the case of sounds with a strong harmonic structure like the two clarinet notes used herein, we found that the inharmonicity coefficient could be a helpful gauge of spurious harmonics. In both sounds, the highest partials had inharmonicity coefficients with a magnitude far exceeding 1, meaning they were well into frequencies that would generally correspond to other harmonic components. An illustration of the inharmonicity coefficients of the $F\sharp_4$ clarinet tone can be seen in figure 5.8.

5.1.5 Extraction and morphing of features from the stochastic component

Given the separation of stochastic and deterministic components, we must also extract features from the stochastic component for their subsequent interpolation. We only need

to extract the amplitude envelope and the spectral envelope. The two extracted stochastic components can also be found on the project's website [9]. The B_3 has been linked here and the $F\sharp_4$ has been linked here in the electronic version.

Amplitude envelope

The goal of extracting the amplitude envelope from the stochastic component lies in being able to scale the *grains* of our subsequent OLA resynthesis. As grains are essentialy created by frequency-domain filtering of noise, using the spectral envelope of the corresponding analysis frame of the stochastic component and then scaling the grains to have the desired power. Thus we deemed it more practical to actually extract a power envelope instead of an amplitude envelope. The chosen method of estimating the temporal envelope was to obtain the stochastic part's power every 128 samples and then smooth it with a 64-point Bartlett-weighted moving average filter. Finally, we compensated the group delay by shifting the envelope 32 frames.

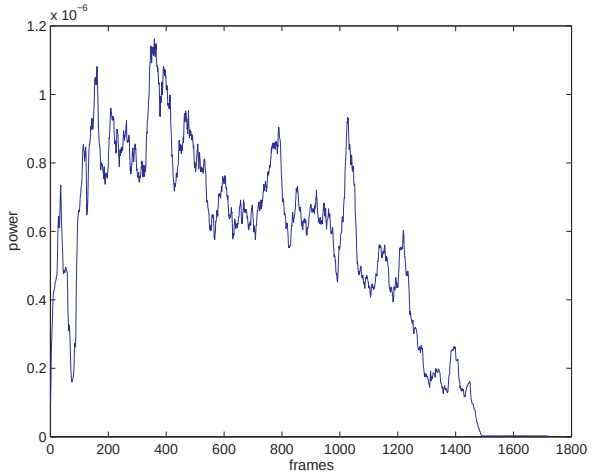


Fig. 5.9 Power envelope for the stochastic component of the B_3 clarinet tone.

5.1.6 Spectral envelope

We had originally planned to extract a single spectral envelope to represent the stochastic component and apply modifications for measured spectral centroid changes throughout the attack and release. However, shifting the spectral centroid can be achieved in several ways,

each of which produce different spectra. We decided to avoid the direct manipulation of the spectral centroid. Instead, we kept the succession of spectral envelopes for the whole stochastic component. Working with a series of envelopes corresponding to each analysis frame guaranteed that we implicitly retained spectral centroid values at each frame. This descriptor was obtained by *supervp*, performing a true-envelope estimation with an analysis hop size of 128 samples.

5.2 Cyclo-stationary morphing

Of the three possible types of morphing, we decided to try our hand at cyclo-stationary morphing first. One of the remaining possible types: stationary morphing, can be reduced to a cyclo-stationary morph of a single intermediate stage. It then followed that whichever procedure we found to work for cyclo-stationary morphs should also prove useful for performing stationary morphs. We performed a cyclo-stationary morph with the interpolation coefficients being 0, 0.25, 0.5, 0.75, and 1. We thus generated three intermediate pitches, between B_3 and $F\sharp_4$ which were slightly further apart than a major second.

5.2.1 Deterministic component morphing

We found a one-to-one partial correspondence to be adequate and opted to discard partials that presented indications of being spurious[15], such as very low amplitudes; highly variable amplitudes or frequencies; or inharmonicity coefficients outside of the range $[-1, 1]$. This left us with the first fifty partials of both sounds. Generally speaking, the process involved the separate morphing of the deterministic and stochastic parts.

For the purpose of morphing the deterministic component, we proceeded by performing dynamic time-warping on the global amplitude envelope; warping f_0; resampling and interpolating the vibrato; interpolating all other parameters and using them for the purpose of imprinting the resulting interpolated features onto a series of 50 harmonic partials of equal magnitude–a band-limited impulse-train. Two steps of the process proved to be slightly more involved than others: vibrato interpolation and generating an amplitude envelope for each partial from the global amplitude envelope and the partial's PAT and PRT coefficients.

Most interpolations were relatively simple to perform. Fundamental frequency[16] inter-

[15] So, generally higher-order partials

[16] Which we took to be the mean frequency of the first partial throughout the stable stage.

polation was performed logarithmically; spectral envelopes were interpolated via reflection coefficients; inharmonicity was interpolated linearly; even-to-smooth ratios[17] were interpolated logarithmically as were PAT and PRT coefficients.

Amplitude envelope

Dinamic time-warping was performed on the global amplitude envelope contemplating pre-attack, attack, stable, release and post-release sections of both sounds. The global envelope was then only employed as a guide for time-warping and as a basis for generating each individual partial's amplitude envelope. Thus, amplitudes between both global envelopes were not interpolated at this point. It was only after having interpolated PAT and PRT for each partial that partial-specific envelopes were created. Times for the beginning of most stages[18] were kept from the global envelope, yet the time for the end of the attack and for the end of the release were found through the interpolated PAT and PRT coefficients. This effectively scaled the time alloted to the stable section of the envelope in each partial. Line segments contained in both global amplitude envelopes were then scaled so that the total duration of both corresponded to the duration allotted to the stable stage of that particular partial. Points from each line-segment approximation were then projected onto the other sound object's stable-stage envelope. As a result, we were able to define amplitudes values at all the times previously defined in either one of the stable-stage amplitude envelopes. The process is illustrated in figure 5.10.

This procedure results in envelopes that recreate the expected spectral flux during the onset by interpolating the partial attack times of both sound objects. Yet the fact that all envelopes have the same shape results in a somewhat unnatural sound. The motivation for using a single amplitude envelope was to evaluate if data could be significantly reduced in the model without too great a loss in quality. Although the end results are reasonable, the compromise is noticeable. As we can see in both figures 5.2 and 5.3, envelopes tend to vary significantly between partials, suggesting that exploring a strategy that preserves individual partial amplitude envelopes would still be desirable.

[17] used instead of EOR.

[18] With the exception of the stable and post-release stages.

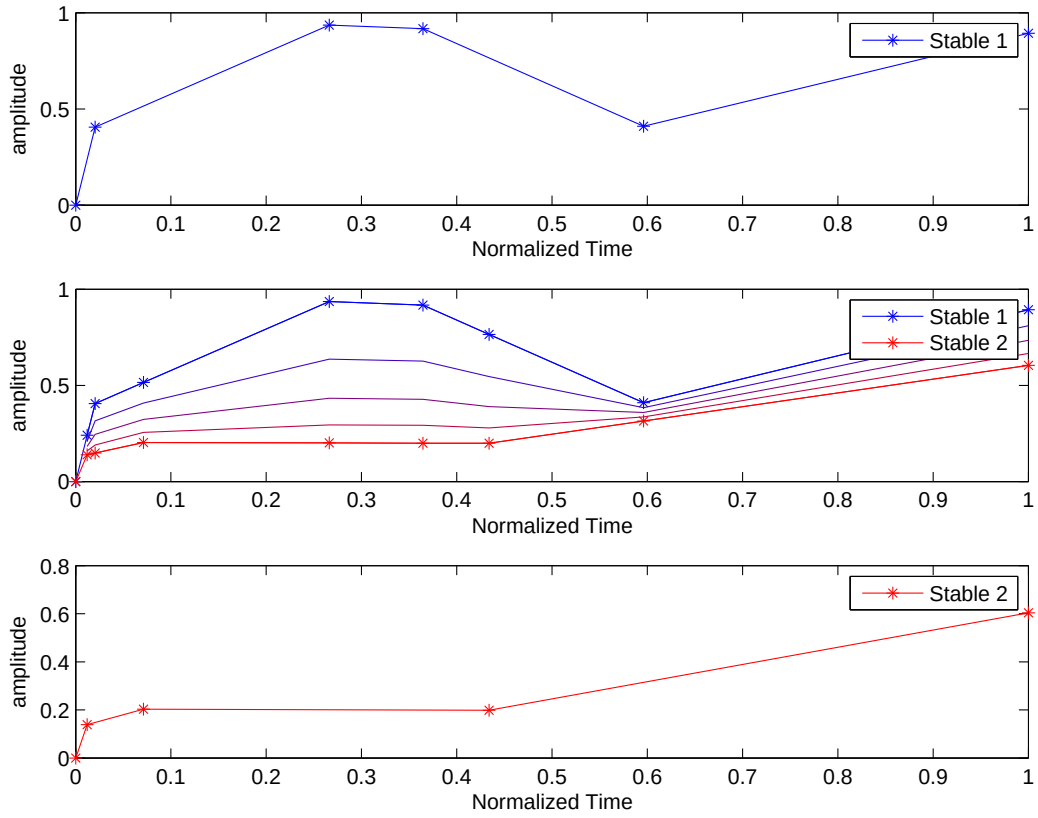
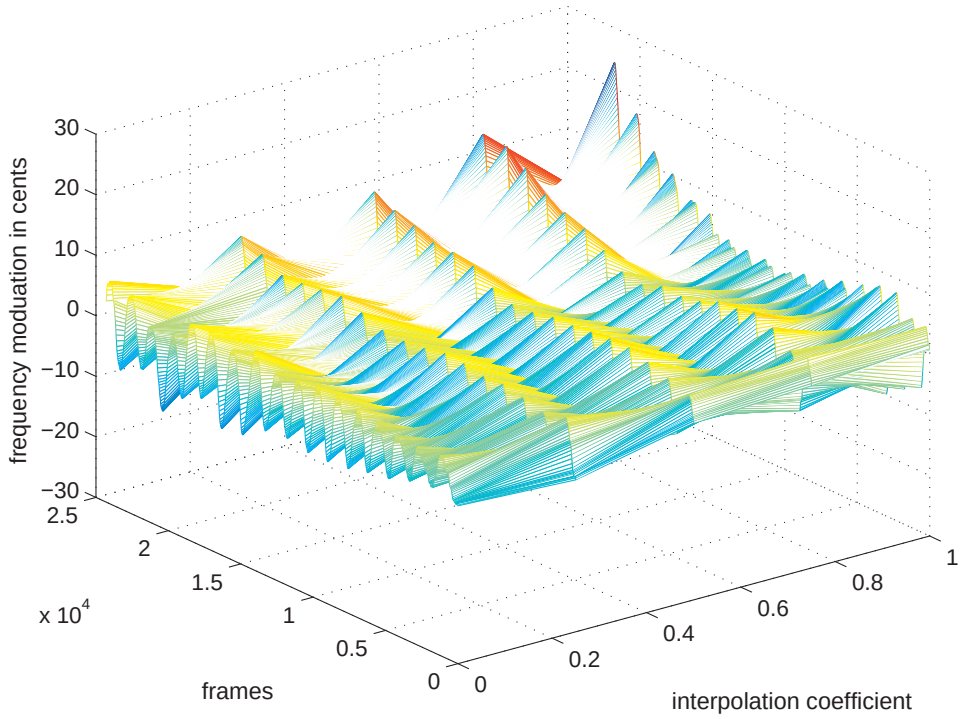


Fig. 5.10 Preparation for the interpolation of the stable stage of amplitude envelopes. Points from each envelope are projected onto it's counterpart, so that both envelopes, without having changed shape, have the same number of points defined at the same times.

Vibrato

Vibrato from both sounds was warped to match the duration of the globally warped stable stage. This was achieved by means of resampling and changing time as described in 4.4.2. Once the target amplitude and frequency modulations were interpolated, they were applied to every partial at frames corresponding to the global stable-stage, regardless of the partial's stable stage after PAT interpolation; this was necessary to guarantee the synchronous modulations across all partials.



(a) Frequency modulations.

Fig. 5.11 Resulting interpolations of frequency modulations of the target clarinet sounds. We refer the reader to figures 5.5(b) and 5.5(d), where both frequency modulation signals are presented.

Synthesizing the deterministic part

After having carried out all necessary interpolations, we then proceeded to imprint all features onto the harmonic structure of a band-limited impulse-train, with a fundamental frequency of the interpolated f_0. The resulting harmonic structure would later be used to synthesize the deterministic component. Partial frequencies were modified according to the inharmonicity coefficients and subsequently modified via the frequency modulation component of the interpolated vibrato. Also, having established the shape of the normalized amplitude envelope for each partial, it was then modulated by the amplitude modulation component of the interpolated vibrato; scaled by its even-to-smooth ratio, in the case of even partials; and scaled by the amplitude of the interpolated spectral envelope at the locus of the instantaneous frequency for each frame. Due to its dependence on the instantaneous frequency, it was deemed important to ensure that frequency modulations

and inharmonicity adjustments took place before scaling with the spectral envelope. The resulting harmonics were exported from *matlab* as an SDIF file and synthesized through *pm2*. The resulting morphs of the deterministic component can be found on the project's website [9]. In the electronic version, they have been linked here.

5.2.2 Stochastic component morphing

Morphing the stochastic component proved to be relatively simple, involving time-warping, amplitude envelope[19] warping and spectral envelope warping. Similarly to the deterministic component, we warped time based on the pre-attack, attack, stable, release and post-release sections of the global amplitude envelope. Warping the power envelope is a trivial process however, it is worth mentioning that warping the spectral envelope frames and preserving a uniform frame-rate was achieved by means of the same reflection-coefficient spectral-envelope interpolation which was presented in 4.2.3. A uniform frame-rate for both the power envelope and the spectral envelope was deemed important since it would allow us to interpolate the features from both sound objects on a frame by frame basis. By the same token, keeping the power envelope and the spectral envelope information at the same rate allowed us to scale each envelope for the OLA re-synthesis. The grains for the overlap-add were generated by multiplying a unity-gain and random-phase spectrum with each scaled spectral envelope[20], performing an inverse FT on the result and multiplying by a window function[21]. The result of this morph can be found on the project's website[9]. In the electronic version, they have been linked here.

5.2.3 Mixing

Having synthesized both morphed components, mixing them was only a matter of adjusting the stochastic component's gain in order for it to conform to the interpolated deterministic-to-stochastic component energy ratio. The resulting mixed morph can be found on the project's website[9] and has been linked here in the electronic version.

[19] In fact, power envelope.
[20] Its symmetric version, that is.
[21] The grains were 1024 samples and we used a Blackman window.

5.3 Dynamic morphing

We found this morph to be the perfect opportunity to try a phantom-partial f_0 interpolation strategy, diminishing the span of the *glissando*. Thus we interpolated between B_3 and an $F\sharp_3$ with phantom partials, eventually turning into an $F\sharp_4$. This decision resolved partial-matching for the morph. The amplitudes of phantom-partials were chosen to be -80 dB. Since we had two sound objects with different durations for each of their five sections, we perceived the difficulty of the dynamic morph to lie in finding a simple time warping strategy. We chose to resample one of the sound objects–the one with the shorter stable stage–to a sampling rate which caused both sounds to have the same number of frames for their stable section. This effectively reduced the sampling interval of the resampled file. Afterwards, we kept an interpolation coefficient of 0 during the pre-attack and attack stages and we enforced an interpolation coefficient of 1 for the release and post-release stages. The dynamic part of the morph took place during the stable stage, where the interpolation coefficient goes linearly from 0 to 1 with an equal rate of change per frame. In order to achieve the dynamic time warping, the sampling interval at each frame was interpolated logarithmically from the sampling periods of both sound objects[22].

Due to the change in framerate, PAT and PRT coefficients were converted to frame values, instead of time. All other parameters were interpolated on a per-frame basis, as the interpolation coefficient changed. The rest of the procedure was generally equivalent to the cyclo-stationary morph. The results can be found on the project's website[9]. They have also been linked here.

[22]Meaning that although the interpolation coefficient changes linearly throughout stable frames, it changes exponentially in time.

Chapter 6

Summary

At the onset of this project, we embarked on a reconnaissance of the scholarly literature on morphing, drawing on established bodies of work in image processing and speech processing. In so doing, the aim was to improve the understanding of morphing within the context of isolated instrumental tones. This work remains exploratory in its nature and is intended as an aide for those who seek to acquaint themselves with morphing audio and perhaps implement morphing algorithms. It is particularly intended as a pedagogical tool for those with more of a musical background than a technical one–albeit hopefully also useful for the latter. In this chapter we recapitulate and evaluate how we fared in our research and conclude by proposing possible improvements to the austere implementation that we have prepared as part of our project, in the event that others wish to replicate such an undertaking.

6.1 Conclusions

Whether stemming from high-resolution analysis or from Fourier-based analysis, we have seen that most modern approaches to sound morphing employ an additive model. Our research corroborates this as a wise choice. We found it to be an ideal model for achieving a controlled timbral manipulation, particularly with instrumental sounds, which are most often characterized by a well defined set partials–if not a fully harmonic spectrum. Moreover, as an established standard in the field, the stochastic-plus-deterministic additive model seems even better suited for the purpose than the purely additive one.

Among it's many virtues, it is a helpful model for extracting timbre descriptors–one of

the focal areas of our research. We remain convinced of the need to have a series of meaningful descriptors and we found the set chosen for this work to be sufficient. Conversely, some of the features of the set seemed far more convincing within the isolated-feature interpolations from chapter 4 than they did in chapter 5. This is a point which will be discussed more fully in 6.2. The reader should not take this to mean that the testing from chapter 4 did not succeed in contributing towards real-world implementation. Rather, we consider that they have shown to be of exceptional heuristic value in the progression of our research.

We briefly summarize our appraisal of the interpolation of each one of the chosen descriptors for the purpose of effecting a convincing morph between musical sound objects.

6.1.1 Amplitude envelope warping

A classical descriptor, the amplitude envelope is invaluable for warping a set of sound objects to bring crucial unique temporal features or temporal stages–such as the attack– to take place simultaneously. We based our implementation on the basis that amplitude envelopes for all partials of a given musical tone will generally retain a degree of similarity. Surprisingly, our first implementation of morphing real-world sounds proved this to be a flawed assumption. We suggest exploring an algorithm that interpolates per-partial amplitude envelopes and per-partial modulations. This is discussed further in section 6.2. Nevertheless, we consider that the use of a global envelope may be used to improve the temporal match of sound objects, allowing a generalized time-warp which enforces temporal feature simultaneity. Furthermore, we have proposed a model for temporal division of a sound object based on its envelope; yielding five sections: pre-attack, attack, stable, release and post-release. Both the pre-attack and post-release stages contain ancillary sounds: respectively noises from preparation for attack and adjustments after playing. The attack, on the other hand, is a transient stage characterized by a relatively high spectral flux and a fast rate of energy increase. The stable stage[1] refers to the part of the sound where energy is still being input into the instrument, such as blowing for a wind instrument or bowing for strings. It is here that we may encounter an extended vibrato[2]. Finally, the release stage is when the instrument's oscillations fade out, not being excited anymore; it generally

[1] Which does not necessarily need to be stable in amplitude.

[2] A limit case springs to mind; while not directly inputting energy into a guitar, we can still produce a vibrato as a note rings. In this case, the imprint of a vibrato is actually inputting energy, however minimal, into the strings' oscillations.

presentins a higher degree of spectral flux than the stable stage. For some musical sound objects, the release stage will be very short, while others might present a prolonged release stage[3].

6.1.2 Warping f_0

This is a fairly standard part of morphing and is generally perceived to be smoother if interpolation is performed logarithmically. For larger intervals in dynamic morphs, f_0 interpolations may produce an overbearing *glissando* which can be circumvented if we glide to a closer pitch which is related to the target pitch in its harmonic structure. A generalized formalization of this idea is presented in algorithm 4.1.

6.1.3 Interpolation of vibrato

The interpolation of vibrato is performed exclusively during the stable stage and is achieved through a second-order sinusoidal analysis[58]. If second-order partial-matching is made following a closest-amplitude criterion, results are optimal. Yet we have based our procedure on the premise that the relative difference of modulations is roughly equal for all partials. However, the results of the procedure revealed this to be yet another incorrect assumption. In fact, during our implementation of morphing, we found relative differences to be equal only for frequency modulations, and we also found that amplitude modulations were much more pronounced in even partials, as can be seen in figure 6.1.

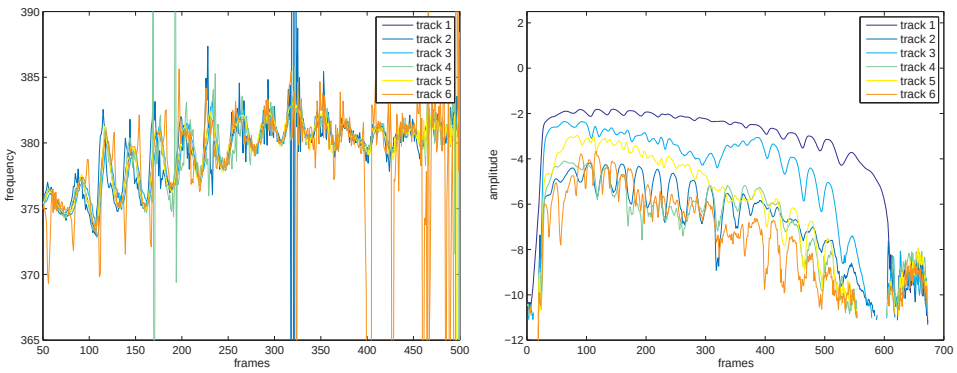
6.1.4 Inharmonicity

Inharmonicity proved to be a meaningful parameter, for which linear interpolation produced reasonably smooth results.

6.1.5 Even to odd partial energy ratio

The proposed EOR is a good descriptor of timbre, having a strong correlation with the spectral irregularity of a given sound. Yet, as we saw in 5.1.4, it is not advantageous for manipulating and resynthesizing, since this ratio is not constant across all partials, decreasing rapidly as the order of partials increases. We have thus proposed an alternative

[3] Again, plucked strings come to mind, where most of the note can be though of as a release stage.



(a) Normalized frequency modulations. (b) Log-amplitudes with modulations

Fig. 6.1 Comparison of frequency modulations and amplitude modulations. 6.1(a) presents the relative breadth of frequency modulations of the first few partials of a clarinet tone, while 6.1(b) presents the log amplitude of a few partials, wider amplitude modulations are evident in even partials.

vector of ratios of all even partials to a smooth version of the sound object's spectral envelope. In the real-world interpolations effected during our research this vector still retains the *stopped pipe* quality of the sound without distorting the spectral envelope. Logarithmic interpolations of either EOR or even-to-smooth ratios yield smooth progression while linear interpolations yield unacceptably unequal progressions.

6.1.6 Partial attack times, partial release times

The purpose of these two measures was to retain spectral centroid changes present in the original sound object while using a global amplitude envelope, yet given that it would be best to work towards a model that interpolates amplitude envelopes for each partial[4], we consider that these two measures should prove to be obsolete. However, if we were to use them, they should be logarithmically interpolated for optimal results.

6.1.7 Spectral envelope

From cross-synthesis to present day state-of-the-art, the spectral envelope constitutes a cornerstone for most attempts at audio morphing. We deem it essential to retain the use of this descriptor in any framework aimed at morphing. Although direct formant

[4]While still extracting amplitude modulations before the interpolation of the envelopes.

interpolation is best when formant parameters are available, we have explored interpolation of an arbitrary envelope via reflection coefficients and find it to produce very good results.

6.1.8 Deterministic vs stochastic energy ratio

This ratio is very important for mixing the resulting components. In the case of cyclostationary or stationary morphs, the logarithmic interpolation of a single scalar is sufficient. In the case of dynamic interpolations, we need to interpolate the energy on a per-frame basis, effectively turning this into a power interpolation. In the latter case, the power interpolation will be a time-varying gain of the stochastic component and therefore render the stochastic component's amplitude envelope useless.

6.2 Further development

Our research findings suggest possibilities for new directions of exploration and research. We here relay to the reader our thoughts on these possibilities.

Firstly, although we resorted to a manual approximation of the global amplitude envelope, this is an extremely unwieldy process. Moving-average filters already yield reasonable results and could easily be used if we sacrifice precision for the timing of the sound object's unique temporal features. One possibility for increasing the precision of a moving-average estimation of the envelope could be to drive the filter's parameters with spectral parameters indicative of transients, such as spectral flatness or spectral flux.

Another important avenue of possible future research relates to a common practice in audio morphing. We have previously mentioned that there are many proponents of morphing partial amplitudes directly. While this is far from being the interpolation of meaningful features that we seek, we must acknowledge that individual track envelopes are very important for preserving the natural quality of a sound object. Thus we propose that each partial's amplitude envelope should be treated in the same way we have dealt with some of the global parameters: extracting vibrato from each partial and temporally warping the partials' smooth amplitude envelopes in accordance with unique-temporal-features warping from the interpolands' global envelopes. Another argument in favor of this proposition is that spectral envelope modulation can be shown to be implicit if both amplitude and frequency modulations are unique to each partial.

The way we performed vibrato extraction, via *pm2*, did not yield values for bin 0. Thus all modulations were centered around zero. This eliminates the possibility of including micro-melodic movements in modulations, so we propose the inclusion of a dc-offset partial in second-order sinusoidal analysis. The relevance of its inclusion has already been stated by Marchand and Raspaud[58].

Most of our efforts have centered around the deterministic component of sound objects. We have thus overlooked parameter extraction from the stochastic component to a certain extent. The need for the extraction of at least one feature has become evident: extended vibrato. Although there is no fundamental frequency in the stochastic component, amplitude modulations and spectral envelope modulations are present. The frequencies of these modulations are correlated with the frequencies of the modulations present in the deterministic component and should also be interpolated for consistency. The extraction of the former should not prove too difficult, following a similar procedure to the one we used to extract amplitude modulations from the deterministic component. Conversely, a method for obtaining the spectral envelope modulations remains unknown to us.

One practical consideration remains in regards to all phantom-partial interpolations. We have observed that the effect of phantom partials is most effective when they decrease in amplitude to a point at which they almost cannot be perceived–which is generally far louder than -96 dB. We therefore consider it a worthwhile effort to set phantom partials' amplitudes to a point just a few decibels lower than their temporal and frequency masking thresholds. Lastly, we consider that moving towards using similar descriptors based on perceptually motivated units would prove highly beneficial.

Appendix A

Note Regarding the Available Code

Our research is yet far from being aimed at producing a working suite of scripts capable of unsupervised morphing. Regardless, we are aware of the pedagogical value and overall usefulness of the scripts that we used to evaluate strategies and to produce examples. The scripts have been tailored to work in a somewhat idiosyncratic setup, including a particular directory structure and licensed copies of Mathworks' *Matlab* and IRCAM's *pm2* and *supervp*[1]. Yet, even if the reader does not have access to the required software, the code may reveal the procedures that we have followed from a different perspective than that offered from the writing. We have thus made it available on the project's website[9].

Reader's who have access to the required software may set up an environment to run the scripts by downloading the code and following the instructions in the README file. Window's users should be aware that, at least at the time of writing this document, supervp does not work on current window's versions. An alternative is to setup the software and downloaded scripts inside a virtual machine running any distribution of LINUX.

Given that the code is a relatively artisanal sandbox for prototyping and evaluating some interpolation strategies, let the user be warned: it does not have consistent naming conventions or consistent encapsulation. Although coding style may change from one script to the next to reflect the coder's whim throughout the research period, we have tried to maintain a habit of writing comments.

[1] These are the kernels of IRCAM's *Audiosculpt*.

References

[1] T. Wishart. "Red Bird, Anticredos"[CD], 1980. EMF.

[2] M. Slaney, M. Covell, and B. Lassiter, "Automatic audio morphing," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 1001–1004, 1995.

[3] X. Serra, "Sound hybridization techniques based on a deterministic plus stochastic decomposition model," in *Proceedings of the International Computer Music Conference (ICMC)*, 1994.

[4] E. Tellman, L. Haken, and B. Holloway, "Timbre morphing of sounds with unequal numbers of features," *Journal of the Audio Engineering Society (AES)*, vol. 43, no. 9, pp. 678–689, 1995.

[5] P. Cano, A. Loscos, J. Bonada, M. D. Boer, and X. Serra, "Voice morphing system for impersonating in karaoke applications," in *Proceedings of the International Computer Music Conference (ICMC)*, 2000.

[6] K. Fitz, L. Haken, S. Lefvert, and M. O'Donnell, "Sound morphing using Loris and the reassigned bandwdith-enhanced additive sound model: Practice and applications," in *Proceedings of the International Computer Music Conference (ICMC)*, 2002.

[7] W. Hatch, "High-level audio morphing strategies," Master's thesis, Music Technology Area, Schulich School of Music, McGill University, August 2004.

[8] W. A. Sethares, A. J. Milne, S. Tiedje, A. Prechtl, and J. Plamondon, "Spectral tools for dynamic tonality and audio morphing," *Computer Music Journal*, vol. 33, no. 2, pp. 71–84, 2009.

[9] F. O'Reilly. `http://www.music.mcgill.ca/~fede/thesis.html`.

[10] J. M. Grey, "Multidimensional perceptual scaling of musical timbres," *Journal of the Acoustical Society of America*, vol. 61, pp. 1270–1277, May 1977.

[11] J. M. Grey and J. W. Gordon, "Perceptual effects of spectral modifications on musical timbres," *Journal of the Acoustical Society of America*, vol. 63, pp. 1493–1500, May 1978.

[12] K. W. Schindler, "Dynamic timbre control for real-time digital synthesis," *Computer Music Journal*, vol. 8, pp. 28–42, 1984.

[13] M. Dolson, "Recent adventures in musique concrete at carl," in *Proceedings of the International Computer Music Conference (ICMC)*, 1985.

[14] R. Mcaulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.

[15] T. Lysaght and D. Vernon, "Timbre morphing of synthesised transients using the wigner time-frequency distribution," in *Proceedings Anais do XIX Congresso Nacional da Sociedade Brasileira de Computaçao*, vol. 3 SBCM Simposio Brasileiro de Computaçao e Musica, pp. 1–7, 1999.

[16] T. Lysaght, D. Vernon, and J. Timoney, "Subgraph isomorphism applied to feature correspondence in timbre morphing," in *Proceedings of the Irish Signals and Systems Conference*, pp. 250–257, 2000.

[17] X. Serra and J. Smith, "Spectral modeling synthesis: A sound analysis/synthesis based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, pp. 12–24, 1990.

[18] X. Serra and J. Bonada, "Sound transformations based on the sms high level attributes," in *Proceedings of the Digital Audio Effects Workshop (DAFX98)*, 1998.

[19] T. Hikichi and N. Osaka, "Sound timbre interpolation based on physical modeling," *Acoustical Science and Technology*, vol. 22, no. 2, pp. 101–111, 2001.

[20] J. W. Beauchamp, ed., *Sound of Music: Analysis, Synthesis, and Perception*, ch. 3: Beyond Traditional Sampling Synthesis: Real-Time Timbre Morphing Using Additive Synthesis. Springer, 2007.

[21] V. Verfaille, J. Boissinot, P. Depalle, and M. M. Wanderley, "Ssynth: a real time additive synthesizer with flexible control," in *Proceedings of the International Computer Music Conference (ICMC)*, 2006.

[22] M. Caetano and X. Rodet, "Evolutionary spectral envelope morphing by spectral shape descriptors," in *Proceedings of the International Computer Music Conference (ICMC)*, pp. 171 – 174, 2009.

[23] M. Caetano and X. Rodet, "Evolutionary spectral envelope morphing by spectral shape descriptors." `http://recherche.ircam.fr/anasyn/caetano/icmc2009.html`.

[24] W. A. Sethares, A. J. Milne, S. Tiedje, A. Prechtl, and J. Plamondon, "Spectral tools homepage." `http://www.dynamictonality.com/spectools.htm`.

[25] "LORIS." `http://www.cerlsoundgroup.org/Loris/`.

[26] "Cameleon 5000." `http://www.camelaudio.com/cameleon5000.php`.

[27] "Diphone studio." `http://forumnet.ircam.fr/703.html?L=1`.

[28] "CDP." `http://www.composersdesktop.com/`.

[29] T. Wishart, *Audible Design*. Orpheus the Pantomime, 1994.

[30] T. Wishart. Liner Notes for the EMF recording "Red Bird, Anticredos"[CD], 1980. EMF.

[31] D. E. Jones, M. Decoust, C. Dodge, J. B. Barrière, T. Wishart, and R. Reynolds. "Computer Music Currents 4"[CD], 1989. WERGO.

[32] D. E. Jones, M. Decoust, C. Dodge, J. B. Barrière, T. Wishart, and R. Reynolds. Liner Notes for the WERGO recording "Computer Music Currents 4"[CD], 1989. WERGO.

[33] T. Wishart, "The composition of vox-5," *Computer Music Journal*, vol. 12, pp. 21–27, 1989.

[34] C. Dodge and T. A. Jerse, *Computer Music: Synthesis, Composition, and Performance*. Schirmer, 2nd edition ed., July 1997.

[35] P. Depalle, G. García, and X. Rodet, "A Virtual Castrato (!?)," in *Proceedings of the International Computer Music Conference (ICMC)*, 1994.

[36] P. Depalle, G. Garcia, and X. Rodet, "Reconstruction of a castrato voice: Farinelli's voice," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 15–18, October 1995.

[37] P. Floyd. "Animals"[CD], 1977. Harvest / EMI.

[38] M. Schröder, "Emotional speech synthesis: A review." http://www.dfki.de/ schroed/articles/schroeder2001, 2001.

[39] B. de Mareüil, P. Célérier, and J. Toen, "Generation of emotions by a morphing technique in english, french and spanish speech prosody," in *Proceedings of the 1st International Conference on Speech and Prosody*, 2002.

[40] Y. Stylianou, T. Dutoit, and J. Schroeter, "Diphone concatenation using a harmonic plus noise model of speech," in *Proc. EUROSPEECH*, pp. 613–616, 1997.

[41] H. R. Pfitzinger, "DFW-based spectral smoothing for concatenative speech synthesis," in *Proc. of the Int. Conf. on Spoken Language Processing (ICSLP)*, pp. 1397–1400, 2004.

[42] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. 655–658, Apr 1988.

[43] W. Verhelst and J. Mertens, "Voice conversion using partitions of spectral feature space," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. 365–368 vol. 1, May 1996.

[44] C. Orphanidou, I. Moroz, and S. Roberts, "Voice morphing using the generative topographic mapping," 2003.

[45] N. Iwahashi and Y. Sagisaka, "Speech spectrum transformation by speaker interpolation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. 461–464, Apr 1994.

[46] N. Iwahashi and Y. Sagisaka, "Speech spectrum conversion based on speaker interpolation and multi-functional representation with weighting by radial basis function networks," *Speech Communication*, vol. 16, no. 2, pp. 139–151, 1995. Voice Conversion: State of the Art and Perspectives.

[47] M. Abe, "Speech morphing by gradually changing spectrum parameter and fundamental frequency," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 4, pp. 2235–2238, 3-6 1996.

[48] H. Ye and S. Young, "High quality voice morphing," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. 9–12, 2004.

[49] H. R. Pfitzinger, "Unsupervised speech morphing between utterances of any speakers," in *Proc. of the 10th Australian Int. Conf. on Speech Science and Technology (SST 2004)*, pp. 545–550, 2004.

[50] H. R. Pfitzinger, "Unsupervised speech morphing." `http://www.phonetik.uni-muenchen.de/~hpt/morphing`.

[51] G. Wolberg, *Digital Image Warping*. IEEE Computer Society Press, 1990.

[52] T. Beier and S. Neely, "Feature-based image metamorphosis," *SIGGRAPH Comput. Graph*, vol. 26, no. 2, pp. 35–42, 1992.

[53] G. Wolberg, "Image morphing: a survey," *The Visual Computer*, vol. 14, no. 8/9, pp. 360–372, 1998.

[54] P. Borrel and D. Bechmann, "Deformation of n-dimensional objects," in *SMA '91: Proceedings of the first ACM symposium on Solid modeling foundations and CAD/-CAM applications*, (New York, NY, USA), pp. 351–369, ACM, 1991.

[55] M. Grundland, R. Vohra, G. P. Williams, and N. A. Dodgson, "Cross dissolve without cross fade: Preserving contrast, color and salience in image compositing," in *Proceedings of EUROGRAPHICS, Computer Graphics Forum*, pp. 577–586, 2006.

[56] B. D. Jacobson, *Combined-Channel Instantaneous Frequency Analysis for Audio Source Separation Based on Comodulation*. PhD thesis, The Harvard-Mit Division Of Health Sciences And Technology, September 2008.

[57] X. Amatriain, J. Bonada, A. Loscos, J. L. Arcos, and V. Verfaille, "Content-based transformations," *Journal of New Music Research*, vol. 32, no. 1, pp. 95–114, 2003.

[58] S. Marchand and M. Raspaud, "Enhanced time-stretching using order-2 sinusoidal modeling," in *Proceedings of DAFx*, 2004. Naples, Italy.

[59] V. Verfaille, C. Guastavino, and P. Depalle, "Perceptual evaluation of vibrato models," in *Proceedings of CIM*, 2005. Montreal, Qc, Canada.

[60] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," tech. rep., IRCAM, 2004.

[61] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. Wiley, March 1996.

[62] H. Fletcher, E. D. Blackham, and R. Stratton, "Quality of piano tones," *The Journal of the Acoustical Society of America*, vol. 34, no. 6, pp. 749–761, 1962.

[63] S. McAdams, "Perspectives on the contribution of timbre to musical structure," *Computer Music Journal*, vol. 23, no. 3, pp. 85–102, 1999.

[64] D. Schwarz and X. Rodet, "Spectral envelope estimation and representation for sound analysis-synthesis," in *Proceedings of the International Computer Music Conference (ICMC)*, pp. 351–354, 1999.

[65] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Prentice Hall PTR, April 1993.

[66] X. Rodet, Y. Potard, and J.-B. Barriere, "The chant project: From the synthesis of the singing voice to synthesis in general," *Computer Music Journal*, vol. 8, no. 3, pp. 15–31, 1984.

[67] P. Depalle, *Analyse, Modélisation et Synthèse des sons basées sur le modèle source-filtre*. PhD thesis, Académie de Nantes, Université du Maine, Faculté des Sciences, Decembre 1991.