Timbre Morphing of Sounds with Unequal Numbers of Features*

EDWIN TELLMAN, LIPPOLD HAKEN, *AES Member,* **AND BRYAN HOLLOWAY**

CERL Sound Group, University of Illinois, Urbana, IL 61801, USA

An algorithm is presented for morphing between sounds (timbre interpolation) using Lemur analysis and synthesis. The Lemur representation is made up of amplitude- and frequency-varying sinusoids. This timbre morphing involves time-scale modification of sounds (to morph between differing attack rates and vibrato rates) as well as amplitude and frequency modification of individual sinusoidal components.

0 INTRODUCTION

0.1 Definition

Timbre morphing is the process of combining two or more sounds to create a new sound with intermediate timbre and duration. For instance, a long loud sound with a fast and narrow vibrato may be morphed with a short quiet sound with a slow and wide vibrato to create a morphed sound with medium length and medium loudness, and with an intermediate vibrato speed and width. This process differs from simply mixing sounds, as only a single sound, with some of the characteristics of each of the original sounds, is audible as the morphed sound.

All the morphing algorithms discussed are in the frequency domain. The sounds to be morphed must first be analyzed to determine the magnitudes and frequencies of each partial (sinusoid) at any point in the sound. The output from the analysis of several sounds is combined to create the analysis of a morphed sound. This analysis may then be converted into samples and played on a synthesizer.

0.2 Motivation

Morphing can be used to create interesting new sounds which have some of the characteristics of familiar, naturally occurring sounds. For instance, a violin tone might be morphed with a trumpet to create a half-violin, half-trumpet effect. The result might be used in electronic music compositions.

Another use of morphing is to provide a more realistic synthesis of natural instrument tones. One way of synthesizing natural instruments is to record a large number of the instrument's tones. When a key is pressed on the synthesizer, the appropriate recording is played back. Tones for which there is no recording are synthesized by playing a recording at a new volume or pitch. Even if there are a large number of recordings, the result is generally easily distinguishable from the actual instrument. This is because there are spectral changes associated with dynamic and pitch changes. These changes are not captured by simply varying the amplitude and pitch of a recording. An actual loud piano note, for instance, sounds different from a quiet piano note played at a high volume.

Timbre morphing could be used to create intermediate tones between two of a synthesizer's recordings. When a tone that did not exactly match a recording was requested, a close approximation of the correct tone could be created by morphing between the closest stored recordings. The interpolated tone would still not duplicate the real instrument exactly, but it would be a closer approximation.

Timbre morphing can be used to provide a continuous transition between recorded timbres. For instance, a synthesizer might include several recordings of a violin, with a different vibrato rate in each of the recordings. In a synthesizer that played recordings without timbre morphing, there would be a sudden jump from one vibrato rate to the next when the synthesizer changes to a new recording. Timbre morphing can be used to increase the speed or amplitude of the vibrato gradually on transitions between recordings.

Another use of timbre morphing is in the design of new types of synthesizers. The Continuum synthesizer, currently in development at the University of Illinois,

* Manuscript received 1994 September 26; revised 1995 May 1.

provides a continuous playing surface which tracks the performer's motions in all three dimensions. Each dimension is associated with some characteristic of the sound. For instance, one dimension may be pitch, one may be volume, and the third may be brightness. A timbre morphing algorithm can be used to produce a continuous transition between the different timbres the instrument is able to produce [1].

A final use of timbre morphing is to provide a smooth transition between two synthesized notes. Without timbre morphing, the transition between one note and the next may be abrupt and unnatural. The transitions between notes are complex and are important in the identification of natural instruments [2], [3]. In 1992 Holloway and Haken investigated a sinusoidal synthesis algorithm which analyzed and synthesized musical transitions [4]. An alternative approach is to use timbre morphing to create smooth transitions between notes. The end of one sound could be morphed with the beginning of another to create a continuous transition between notes. This sort of transition may more closely duplicate some natural instrument transitions.

0.3 Terminology

0.3.1 Vibrato

Vibrato and *tremolo* are sometimes used to refer to repetitive variation of the frequency and amplitude, respectively, of a sound. In this paper we consider only sounds whose repetitive frequency and amplitude variations occur at the same rate. Consequently we use the term *vibrato* to refer to any spectral change that occurs repetitively over the course of the sound.

0.3.2 Timbre Space

Sounds may be thought of as occupying a *timbre space* with an arbitrary number of dimensions. Axes of timbre space may be labeled with various attributes of the sound: amplitude, frequency, distance of a violin bow from the bridge, bow pressure, and so on. Alternatively, the timbre space axes may not correspond to specific physical attributes of sounds.

Any sound can be placed somewhere in the timbre space. If timbre space dimensions correspond to physical attributes of the sounds, then sounds with the same characteristics in one dimension should have matching coordinates in that dimension. For instance, if the x axis of a three-dimensional timbre space corresponds to frequency, two sounds at the same frequency should have the same x coordinate. If the timbre space dimensions do not correspond to physical attributes, sounds to be morphed may be placed arbitrarily in the timbre space.

If several tones are provided somewhere in timbre space, it seems natural to create new sounds elsewhere in the space. The attributes of these new sounds would be determined by their distance and position relative to the original sounds.

0.3.3 Features

A *feature* in a sound may be any portion of the sound that is important in the morphing process. Examples of features are start of sound, peak of attack, peak of a vibrato cycle, and so on. Morphing algorithms often attempt to line up corresponding features so that the morphed sound has the characteristics of the original sounds.

0.4 Difficulties

At first, interpolating between two sounds may seem to be a trivial task. The obvious approach is to simply take a weighted average of each point on the amplitude and frequency envelopes of the partials of the original sounds to create the envelopes for the new sound. This straightforward approach will produce odd-sounding results for many sounds.

The attacks of sounds are important in their identification [5], [6]. Naturally occurring sounds have a variety of attack rates. A trumpet sound might have a fast attack whereas a French horn has a slower attack. If the envelopes of these two sounds are averaged, the new sound will have two averaged attack peaks rather than a single intermediate speed attack. An effective interpolation algorithm should average the attack speeds to produce a single averaged attack.

Natural sounds often vary in amplitude and frequency over their duration. As long as the overall contours of the sounds are similar, the contour of the morphed sound should be the same as the contour of each of the original sounds. For instance, one sound might begin at low amplitude, increase in amplitude for 1 s, then decrease in amplitude for 2 s. Another sound might begin at a low amplitude, increase in amplitude for 2 s, then decrease in amplitude for 1 s. If the partials of these two sounds are simply averaged, the morphed sound will have two amplitude increase–decrease cycles. A reasonable morphing algorithm would combine these sounds to create a sound that increases in amplitude for 1.5 s and decreases in amplitude for 1.5 s.

Vibrato is created by varying frequency or amplitude, or both, repetitively over the course of the sound. If two sounds to be interpolated have different vibrato rates, a simple weighted average will produce a new sound with peculiar and unnatural sounding or nonexistent vibrato. An effective interpolation algorithm would determine the rate and magnitude of the vibratos of the original sounds and produce a new sound with a vibrato of intermediate rate and magnitude.

A final difficulty is that interpolated sounds may have different lengths. If this is the case, it would be necessary to normalize the ending times of the two sounds to avoid having only a quiet version of one sound (the longer sound averaged with 0) playing at the end of the interpolated sound.

1 HISTORY

In 1975 Gray did a study of instrumental timbres [7]. He recorded a number of short instrumental sounds and studied the perceptual effects of various modifications of the sounds. One of his studies involved interpolating between two original sounds to create new intermediate sounds.

In 1984 Schindler investigated a method of reducing the amount of data required to represent amplitude and frequency envelopes [8]. He describes sounds as a hierarchical tree of timbre frames and discusses a morphing algorithm that operates on this representation of the data.

Haken implemented a real-time interpolation algorithm for the Platypus signal processor [9], [10]. His morphing algorithm interpolates in real time between sounds with equal numbers of features located on the corners of a cube in a three-dimensional timbre space.

Peterson investigated a different approach to combining sounds of differing timbres [11]. He analyzed one of the sounds to create a time-varying filter. This filter could be used to approximate the original sound when provided with a white-noise or pulse-train input. If an instrumental sound is input to the filter instead, the result is a combination of the two original sounds. For example, speech might be used as the analyzed sound and bassoon might be used as input to create a "talking bassoon" effect.

Depalle, Poirot, and Rodet implemented an algorithm for combining voice sounds [12]. They recorded and analyzed several singers and created an intermediate timbre voice by filtering one voice so that it had the formant characteristics of the other voice. They used this technique to combine tenor and soprano voices to create a castrato voice.

A generic system for analyzing and combining sounds was implemented at IRCAM. It is composed of several independent modules, each of which does some operation on the sounds. A system may include a combination module, which accepts multiple sounds as inputs and combines them in some way to produce one or more output sounds. Combination modules could be created to implement a morphing algorithm [13].

An example of a composition based on timbral modification of a recorded voice is "In Celebration" by Charles Dodge. He recorded and analyzed a reading of a poem. The analyses of sections were altered, combined, and resynthesized to create a multivoice rendition of the poem [14].

A commercially available morphing program is scheduled for release in 1995 by Oberheim Digital and G-WIZ Labs [15]. Their Fourier Analysis Resynthesis program will allow the user to manipulate the timbre of a morphed sound by using a joystick to move through a three-dimensional timbre space.

Here we study the morphing of Lemur files describing sounds that may contain an unequal number of features and nonharmonic components.

2 THE LEMUR REPRESENTATION

An implementation of the McAulay–Quatieri [16] sinusoidal technique was carried out by Maher and Beauchamp [17]. Members of the CERL Sound Group later extended the technique and created an Apple Macintosh implementation, Lemur [18]. The technique models sounds as sinusoids with time-varying amplitudes and frequencies, called partials. Partials have the ability to be "born" (when their magnitude exceeds a threshold) and to "die" (when their magnitude falls below the threshold). The number of partials present in a sound may vary over the course of the sound.

The output from the Lemur analysis is a file containing a sequential list of frames, each describing a small portion of the sound. Each frame contains a list of the amplitudes and frequencies of the partials present in that part of the sound.[1]

3 MORPHING ALGORITHM

We investigated the morphing of pitched sounds with vibrato. This discussion describes the morphing of two sounds. The principles discussed, however, can be applied to any number and proportions (or time-varying proportions) of original sounds.

3.1 Logarithmic Scaling

Musical tuning systems are generally based on frequency ratios rather than frequency differences—roughly duplicating the way in which pitch is perceived, although pitch perception is complex and depends of a variety of factors. A sound whose frequency is double the frequency of another sound will be perceived as a note one octave higher in pitch. If interpolation was done on a linear scale, the interpolated pitch would not be what was expected [19, pp. 28–31]. Consequently a logarithmic scale is used to calculate the interpolated frequency.

Loudness is perceived in a somewhat similar manner, although the perception of loudness is also complex and depends on the timbre and pitch of the sound as well as the magnitude. The perception of loudness is more nearly geometric than linear, however, so a logarithmic scale is also used for magnitude interpolation. The expression used to calculate an interpolated value x (either a frequency or a magnitude) from values x_1 and x_0, with an x_1 weight of w_1, is

$$x = 2^{w_1 \log2(x_1) + (1 - w_1) \log2(x_0)} .$$

3.2 Partial Selection

The first step in the morphing process is to determine which partials should be paired for morphing. Since the sounds are pitched, most of the partials are approximately integer multiples of the fundamental of that sound. Partials which are the same multiple of the fundamental in the two sounds should be morphed together. This is determined by looking for partials in each of the sounds in which the ratio of the analyzed frequency to that sound's fundamental frequency is approximately equal.

There may be a partial in one sound with no corres-

[1] The Lemur program and the sound examples used in this paper are available through anonymous ftp. For information, send email to Lemur@uiuc.edu.

ponding partial in the second sound. If this occurs, the existing partial is morphed with a zero-magnitude partial. The frequency of the zero-magnitude partial is determined by the ratio of the fundamentals and the existing frequency.

3.3 Frequency

The analysis process cannot determine the frequency of very low-amplitude partials accurately. Such partials are generally so quiet that they are inaudible in the original sound. A problem occurs, however, when a low-amplitude partial containing inaccurate frequency information is morphed with a high-amplitude partial with accurate frequency information. In this case the morphed partial will be a medium-amplitude partial with audibly inaccurate frequency information.

This problem is avoided by not relying exclusively on the analysis for frequency information. For very quiet partials the frequency of the nearest harmonic is used for interpolation. For sufficiently loud partials the frequency from the analysis is used in interpolation. For intermediate-amplitude partials the frequency used in interpolation is derived from both the analysis frequency and the frequency of the nearest harmonic,

$$\text{morph frequency} = \begin{cases} 2^{(1-m/t)\log_2(f_h)+(m/t)\log_2(f_a)}, & \text{if } m < t \\ f_a, & \text{otherwise} \end{cases}$$

where

m = magnitude from analysis
f_a = frequency from analysis
t = magnitude threshold
f_h = closest harmonic frequency.

As a partial gradually increases in amplitude, more of the frequency used in morphing is taken from the analysis. Consequently there is no abrupt change between a calculated frequency and an analyzed frequency when a partial reaches the threshold beyond which only the analyzed frequency is used.

3.4 Features

Considering sounds that have any number of features, we distinguish between two types of features—unique features and repeatable features.

3.4.1 Unique Features

Unique features are specific points in each sound that must be lined up in the morphing process. For instance, the sounds to be morphed may start at slightly different times. If the sounds are morphed without lining up the beginnings of the attacks, a quiet version of one sound will be heard before the second sound begins. Other examples of unique features are the peak of the attack, the start of the decay, the loudest point, and so on. Each of these points in the original sounds should be lined up algorithmically to create the morphed sound. Since unique features are lined up exactly in the two sounds, there must be the same number of unique features in both sounds.

3.4.2 Repeatable Features

Repeatable features are features that may be duplicated or omitted in the morphing process. They are features in one sound that do not necessarily correspond to a specific feature in the other sound. For instance, it is not necessary that the fifth vibrato cycle from one sound be interpolated with the fifth vibrato cycle in the second sound. It is important, however, that vibrato peaks in the two sounds match up in the interpolation, so the morphed sound has a single vibrato rate.

It is assumed that repeatable features may be skipped or repeated as necessary. This implies that the frequency and amplitude at the beginning of each repeatable feature should be approximately equal to the frequency and amplitude at the beginning of adjacent features.

3.4.3 Feature Numbering

Each unique feature is assigned a unique number. Similarly numbered unique features are always aligned in the morphing process. The frame containing unique feature number 5 in file 1, for instance, is always morphed with the frame containing unique feature number 5 in file 2.

The numbering of repeatable features begins again at 1 each time a unique feature is reached. Repeatable features between two unique features or between a unique feature and the end of the file are assigned unique, monotonically increasing numbers.

For example, if *u#* represents a unique feature and *r#* a repeatable feature, a file might be numbered as shown in Fig. 1. The unique features marked correspond to:

- u_1 beginning of attack
- u_2. peak of attack
- u_3 end of attack
- u_4 quietest point
- u_5 beginning of decay
- u_6 end of decay.

There are several vibrato cycles marked as repeatable features. The numbering of vibrato cycles between the quietest point and the beginning of the decay begins again at *r*1.

When this sound is morphed with another sound, some of the vibrato cycles between the end of the attack and the quietest point may be repeated or omitted, depending on the number of vibrato cycles between the end of the attack and the quietest point in the second sound. Similarly, vibrato cycles between the quietest point and the beginning of the decay may be repeated or omitted.

3.4.4 Step Rates

There are two rates to be considered when stepping through the Lemur files.

1) The rate at which to step through the current feature. This rate depends on the number of frames remaining in the current feature in the two files.

2) The rate at which to step from one feature to the next. This rate depends on the number of repeatable features remaining before the next unique feature in the two files.

Stepping within a Feature: When morphing, the number of frames remaining in the current morphed feature is calculated as a weighted average of the number of frames remaining in the current feature in the two original sounds. This number is recalculated for each frame so that a time-varying weight will result in a gradually changing feature length.

The partial envelopes for the two original sounds are stepped through at a rate such that the ends of the current feature in each sound are reached at the same time. The current frame number in each file as well as the current frame number in the morphed file must be considered.

The correct current frame number in each of the original files is computed by adding a floating-point frame increment to a floating-point current frame number. The rounded result is the number of the next frame to be morphed from that file. The frame increment is the ratio of the number of frames remaining in the current feature in this file to the number of frames remaining in the current feature in the morphed file. If the frame increment is less than 1, a frame may be repeated two or more times. If the frame increment is greater than 1, one or more frames may be skipped.

Stepping between Features: The same principle is applied at a larger scale when the end of a repeatable feature is reached. There are two possibilities in this case:

1) The next feature in one of the files is a unique feature. In this case the corresponding unique feature is located in the other file and morphing continues from that point.

2) The next feature in both files is a repeatable feature. In this case a calculation must be done to determine whether to skip or repeat features in each of the files.

For the second case the number of repeatable features in the morphed sound before the next unique feature or the end of the sound is calculated. This number is a weighted average of the number of repeatable features remaining in each of the original sounds before the next unique feature or the end of the sound. The correct current feature number in each of the original files is computed by adding a floating-point feature increment to a floating-point current feature number. The rounded result is the number of the next repeatable feature to be morphed from that file. The feature increment is the ratio of the number of repeatable features remaining before the next unique feature in this file to the number of repeatable features remaining in the next unique feature in the morphed file.

In a file with few repeatable features, a feature may be repeated two or more times. In a file with numerous repeatable features, one or more repeatable features may be skipped. In this case the next feature to be morphed may be repeatable or unique, depending on the number of repeatable features remaining before the next unique one. The current feature number is reset to 1 after each unique feature, as the numbering of repeatable features always begins again at 1 at these points.

There are no audible discontinuities in the sound for two reasons. Because of the restriction that adjacent repeatable features should have roughly similar frequency and amplitude characteristics, frames which are adjacent in the morphed sound are always fairly alike. In addition, because morphing is done in the frequency domain, changes in frequency or amplitude occur over the course of a frame. Consequently there are none of the audible clicks that might occur if portions of sample files were selected for omission or duplication.

3.5 Weights

The weight for each sound may vary as the morph progresses, as the increments are recomputed for each

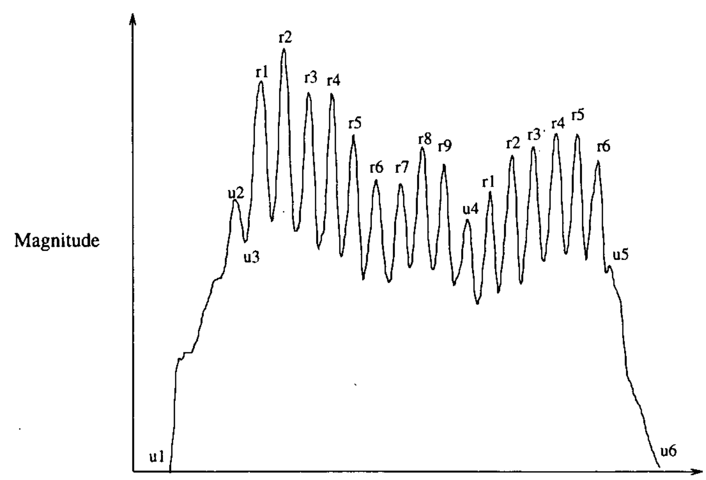


Fig. 1. Features in a sound.

frame. If the weight gradually changes when two sounds with different vibrato rates are morphed, the vibrato rate of the morphed sound will change gradually.

4 FUTURE WORK

The perception of a sound's loudness depends on its pitch as well as its energy. A high-pitched sound is perceived as being louder than a low sound with the same energy. An improved morphing algorithm would take this fact into consideration when morphing sounds of different pitches, rather than simply using logarithmic-scale-magnitude interpolation.

Repeatable features in two sounds are assumed to have the same shape. This may not be the case in actual sounds. Some vibrato cycles, for instance, may rapidly reach a peak and decay slowly, whereas others may slowly reach a peack and decay quickly. Such sounds cannot be handled correctly by an interpolation algorithm with only two types of features. The sounds described, for instance, would be morphed to form a new sound with two vibrato peaks in each vibrato cycle. An improved algorithm could provide for subfeatures within a repeatable features. Each repeatable feature might have a peak, for instance. These peaks would be aligned in the morphed sound so that each vibrato peak in the morphed sound has only one peak.

It is assumed that repetitive amplitude and frequency changes occur at the same rate. This may not be the case for some sounds. A more general interpolation algorithm might account for this so that amplitude and frequency variations may occur at differing rates.

5 SOUND EXAMPLES

5.1 Sounds Used

All of the examples described are morphs of natural instrument sounds. This is not because the algorithm is restricted to such sounds, but because it is easier to determine whether the algorithm is working reasonably by examining its use on natural instrument sounds.

A morph between a glass breaking and a dog barking, for instance, may produce an interesting sound. It would be difficult, however, to decide whether the sound produced is "correct." More listeners might agree, however, on what a morph between a clarinet and a violin should sound like. If the algorithm produces a sound that does not meet these expectations, it is easier to determine that the algorithm is functioning incorrectly.

5.2 Graphs

5.2.1 Types

There are two types of graphs. One is a typical Lemur graph, which shows time along the horizontal axis and frequency along the vertical axis, and uses different shades of gray to represent amplitude. This type of graph shows precise frequency information, but provides only a rough depiction of magnitude information.

The second is a three-dimensional graph which shows time on the x axis, frequency on the y axis, and magnitude on the z axis. These graphs give precise magnitude information but only rough frequency information. The frequency shown for each partial is the average frequency of the partial over its existence.

A complete picture of the amplitude and frequency envelopes of the sound can be obtained by looking at both graphs for a sound.

5.2.2 Views

Graphs that show all partials give a good picture of the spectrum. In these graphs, however, a low-frequency partial often obscures an adjacent partial, so it is difficult to get an accurate picture of low-frequency partials from a graph showing all partials. Consequently views of only the bottom few partials of each sound are also provided.

5.3 Sources

With the exception of the viola note, the sounds in this section were recorded from the McGill University Master Samples disk [20]. The viola sound was recorded at the University of Illinois. The graphs shown are of Lemur analyses of these sounds.

5.4 Control Files

Two different weight functions are used in the morphs shown:

1) *A ramp function*: The weight of the first sound goes from 0 to 100% over the course of the morph.

2) *A constant function*: A constant weight of 50% of each sound is used throughout the morph.

5.5 Endpoint Sounds

The graphs of the string instruments show the magnitude varying in an irregular way within vibrato cycles. We attribute this to the variations in the mechanical system as different parts of the player's finger come in contact with the string. Another contributing factor is the sympathetic vibrations of neighboring strings which affect the magnitude of a partial, depending on its frequency.

When a sound with an irregularly shaped vibrato is morphed with a sound with a regularly shaped vibrato, the resulting waveform is a compromise between the two. It will contain some of the irregularities of the first sound, but these will be less pronounced as a result of being averaged with the regularly shaped vibrato.

5.5.1 Violin

Figs. 2 through 5 depict a bowed F above middle C (350 Hz) with vibrato on a violin [20, p. 54]. Features used in morphing are shown in Table 1.

5.5.2 Viola

Figs. 6 through 9 depict a bowed E flat above middle C (330 Hz) with vibrato on a viola. Features used in morphing are shown in Table 2.

5.5.3 Bach Trumpet

Figs. 10 through 12 depict a D above middle C (293 Hz) on a trumpet [20, p. 82]. The trumpet sound is noteworthy as the fifth partial has a higher amplitude than all other partials, including the fundamental. Features used in morphing are shown in Table 3.

5.5.4 Clarinet

Figs. 13 through 15 depict a middle C (262 Hz) on a clarinet [20, p. 77]. The clarinet sound is distinctive because the odd-numbered partials have a much greater magnitude than the even ones. Features used in morphing are shown in Table 4.

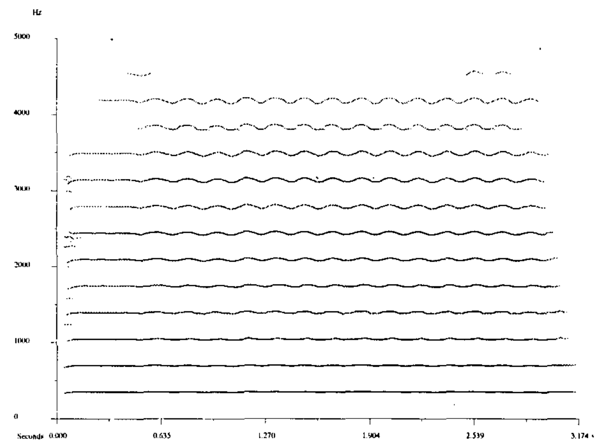


Fig. 2. Frequency graph of all partials of a violin sound.

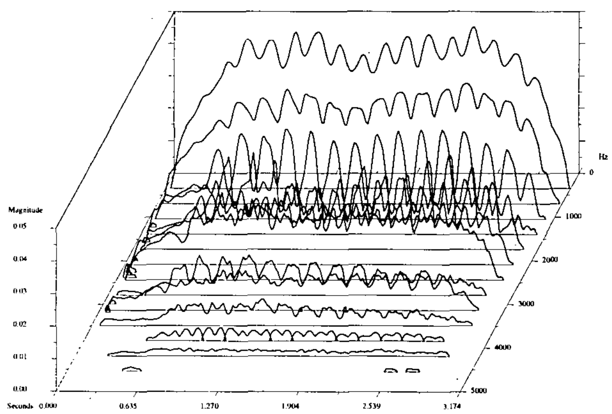


Fig. 3. Magnitude graph of all partials of a violin sound.

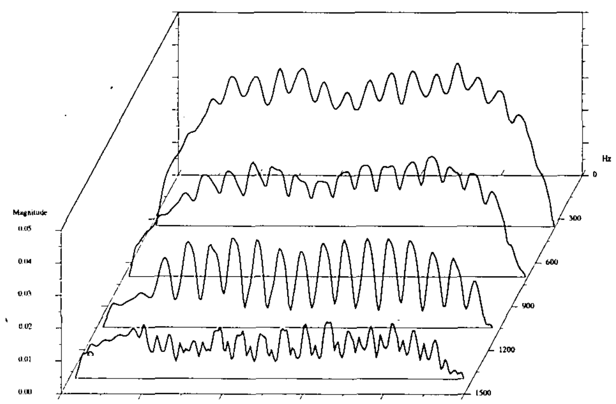


Fig. 4. Magnitude graph of four lowest partials of a violin sound.

5.6 Morphed Sounds

5.6.1 Violin and Viola Sounds with Differing Vibrato Rates

This section depicts two morphs of the violin and viola sounds with differing vibrato rates. The first is a morph with no attempt to match vibrato rates. The second is a morph that considers vibrato rates. A constant 50% ratio was used for each morph.

No Vibrato Features. Figs. 16 through 19 depict a morph between the violin and viola sounds. No repeatable features are marked in either of the sounds. The resulting sound does not have a regular vibrato as vibrato peaks from the original sounds either add or cancel in the morphed sound.

Vibrato Features. Figs. 20 through 23 depict a morph between the violin and viola sounds. Vibrato peaks are marked as repeatable features in each of the sounds. The resulting vibrato of the sound is similar to the vibratos of the two original sounds, but with intermediate speed and width.

5.6.2 Clarinet and Bach Trumpet

Figs. 24 through 26 depict a morph between the trumpet note and the clarinet note. The ramp control file was

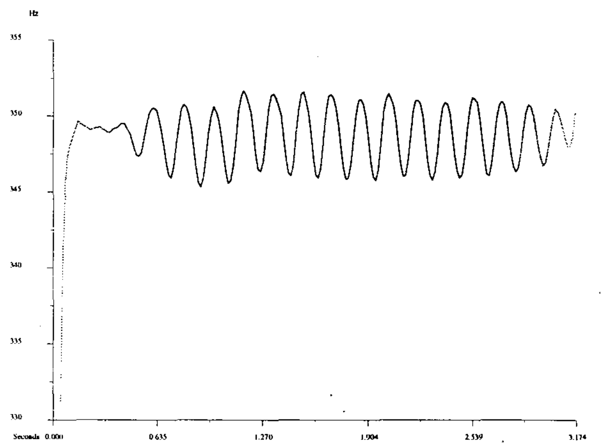


Fig. 5. Frequency graph of the fundamental of a violin sound.

Table 1. Violin features.

Frame Number	Time (s)	Description
13	0.039	Start of attack
53	0.159	Attack peak
206	0.618	End of attack
271	0.813	Vibrato peak
326	0.978	Vibrato peak
387	1.161	Vibrato peak
442	1.326	Vibrato peak
506	1.518	Vibrato peak
568	1.704	Vibrato peak
626	1.878	Vibrato peak
684	2.052	Vibrato peak
739	2.217	Vibrato peak
794	2.382	Vibrato peak
856	2.568	Vibrato peak
911	2.733	Start of decay
964	2.892	Vibrato peak
1016	3.048	Vibrato peak
1055	3.165	End of decay

used. The odd-numbered partials increase in intensity as the clarinet contributes a larger percentage of the sound. Since the clarinet note has a lower fundamental frequency than the trumpet note, the frequency graphs show a glissando as the fundamental frequency descends one whole step.

6 CONCLUSION

We have described a timbre morphing algorithm for pitched sounds which may contain an unequal number of features. The primary features of the algorithm follow.

- The number of partials present at any time in the original sounds may vary over time.
- At any instant, the sounds may contain an unequal number of partials.
- The sounds may contain nonharmonic partials.
- The sounds may have differing fundamental frequencies.
- The sounds may have an unequal number of features. For example, they may have differing vibratos. In this case the morphed sound has a vibrato intermediate between the vibratos of the two original sounds.

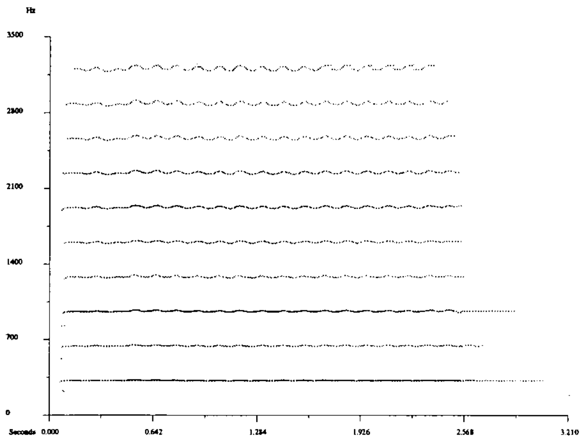


Fig. 6. Frequency graph of all partials of a viola sound.

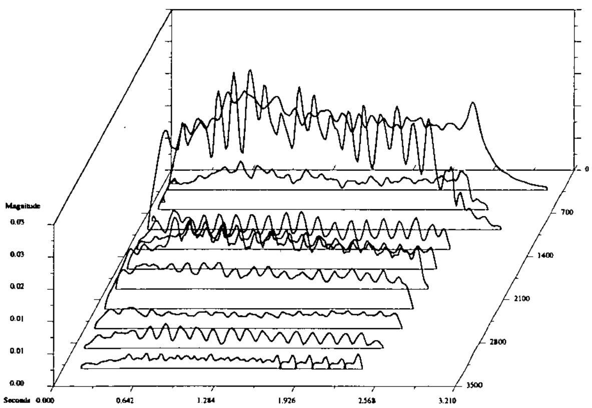


Fig. 7. Magnitude graph of all partials of a viola sound.

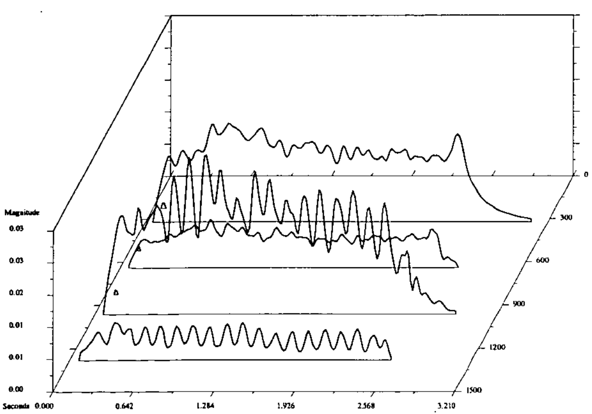


Fig. 8. Magnitude graph of four lowest partials of a viola sound.

7 REFERENCES

[1] L. Haken, R. Abdullah, and M. Smart, "The Continuum: A Continuous Music Keyboard," in *Proc. 1992 Int. Computer Music Conf.* (San Jose, CA), International Computer Music Association, San Francisco, CA, pp.

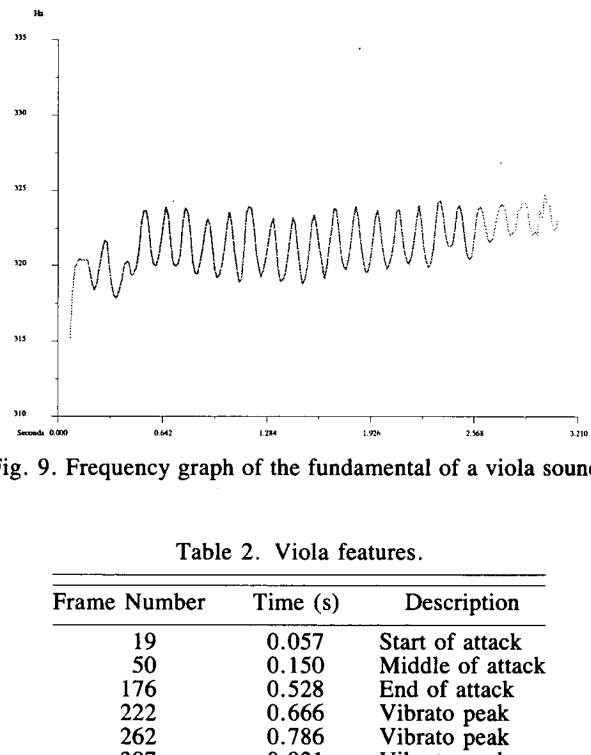


Fig. 9. Frequency graph of the fundamental of a viola sound.

Table 2. Viola features.

Frame Number	Time (s)	Description
19	0.057	Start of attack
50	0.150	Middle of attack
176	0.528	End of attack
222	0.666	Vibrato peak
262	0.786	Vibrato peak
307	0.921	Vibrato peak
350	1.050	Vibrato peak
390	1.170	Vibrato peak
442	1.326	Vibrato peak
482	1.446	Vibrato peak
525	1,575	Vibrato peak
565	1.695	Vibrato peak
608	1.824	Vibrato peak
654	1.962	Vibrato peak
694	2.082	Vibrato peak
739	2.217	Vibrato peak
779	2.337	Vibrato peak
828	2.484	Start of decay
863	2.589	Vibrato peak
907	2.721	Vibrato peak
953	2.859	Vibrato peak
1000	3.000	Vibrato peak
1036	3.108	End of decay

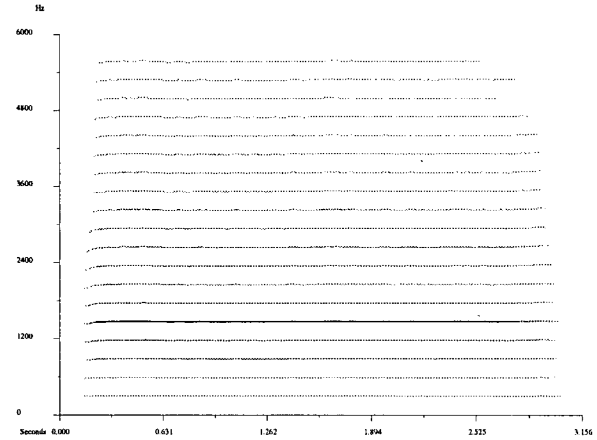


Fig. 10. Frequency graph of all partials of a trumpet sound.

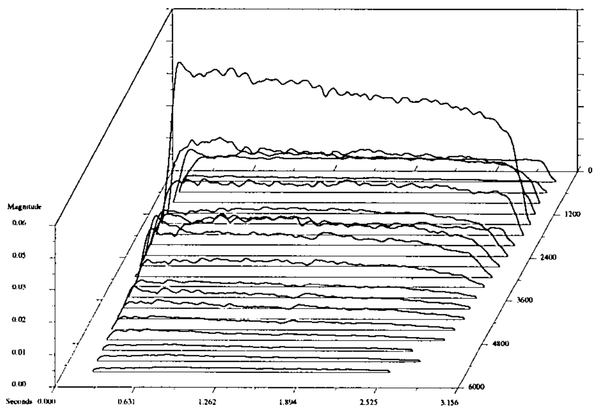


Fig. 11. Magnitude graph of all partials of a trumpet sound.

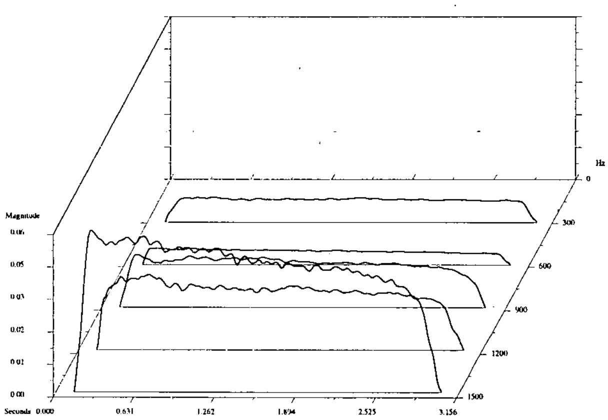


Fig. 12. Magnitude graph of five lowest partials of a trumpet sound.

Table 3. Trumpet features.

Frame Number	Time (s)	Description
49	0.147	Start of attack
172	0.516	End of attack
974	2.922	Start of decay
1012	3.036	End of decay

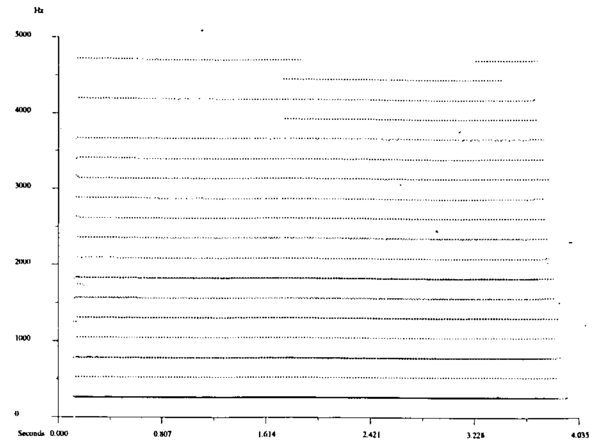


Fig. 13. Frequency graph of all partials of a clarinet sound.

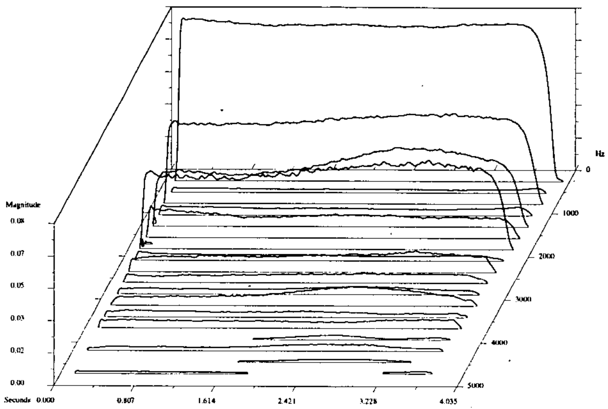


Fig. 14. Magnitude graph of all partials of a clarinet sound.

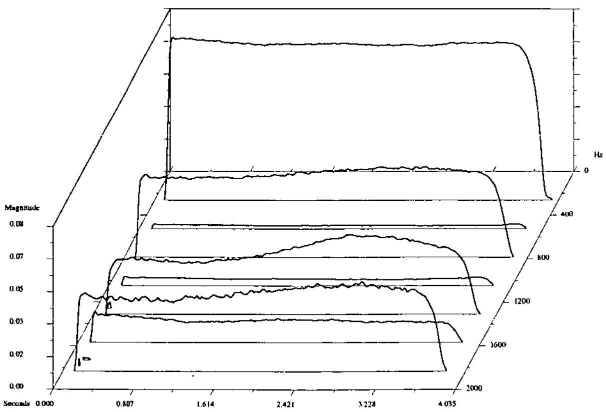


Fig. 15. Magnitude graph of seven lowest partials of a clarinet sound.

Table 4. Clarinet features.

Frame Number	Time (s)	Description
32	0.096	Start of attack
63	0.181	End of attack
1283	3.849	Start of decay
1343	4.029	End of decay

81–84.

[2] J. Strawn, "Orchestral Instruments: Analysis of Performed Transitions," *J. Audio Eng. Soc.*, vol. 34, pp. 867–880 (1986 Nov.).

[3] J. Strawn, "Analysis and Synthesis of Musical Transitions Using the Discrete Short-Time Fourier Transform," *J. Audio Eng. Soc.*, vol. 35, pp. 3–14 (1987 Jan./Feb.).

[4] B. Holloway and L. Haken, "A Sinusoidal Synthesis Algorithm for Generating Transitions between Notes," in *Proc. 1992 Int. Computer Music Conf.* (San Jose, CA, 1992 Oct.), International Computer Music Association, San Francisco, CA, pp. 14–17.

[5] K. W. Berger, "Some Factors in the Recognition of Timbre," *J. Acoust. Soc. Am.*, vol. 36, pp. 1888–1891 (1964 Oct.).

[6] E. L. Saldanha and J. F. Corso, "Timbre Cues for the Recognition of Musical Instruments," *J. Acoust. Soc. Am.*, vol. 36, pp. 2021–2026 (1964 Nov.).

[7] J. M. Grey, "An Exploration of Musical Timbre," Ph.D. thesis, Rep. Stan-M-2, Dept. of Music, Stanford University, Stanford, CA (1975).

[8] K. W. Schindler, "Dynamic Timbre Control for Real-Time Digital Synthesis," *Computer Music J.*, vol. 8, pp. 28–42 (1984).

[9] L. Haken, "Real-Time Fourier Synthesis of Ensembles with Timbral Interpolation," Ph.D. thesis, Dept. of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL (1989).

[10] L. Haken, "Computational Methods for Real-Time Fourier Synthesis," *IEEE Trans. Signal Process.*, vol. 40, pp. 2327–2329 (1992 Sept.).

[11] T. L. Peterson, "Vocal Tract Modulation of Instrumental Sounds by Digital Filtering," in *Proc. 2nd Ann. Music Computation Conf.* (1975), pp. 33–41.

[12] P. Depalle, G. Poirot, and X. Rodet, "A Virtual Castrato (!?)," in *Proc. Int. Computer Music Conf.* (1994), pp. 357–360.

[13] P. Depalle and G. Poirot, "Svp: A Modular System for Analysis, Processing and Synthesis of Sound Signals," in *Proc. Int. Computer Music Conf.* (1991), pp. 161–164.

[14] C. Roads, Ed., *Composers and the Computer* (MIT Press, Cambridge, MA, 1985).

[15] K. McMillen, "A Technology Dossier on Fourier

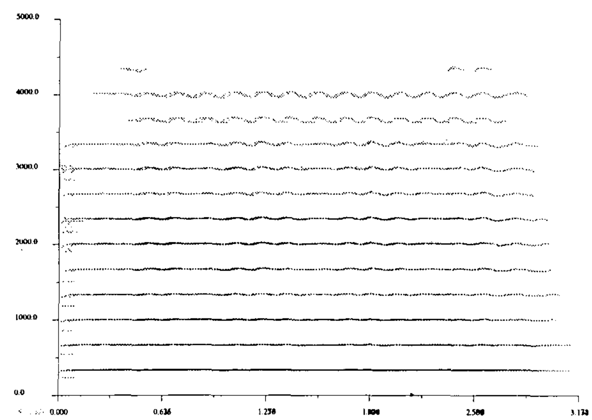


Fig. 16. Frequency graph of all partials of a violin with slow vibrato morphed with a viola with fast vibrato. No vibrato features.

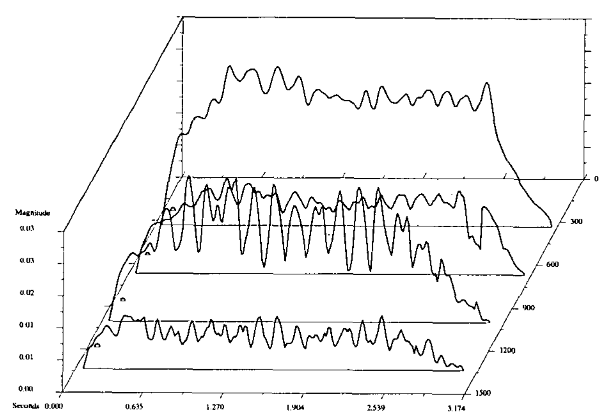


Fig. 18. Magnitude graph of four lowest partials of a violin with slow vibrato morphed with a viola with fast vibrato. No vibrato features.

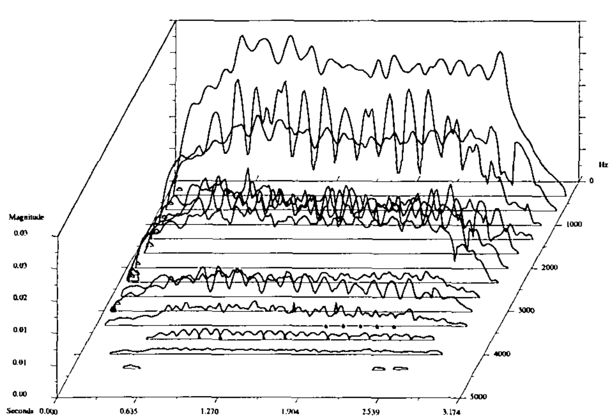


Fig. 17. Magnitude graph of all partials of a violin with slow vibrato morphed with a viola with fast vibrato. No Vibrato features.

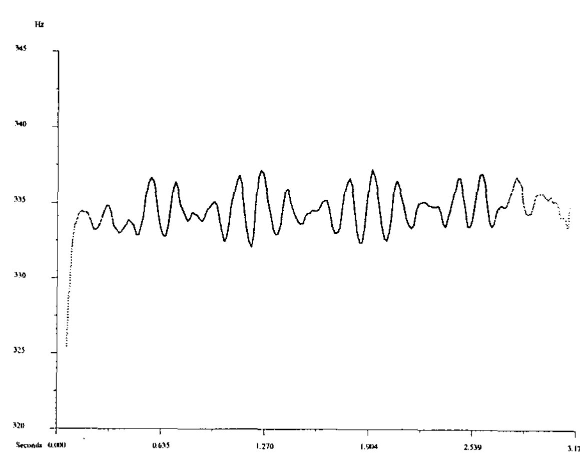


Fig. 19. Frequency graph of the fundamental of a violin with slow vibrato morphed with a viola with fast vibrato. No vibrato features.

Analysis Resynthesis," Oberheim Digital and G-WIZ Labs, Berkeley, CA. (1994).

[16] R. J. McAulay and T. F. Quatieri, "Speech Analysis/Synthesis Based on a Sinusoidal Representation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-34, pp. 744–754 (1986).

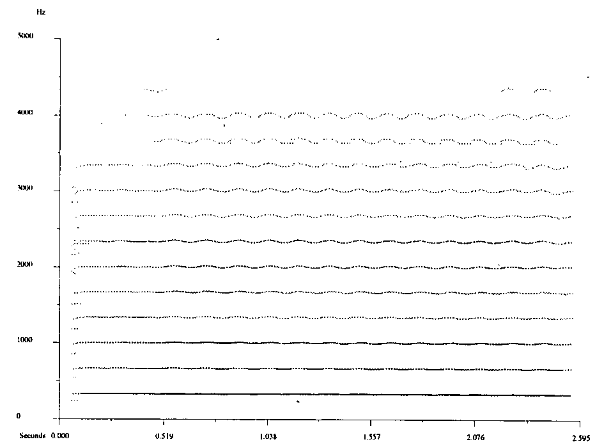


Fig. 20. Frequency graph of all partials of a violin with slow vibrato morphed with a viola with fast vibrato. Vibrato features.

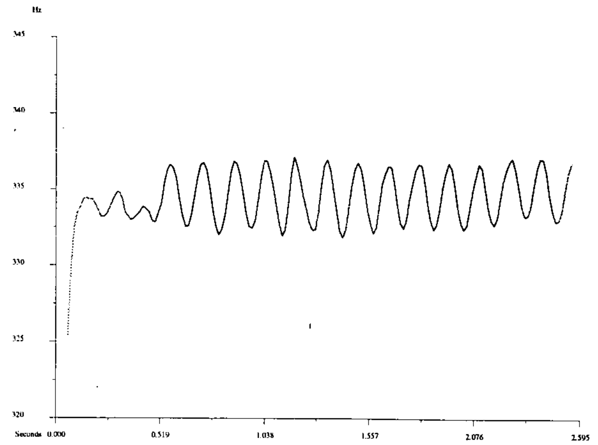


Fig. 23. Frequency graph of the fundamental of a violin with slow vibrato morphed with a viola with fast vibrato. Vibrato features.

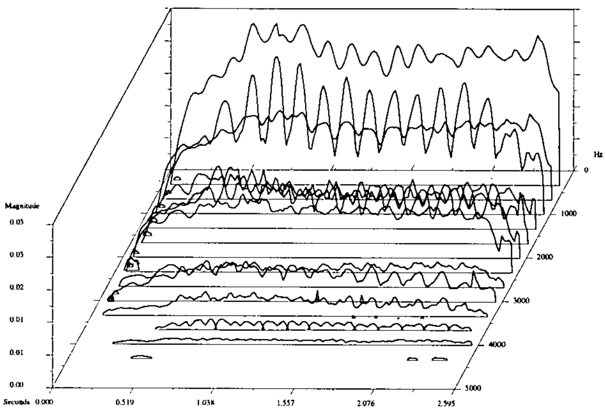


Fig. 21. Magnitude graph of all partials of a violin with slow vibrato morphed with a viola with fast vibrato. Vibrato features.

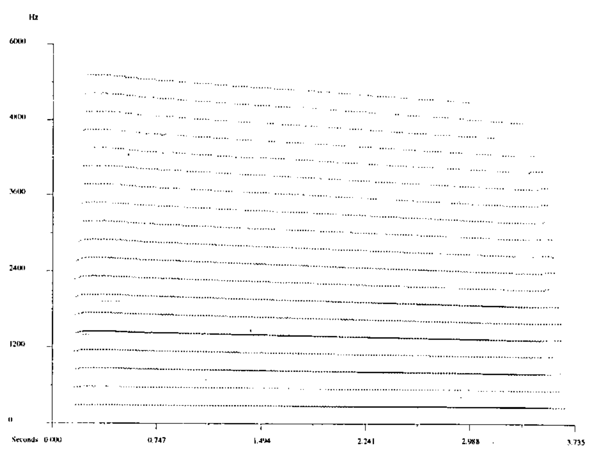


Fig. 24. Frequency graph of all partials of a trumpet note morphed with a clarinet note. Graph shows a glissando as the fundamental frequency changes from 293 to 262 Hz.

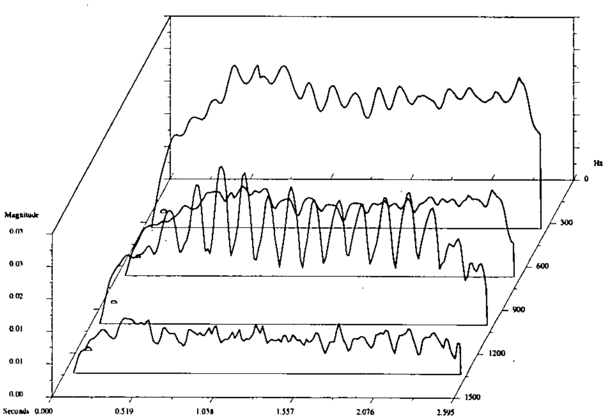


Fig. 22. Magnitude graph of four lowest partials of a violin with slow vibrato morphed with a viola with fast vibrato. Vibrato features.

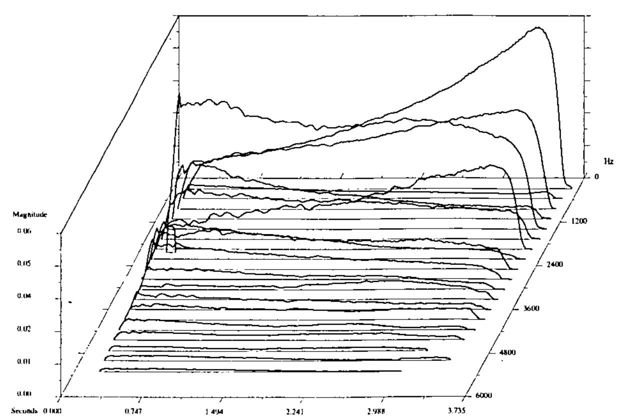


Fig. 25. Magnitude graph of all partials of a trumpet note morphed with a clarinet note. Odd-numbered partials increase in intensity as clarinet contributes a larger percentage of sound.

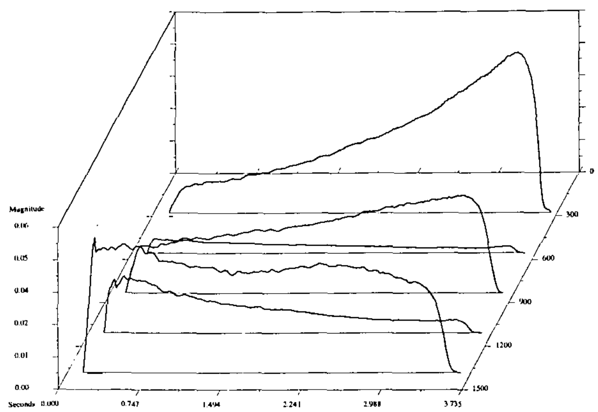


Fig. 26. Magnitude graph of five lowest partials of a trumpet note morphed with a clarinet note. Odd-numbered partials increase in intensity as clarinet contributes a larger percentage of sound.

[17] R. Maher, "An Approach of the Separation of Voices in Composite Musical Signals," Ph.D. thesis, Dept. of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL (1989).

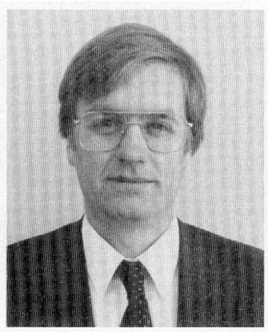
[18] K. Fitz, W. Walker, and L. Haken, "Extending the McAulay–Quatieri Analysis for Synthesis with a Limited Number of Oscillators," in *Proc. 1992 Int. Computer Music Conf.* (San Jose, CA), International Computer Music Association, San Francisco, CA, pp. 381–382.

[19] P. Christensen, "User Interfaces for Audio Signal Generation," Master's thesis, Dept. of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL (1994).

[20] F. Opolko and J. Wapnick, *McGill University Master Samples User's Manual* (McGill University, Montreal, Que., Canada, 1989).

THE AUTHORS

E. Tellman



L. Haken

B. Holloway

Edwin Tellman holds bachelor's degrees in cello performance from the University of Arizona and computer science from the University of Washington. He recently completed his master's degree in computer science at the University of Illinois. He is currently working as a software engineer in the Seattle area.

•

Lippold Haken is leader of the CERL Sound Group at the University of Illinois, with research interests in computer music, computer architecture, and user interface. He has developed real-time audio signal processing hardware and software, and is coauthor of a sophisticated music notation editor, Lime. He is also leader of the hardware design group for a high-speed mainframe computer that provides service for thousands of simultaneous users on NovaNET, a real-time nationwide network. Professor Haken developed a new project-oriented course that introduces the major areas of electrical and computer engineering to freshmen.

•

Bryan Holloway is a doctoral student in electrical engineering at the University of Illinois. He is a member of the CERL Sound Group and the NCSA Audio Development Group. His interests include time-frequency representation of audio signals, timbre morphing (speech, musical instruments, and other signals), audio compression techniques, and network applications involving sound. In his spare time, he is president of an Internet Service Provider in Champaign, IL. (http://www.shout.net)