Calculation of a constant Q spectral transform

Judith C. Brown

Media Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139 and Physics Departments, Wellesley College, Wellesley, Massachusetts 02181

(Received 28 December 1988; revised 12 February 1990; accepted 10 September 1990)

The frequencies that have been chosen to make up the scale of Western music are geometrically spaced. Thus the discrete Fourier transform (DFT), although extremely efficient in the fast Fourier transform implementation, yields components which do not map efficiently to musical frequencies. This is because the frequency components calculated with the DFT are separated by a constant frequency difference and with a constant resolution. A calculation similar to a discrete Fourier transform but with a constant ratio of center frequency to resolution has been made; this is a constant Q transform and is equivalent to a 1/24-oct filter bank. Thus there are two frequency components for each musical note so that two adjacent notes in the musical scale played simultaneously can be resolved anywhere in the musical frequency range. This transform against log (frequency) to obtain a constant pattern in the frequency domain for sounds with harmonic frequency components has been plotted. This is compared to the conventional DFT that yields a constant spacing between frequency components. In addition to advantages for resolution, representation with a constant pattern has the advantage that note identification ("note identification" rather than the term "pitch tracking," which is widely used in the signal processing community, is being used since the editor has correctly pointed out that "pitch" should be reserved for a perceptual context), instrument recognition, and signal separation can be done elegantly by a straightforward pattern recognition algorithm.

PACS numbers: 43.75.Bc, 43.75.Cd, 43.60.Lq

INTRODUCTION

The present work is based on the property that, for sounds made up of harmonic frequency components, the positions of these frequency components relative to each other are the same independent of fundamental frequency if they are plotted against log frequency. An example of this property is found in Fig. 1, which is a plot of a hypothetical spectrum with equal amplitude frequency components f, $2f$, $3f$,... and so on. The spacing between the first two harmonics is log (2), that between the second and third harmonics is log(3/2), and so forth. That is, the absolute positions depend on the frequency of the fundamental, but the relative positions are constant. Thus these spectral components form a "pattern" in the frequency domain, and this pattern is the same for all sounds with harmonic frequency components. Differences will, of course, be manifested in the amplitudes of the components despite their fixed relative positions; these reflect differences in timbre of the sound analyzed.

The conventional linear frequency representation given by the discrete Fourier transform gives rise to a constant separation between components for musical sounds consisting of harmonic components. This is the dominant feature in the pattern produced, and both the separation constant and the overall position of this pattern vary with fundamental frequency. The result is that it is more difficult to pick out differences in other features of the sound, such as timbre and attack and decay.

The log frequency representation, on the other hand, gives a constant pattern for the spectral components, and thus, the problem of instrument identification or of fundamental frequency identification becomes a straightforward problem of recognizing a previously determined pattern. In addition to its practical advantages, this idea has theoretical appeal for its similarity to modern theories of pitch perception based on pattern, recognition.[1] In one of these theories, the perception of the pitch of a sound with a missing fundamental is explained by the "pattern" formed by the remaining harmonics on the basilar membrane. Similarly, we have devised a computer algorithm that recognizes the pattern made by these harmonics in the log frequency domain; it can thus identify the frequency as that of the fundamental even in those cases where there is no spectral energy at the frequency of the fundamental.

To demonstrate this "constant pattern" for a variety of musical sounds, we first tried to utilize the speed and efficiency of the fast Fourier transform algorithm and then plot the data against log(f). It soon became clear that the mapping of these data from the linear to the logarithmic domain gave too little information at low frequencies (data from a few linear points mapping to a large number of logarithmic points) and too much information at high frequencies. Even more problematic were resolution considerations. The discrete short-time Fourier transform gives a constant resolution for each bin or frequency sampled equal to the sampling rate divided by the window size in samples. This means, for example, if we take a window of 1024 samples with a sampling rate of 32 000 samples/s (reasonable for musical sig-

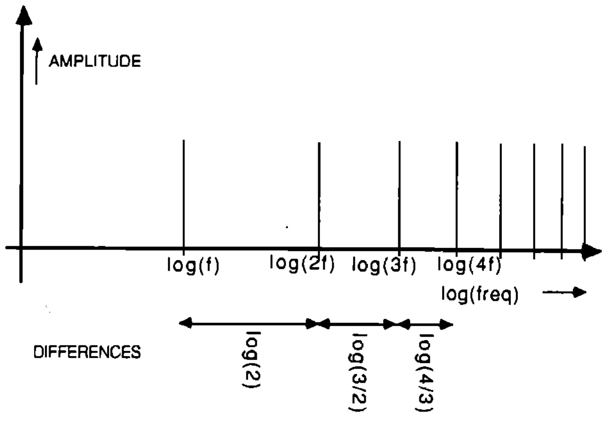

FIG. 1. Pattern of Fourier transform of harmonic frequency components plotted against log(frequency).

nals), the resolution is 31.3 Hz. At the low end of the range for a violin, the frequency of G_3 is 196 Hz so this resolution is 16% of the frequency.

This is much greater than the 6% frequency separation for two adjacent notes tuned in equal temperament. At the upper end of the piano range, the frequency of C_8 is 4186 Hz, and 31.3 Hz is equal to 0.7% of the center frequency. Thus at this end, we are calculating far more frequency samples than are needed.

It is thus clear that for musical applications the use of the conventional Fourier transform is inefficient. What is needed is information about the spectral components produced across the wide frequency range of a particular musical instrument. The resolution should be geometrically related to the frequency, e.g., 3% of the frequency in order to distinguish between frequencies with semitone (6%) spacing. Thus the frequencies sampled by the discrete Fourier transform should be exponentially spaced and, if we require quartertone spacing, this gives a variable resolution of at most $(2^{1/24} - 1) \simeq 0.03$ times the frequency. This means a constant ratio of frequency to resolution, $f/\delta f = Q$, or a constant Q transform. Here, $Q = f/0.029f = 34$ and the transform is equivalent to a 1/24-oct filter bank.

In Sec. II, we describe a particularly straightforward means of calculating a constant Q transform starting from the discrete Fourier transform. Following this section, we show results of this calculation on sounds produced by a violin, piano, and flute. These sounds consist of harmonic frequency components and demonstrate a constant pattern in the log frequency domain as predicted. The conventional discrete Fourier transform is included for comparison in two cases. In a subsequent article, we will present results for these musical instruments using a note identification system based on pattern recognition.

I. BACKGROUND FOR CALCULATION

The constant Q transform in our implementation is equivalent to a 1/24th-oct bank of filters. The constant Q filter bank and its similarity to the auditory system has been explored in two recent theses[2,3] that reference previous work extensively. The article by Higgins[4] is recommended as a background discussion of sampling effects in the calculation of the discrete Fourier transform for those wishing to review the techniques of digital signal processing. The theory of the short-time Fourier Transform was originally developed by Schroeder and Atal.[5] More recently, it has been extensively reviewed by Nawob and Quatieri in an excellent article.[6]

Various schemes for implementing constant Q spectral analysis outside a musical context have been published.[7-11] Gambaradella[12,13] demonstrates equivalence of the constant Q transform to the Mellin transform[14] and the existence of the inverse transform. This is of importance if manipulation of the signal in the spectral domain followed by transformation back to the time domain is desired. Most recently Teaney et al.[15] have calculated a "tempered Fourier transform" using four A-to-D conversions. They then exploit the "perfect" ratios for the musical intervals of an octave, fourth, and fifth to further reduce the complexity of the calculation.

Music researchers at the Center for Computer Research in Music and Acoustics (CCRMA)[16] at Stanford have used a "Bounded Q" Transform similar to that of Harris.[8] They calculate a fast transform and discard frequency samples except for the top octave. They then filter, downsample by a factor of 2, and calculate another FFT with the same number of points as before, which gives twice the previous resolution. From this they keep the second highest octave. The procedure is repeated until they arrive at the lowest octave desired. The advantage of this method is that they have the speed of the FFT, with variable frequency and time resolution and are thus able to optimize information for both frequency and time.

Kronland–Martinet[17] and others have employed a "wavelet transform" for musical analysis and synthesis. This is a constant Q method similar to the Fourier transform and to this method but based on a theoretical treatment for the use of so-called "wavelets" as generalized basis functions. Their method has been successful as a compositional tool where the transform is altered to obtain effects in the time domain when the inverse transform is taken. However, this method does not have sufficient resolution to be used for note identification.

The present method, described in detail in the following section, has two advantages over these other methods. The first is its simplicity; the second is that it is calculated for frequencies that are exponentially spaced with two frequency components per musical note. Thus it supplies exactly the information that is needed for musical analysis with sufficient resolution to distinguish adjacent musical notes. Further, a sound with harmonic frequency components will give rise to a constant pattern in the log frequency domain.

II. CALCULATION

For musical analysis, we would like frequency components corresponding to quarter-tone spacing of the equal tempered scale. The frequency of the kth spectral component is thus

$$f_k = (2^{1/24})^k f_{\min}, \qquad (1)$$

where f will vary from $f_{\min}$ to an upper frequency chosen to be below the Nyquist frequency. The minimum frequency $f_{\min}$ can be chosen to be the lowest frequency about which information is desired, e.g. a frequency just below that of the G string for calculations on sound produced by a violin. The resolution or bandwidth δf for the discrete Fourier transform is equal to the sampling rate divided by the window size (the number of samples analyzed in the time domain). In order for the ratio of frequency to bandwidth to be a constant (constant Q), then the window size must vary inversely with frequency.

More precisely, for quarter-tone resolution, we require

$$Q = f/\delta f = f/0.029f = 34, \qquad (2)$$

where the quality factor Q is defined as $f/\delta f$. We note that the bandwidth $\delta f = f/Q$. With a sampling rate $S = 1/T$ where T is the sample time, the length of the window in samples at frequency f_k,

$$N[k] = S/\delta f_k = (S/f_k)Q. \qquad (3)$$

Note also from this equation that the window contains Q complete cycles for each frequency f_k, since the period in samples is S/f_k. This makes sense physically since, in order to distinguish between f_{k+1} and f_k when their ratio is, e.g., $2^{1/24} \simeq 34/33$, we must look at at least 33 cycles. It is also interesting for comparison to consider the conventional discrete Fourier transform in terms of the quality factor $Q = f/\delta f$. We find that $f/\delta f$ is equal to the number of the coefficient, k, and this is, of course, the number of periods in the fixed window for that frequency.

We obtain an expression for the kth spectral component for the constant Q transform by considering the corresponding component for the discrete short time Fourier transform:[18]

$$X[k] = \sum_{n=0}^{N-1} W[n]\, x[n] \exp\{-j2\pi kn/N\}. \qquad (4)$$

Here, $x[n]$ is the nth sample of the digitized temporal function being analyzed. The digital frequency is $2\pi k/N$. The period in samples is N/k and the number of cycles analyzed is equal to k. Here $W[n]$ gives the shape of the window and will be discussed below.

For the constant Q transform, taking account of the constraints of Eqs. (1)–(3), the digital frequency of the kth component is $2\pi Q/N[k]$. The window function has the same shape for each component, but its length is determined by $N[k]$ so it is a function of k as well as n. We must also normalize by dividing the sum by $N[k]$ since the number of terms varies with k. Equation (4) thus becomes

$$X[k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} W[k,n]\, x[n] \exp\{-j2\pi Qn/N[k]\}. \qquad (5)$$

Here, the period in samples is $N[k]/Q$, and we always analyze Q cycles. A comparison of variables appearing in the calculation of the constant Q and the conventional Fourier transforms is given in Table I.

TABLE I. Comparison of variables in calculation of discrete Fourier transform (DFT) and of constant Q transform.

	Constant Q	DFT
Frequency	$(2^{1/24})^k \cdot f\mathrm{min}$ exponential in k	$k\,\Delta f$ linear in k
Window	variable = $\mathrm{N}[k] = \dfrac{SR \cdot Q}{f_k}$	constant = N
Resolution		
Δf	variable = f_k/Q	constant = SR/N
$\dfrac{f_k}{\Delta f_k}$	constant = Q	variable = k
Cycles in Window	constant = Q	variable = k

In practice, Eq. (5) is used as the basis for our calculations with $N[k] = N_{\max}/(2^{1/24})^k$. $N_{\max}$ is Q times the period of the lowest analysis frequency in samples. The Nyquist condition becomes $2\pi Q/N[k] < \pi$, which means $N[k] > 2Q$. This is identical to the usual statement that there must be at least two samples per period to avoid aliasing.

If the window function $W[k,n]$ is set equal to one over the interval $(0, N[k]-1)$, this corresponds to using a rectangular window.[18] This window can be shown to have maximum spill over into adjacent frequency bins.[19] We have accordingly used a Hamming window that has the form,

$$W[k,n] = \alpha + (1-\alpha)\cos(2\pi n/N[k]),$$

where $\alpha = 25/46$ and $0 \leqslant n \leqslant N[k] - 1$.

A number of initial calculations were made with frequencies corresponding to those of the equal tempered scales and with a Q of 17 corresponding to a resolution of a semitone. This resolution is insufficient to distinguish between adjacent frequency components particularly for the higher harmonics where ratios of frequencies of adjacent components approach 1. We then chose a Q of 34 corresponding to quarter-tone spacing as indicated in the preceding equations. This was still insufficient to resolve very high harmonics such as those in the violin spectrum, so Q was doubled to 68 for frequencies corresponding to G_6 (1568 Hz) and over. Since the time windows are quite short for these high frequencies, this did not add appreciably to the calculation time.

Equation (5) was calculated using a storage buffer containing the values of $W[n,k]\cos(2\pi Qn/N[k])$ and $W[n,k]\sin(2\pi Qn/N[k])$ for $k = 1$ to 156. From Eq. (1) these k values correspond to frequencies of 174.5 Hz to the Nyquist frequency.

We should note that the constant Q transform as calculated in Eq. (5) is not invertible for the following reasons.[6] First, the temporal decimation factor (the number of samples between calculations) is greater than the analysis window length for the high-frequency bins. This means that there are some samples that are never analyzed for the higher

TABLE II. Window length in samples (for a sampling rate of 32 000 samples/s) and in ms as a function of analysis frequency.

Channel	Midinote	Frequency (Hz)	Window (Samples)	(ms)
0	53	175	6231	195
6	56	208	5239	164
12	59	247	4406	138
18	62	294	3705	116
24	65	349	3115	97
30	68	415	2619	82
36	71	494	2203	69
42	74	587	1852	58
48	77	699	1557	49
54	80	831	1309	41
60	83	988	1101	34
66	86	1175	926	29
72	89	1398	778	24
78	92	1664	1308	41
84	95	1978	1100	34
90	98	2350	926	29
96	101	2797	778	24
102	104	3327	654	20
108	107	3956	550	17
114	110	4710	462	14
120	113	5608	388	12
126	116	6675	326	10
132	119	7942	274	9
138	122	9461	230	7
144	125	11 216	194	6
150	128	13 432	162	5

frequencies. Second, the bandwidth is less than the frequency sampling interval for the bins where $Q = 68$. The latter was not considered a problem since one of the real advantages of this method is that the analysis center frequencies are "tuned" to the frequencies of the source.

If computing time is an important consideration, the algorithm can be modified to low-pass filter at digital frequency $\pi/2$ and downsample by a factor of two after each octave.[8] If filters were chosen requiring, for example, 7 multiplies per output point, this would result in a saving in computation time of about a factor of 5. A large amount of space in RAM (random access memory) should also be gained by this method as the numbers in the storage buffer would be the same for each octave.

The number of multiplies in our method is roughly the same as for a 512-point discrete Fourier Transform yielding 256 real points in the frequency domain.This method gives much more useful information for frequencies varying over a wide range. Finally, if the current trend toward parallel processing machines is realized, the downsampled version of the algorithm can be implemented in real time with calculations for each of the center frequencies being carried out in parallel by 156 processors.

III. RESULTS

All calculations were programmed in C and run on a Hewlett Packard Model 9000 Series 300 "Bobcat" Computer. For those interested, the code can be obtained from brown@ems.media.mit.edu on the arpanet. Examples of sounds of musical instruments were digitized from live performances in the Music and Cognition Group at Massachusetts Institute of Technology. Other examples were generated using Barry Vercoe's Csound software. The calculation is carried out every 500 samples corresponding to about 15 ms at a sampling rate of 32 000 samples/s, but it should be recalled from Eq. (3) that different frequencies are analyzed over different time periods. Examples of the analysis windows

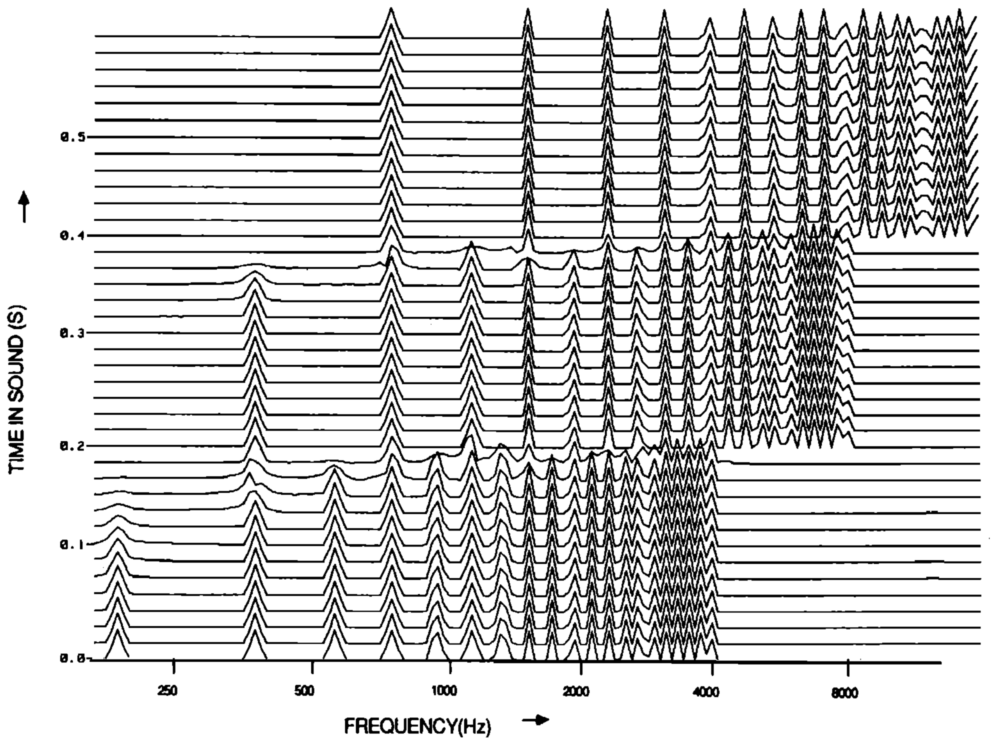

FIG. 2. Constant Q transform of three complex sounds with fundamentals G_3 (196 Hz), G_4 (392 Hz), and G_5 (784 Hz), and each having 20 harmonics with equal amplitude.

and corresponding times are given in Table II as a function of center frequency. Recall that the Q was increased with a corresponding increase in window size for frequencies over that of G_6 (1568 Hz).

Most of the figures are plots of the constant Q transform on the vertical axis against logarithm of the frequency on the horizontal axis. The labels on the vertical axis correspond to the time in the sound that was sampled. The horizontal labels give the frequency with spacings corresponding to log (frequency).

Figure 2 shows the constant Q transform for three notes ($G_3 = 196$ Hz, $G_4 = 392$ Hz, and $G_5 = 784$ Hz) generated in software. The fundamental frequencies increase by an octave and each sound contains 20 harmonics of equal amplitude. It is clear that the pattern of each is identical; only the positions on the frequency axis indicate that the notes are different. Figure 3 represents a 256-point traditional Fourier transform of this same sound for comparison. Here, the harmonics are equally spaced, and this is the major feature that stands out. The resolution for this case is 62 Hz so the harmonics of even the lowest note are resolved. However, the frequency of the next note in the musical scale differs from that of G_3 by only 12 Hz, so if the two notes were present simultaneously in a piece of music, it is clear that they could not be resolved. For the constant Q transform of Fig. 2, the percentage difference of nearby frequencies is the relevant parameter to consider for the question of resolution, and we note that these are indeed resolved up to the 20th harmonic where the frequencies differ by about 5%.

Figures 4 and 5 offer a comparison of the traditional (Fig. 4) and constant Q (Fig. 5) transforms for the analysis of the sound from a violin. Each is the transform of a diatonic scale ranging from G_3 to G_5. It is very difficult to say anything at all about spectral content for the conventional plot of Fig. 4; it is even difficult to determine note changes for the low-frequency notes. Figure 5, on the other hand, shows very clearly not only the note changes but also the spectral content; for example, G_3 and A_3 have an almost undetectable fundamental. Most striking of the spectral features is the formant in the region of 3000 Hz.

Figures 6–10 are included both for their musical interest and to demonstrate the power and versatility of this method of visualization for a variety of sounds. Figure 6 is the constant Q transform of a diatonic scale beginning on G_3 played pizzicato by the same violin used for Figs. 4 and 5 (and to be used in Figs. 7 and 8). The upper harmonics for the plucked string drop off in amplitude much more rapidly than for the bowed string, and there is little excitation in the region of the formant seen in Fig. 5. The low-frequency peak seen throughout is due to a ringing of the D string.

Figure 7 shows the constant Q transform for the violin playing the note D_5 with vibrato. The second harmonic is considerably weaker for the higher region of the vibrato while the 7th and 9th harmonics are weaker for the lower frequency region. Most remarkable in the spectrum is the extremely strong 6th harmonic. This note falls right in the formant in the region of 3000 Hz mentioned in connection with Fig. 5 so it is amplified by this body resonance. Another feature of the constant Q calculation that is brought out in this figure is the effect of the relationship between frequency of the spectral component and the center frequency of the bin in which it falls. This is clearest for the fundamental.

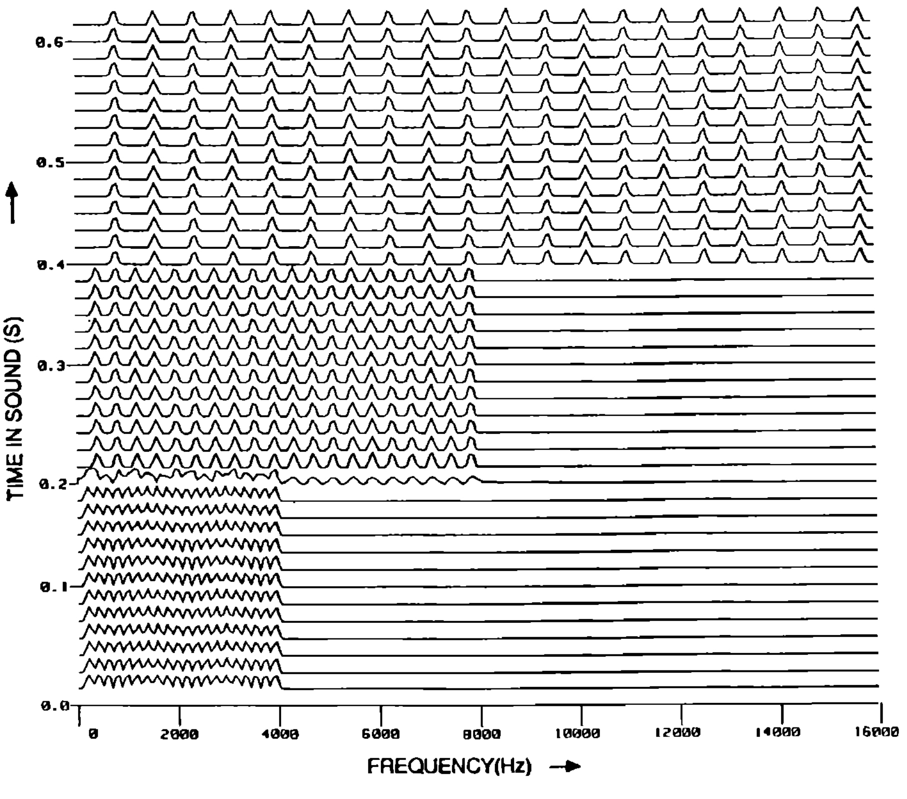

FIG. 3. Discrete Fourier transform of three complex sounds with fundamentals G_3 (196 Hz), G_4 (392 Hz), and G_5 (784 Hz), and each having 20 harmonics with equal amplitude.

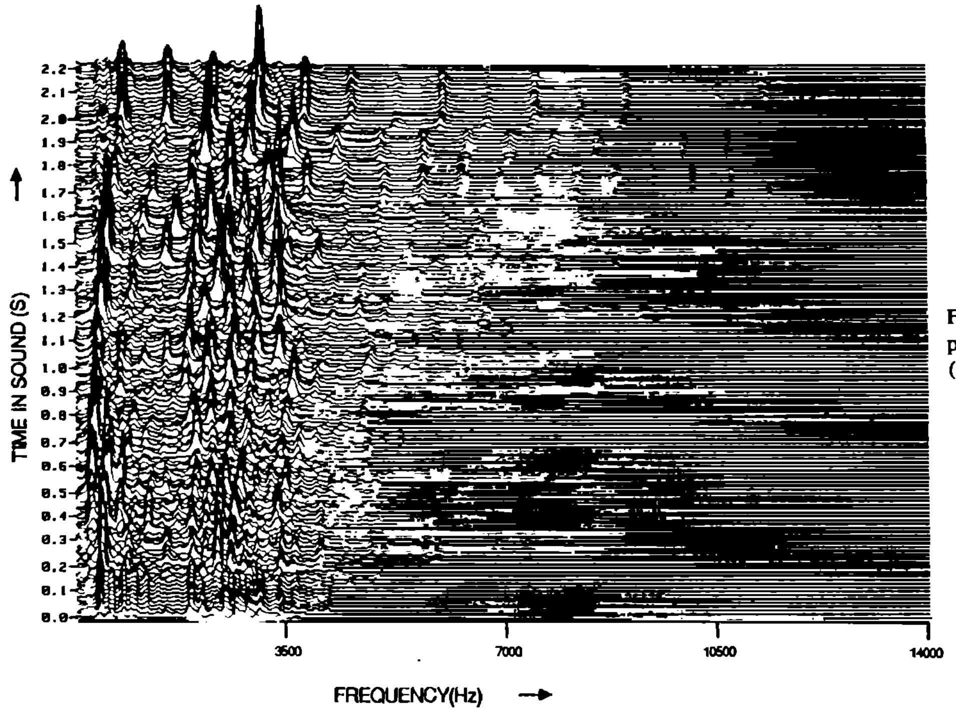

FIG. 4. Discrete Fourier transform of violin playing diatonic scale from G_3 (196 Hz) to G_5 (784 Hz).

Figure 8 shows a violin glissando from D_5 to A_5 and associated spectral changes. Figure 9 is a diatonic scale played by a flute beginning on C_4 where the amplitude is increasing dramatically. In the literature, it is often stated that the flute is nearly a pure tone, but this is far from the case here where approximately nine harmonics are visible.

Finally, Fig. 10 is the transform of a piano scale played from C_4 to C_5. The attack on D_5 is visible at the upper end. The graph is tilted so that the attacks and decays of the spectral components can be seen. The fundamental shows a rapid decay followed by a slow decay; this effect has been discussed by Weinrich.[20] The low end of the frequency range of the horizontal axis extends below that of the other graphs beginning with the frequency corresponding to B_2 rather than that

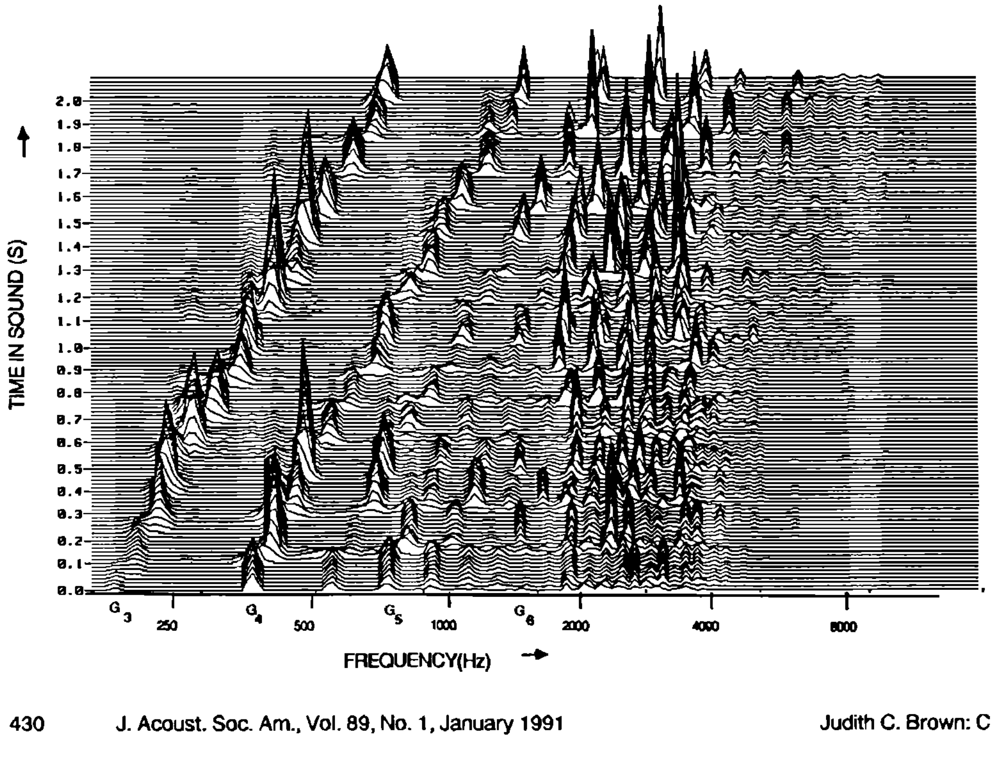

FIG. 5. Constant Q transform of violin playing diatonic scale from G_3 (196 Hz) to G_5 (784 Hz).

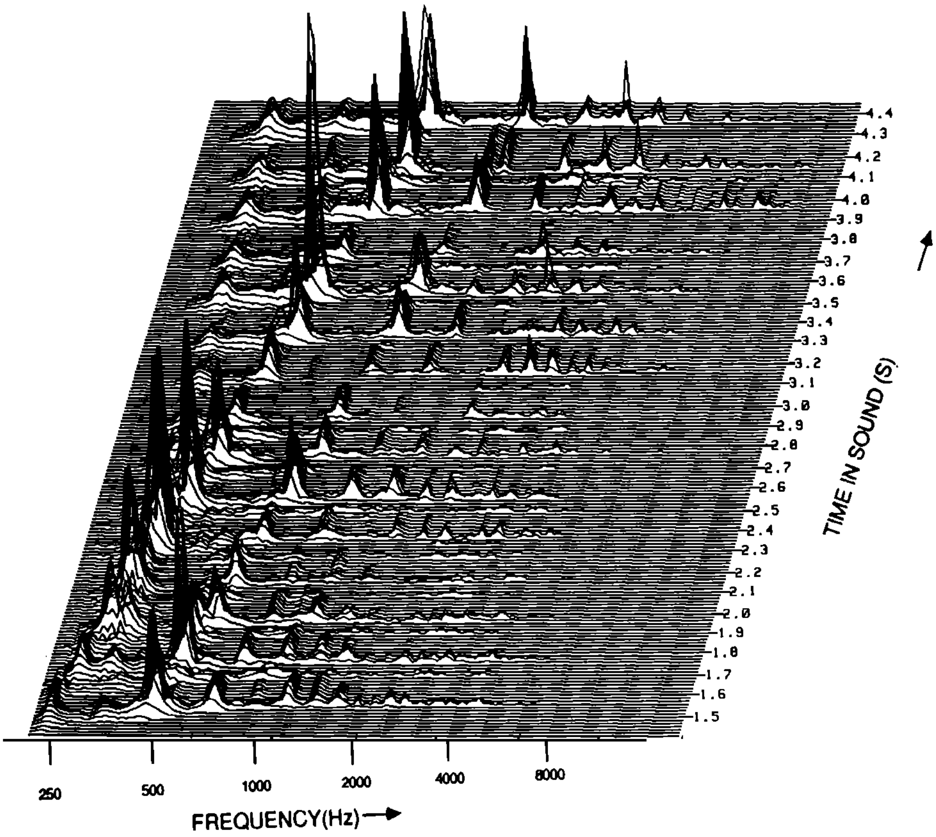

FIG. 6. Constant Q transform of violin playing diatonic scale pizzicato from G_3 (196 Hz) to G_5 (784 Hz).

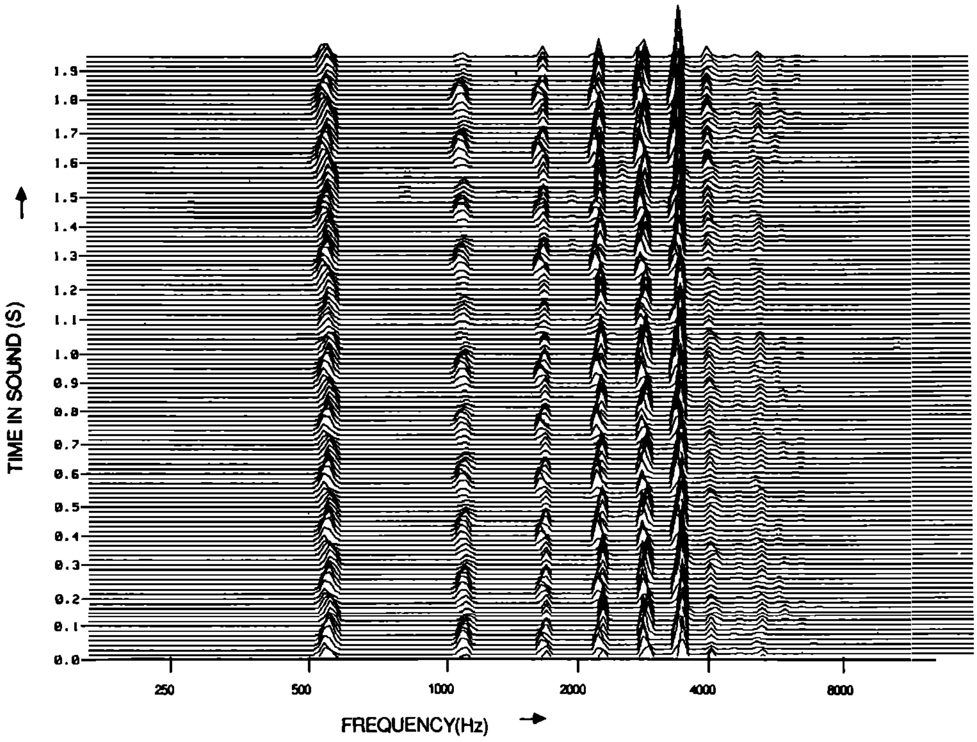

FIG. 7. Constant Q transform of violin playing D_5 (587 Hz) with vibrato.

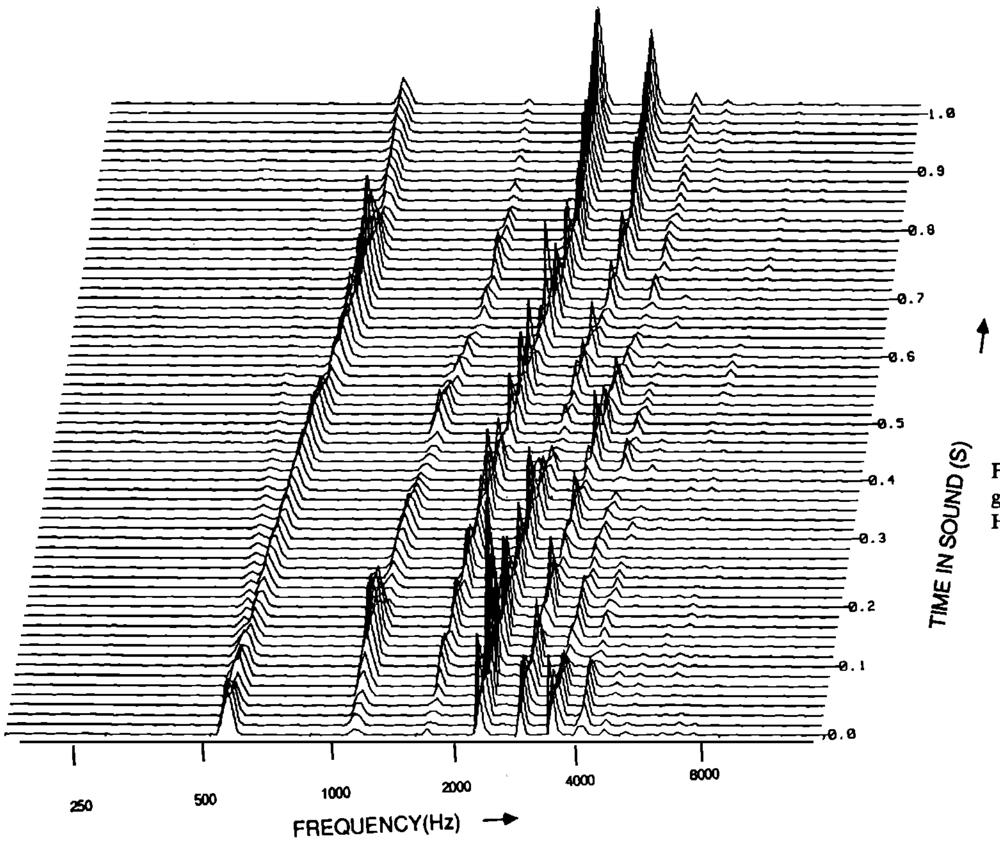

FIG. 8. Constant Q transform of violin glissando from D_5 (587 Hz) to A_5 (880 Hz).

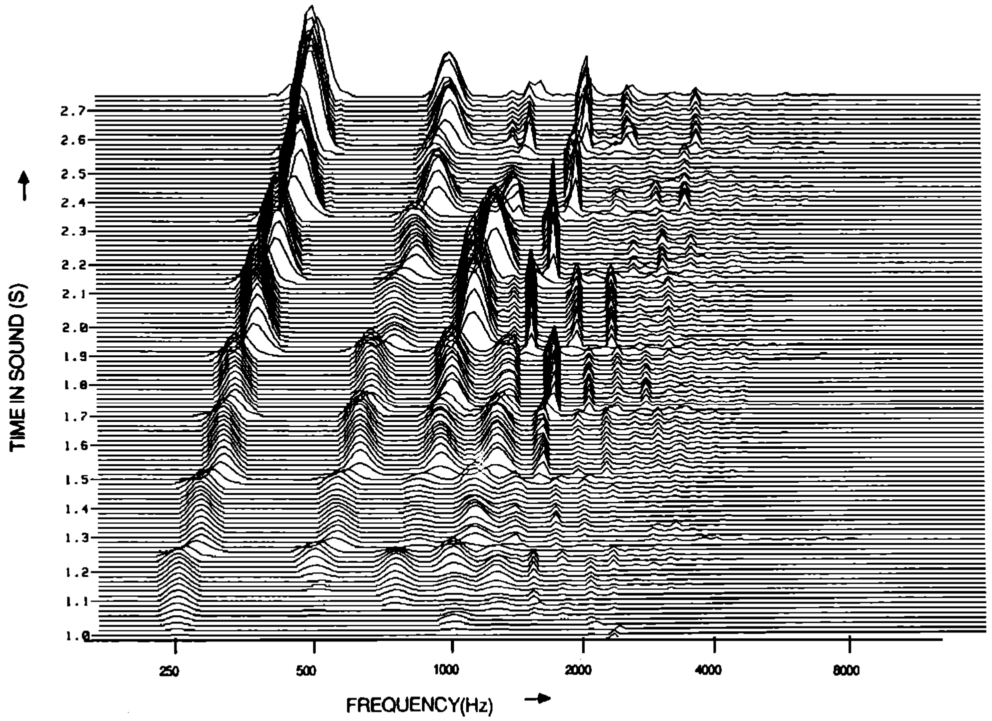

FIG. 9. Constant Q transform of flute playing diatonic scale from C_4 (262 Hz) to C_5 (523 Hz) with increasing amplitude.

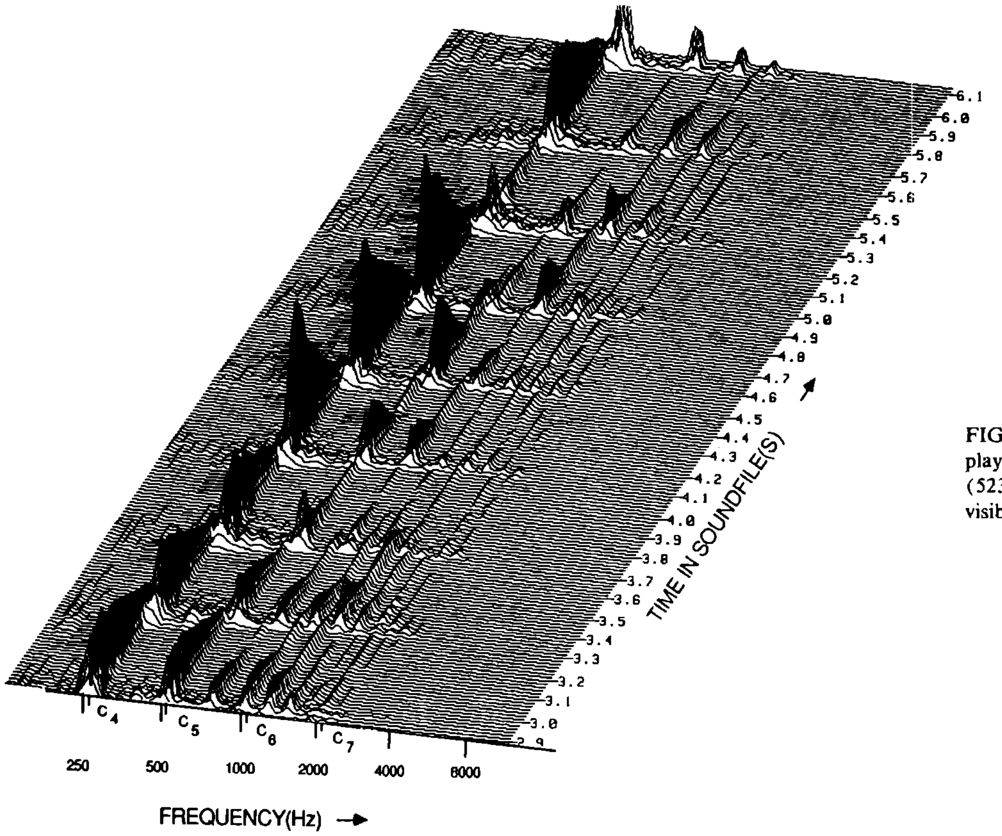

FIG. 10. Constant Q transform of piano playing diatonic scale from C_4 (262 Hz) to C_5 (523 Hz). The attack on D_5 (587 Hz) is also visible.

of F_3. This sound was recorded in a rectangular room (32×12×12 ft) with a highly directional microphone to minimize the effects of room modes.

IV. SUMMARY

We have used a straightforward method of calculating a constant Q transform designed for musical representations. This has been applied to sounds generated by a violin, flute, and piano representing the string, wind, and keyboard families of instruments. Waterfall plots of these data make it possible to visualize the large amount of information present in digitized musical waveforms. As predicted for sounds with harmonic frequency components, we obtain a constant pattern in the log frequency domain.

ACKNOWLEDGMENTS

I am very grateful to the Marilyn Brachman Hoffman Committee of Wellesley College for a fellowship for released time during which this work was accomplished and to Steven Haflich who originally suggested the idea behind this work. I would also like to thank David Rosenthal for many helpful comments on the manuscript.

[1] A. Gerson, and J. L. Goldstein, "Evidence for a general template in central optimal processing for pitch of complex tones," J. Acoust. Soc. Am. **63**, 498–510 (1978).

[2] S. Seneff, "Pitch and spectral analysis of speech based on an auditory synchrony model," Ph.D thesis, MIT (1985).

[3] J. P. Stautner, "Analysis and synthesis of music using the auditory transform," MIT M. S. thesis in Electrical Engineering and Computer Science, May (1983).

[4] R. J. Higgins, "Fast fourier transform: An introduction with some minicomputer experiments," Am. J. Phys. **44**, 766–773 (1976).

[5] M. R. Schroeder and B. S. Atal, "Generalized short-time power spectra and autocorrelation functions," J. Acoust. Soc. Am. **34**, 1679–1683 (1962).

[6] J. S. Lim and A. V. Oppenheim, *Advanced Topics in Signal Processing* (Prentice-Hall, Englewood Cliffs, NJ, 1988).

[7] J. E. Youngenberg and S. F. Boll, "Constant-Q signal analysis and synthesis," Rec. IEEE Int. Conf. Acoust., Speech, Signal Process. **CH-1285**, 375–378 (1978).

[8] F.J. Harris, "High-resolution spectral analysis with arbitrary spectral centers and arbitrary spectral resolutions," Comput. Electr. Eng. 3, 171–191 (1976).

[9] H. D. Helms, "Power spectra obtained from exponentially increasing spacings of sampling positions and frequencies," IEEE Trans. Acoust. Speech, Signal Process., **ASSP-24**, 63–71 (1976).

[10] A. Oppenheim, D. Johnson, and K. Steiglitz, "Computation of spectra with unequal resolution using the fast Fourier transform," Proc. IEEE **59**, 299–301 (1971).

[11] C. Braccini and A. V. Oppenheim, "Unequal bandwidth spectral analysis using digital frequency warping," IEEE Trans, ASSP **22**, 236–244 (1974).

[12] G. Gambardella, "A contribution to the theory of short-time spectral analysis with nonuniform bandwidth filters," IEEE Trans. Circuit Theory **CT-18**, 455–460 (1971).

[13] G. Gambardella, "The Mellin transforms and constant-Q spectral analysis," J. Acoust. Soc. Am. **66**, 913–915 (1979).

[14] R. Bracewell, *The Fourier Transform and Its Applications* (McGraw-Hill, New York, 1965).

[15] D. T. Teaney, V. L. Moruzzi, and F. C. Mintzer, "The tempered Fourier transform," J. Acoust. Soc. Am. **67**, 2063–2067 (1980).

[16] K. L. Kashima and B. Mont-Reynaud, "The bounded-Q approach to time-varying spectral analysis," Stanford Dept. of Music Tech. Rep.

STAN-M-28 (1985).

[17] R. Kronland-Martinet, "The wavelet transform for analysis, syntheses, and processing of speech and music sounds," Comput. Music J. **12** (4), 11–10 (1988).

[18] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing* (Prentice-Hall, Englewood Cliffs, NJ, 1989).

[19] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," Proc. IEEE **66**, 51–83 (1978).

[20] G. Weinrich, "Coupled piano strings," J. Acoust. Soc. Am. **62**, 1474–1484 (1977).