Audio Engineering Society

Convention Paper

Presented at the 130th Convention
2011 May 13–16 London, UK

The Quintessence of a Waveform: Focus and Context for Audio Track Displays

Jörn Loviscach[1]

[1] *Fachhochschule Bielefeld (University of Applied Sciences), 33615 Bielefeld, Germany*

Correspondence should be addressed to Jörn Loviscach (joern.loviscach@fh-bielefeld.de)

ABSTRACT

Oscilloscope-style waveform plots offer great insight into the properties of the audio signal. However, their use is impeded by the huge spread of timescales extending from fractions of a millisecond to several hours. Hence, waveform plots often require zooming in and out. This work introduces a graphical representation through a synthesized quintessential waveform that shows the spectrum of the traditional waveform plot but does so at a much larger timescale. The quintessential waveform can reveals details about single periods at zoom levels where a regular waveform plot only indicates the signal's envelope. Compression renders the enormous ranges of frequencies and amplitudes more legible.

1. INTRODUCTION

A waveform as shown on an oscilloscope helps to answer lots of questions about an audio signal: What is its amplitude? How does it evolve over time? Is the signal clipped? Does it contain unwanted noise? What are its main frequency components? Is it tonal, metallic, or noise-like? Is it polyphonic? Answers to these questions help solve higher-level questions such as: Can the audio track be divided into meaningful regions? Which musical instruments or voices or noises are contained in the audio signal?

One could answer these questions by actually listening to the audio signal. This would, however, take much more time and may not reveal all details. Although the waveform plot helps overcome this, it is limited by the need to zoom to different levels of detail. This is a fundamental property of audio signals, as they comprise timescales ranging from a fraction of a millisecond (a single period of an oscillation) up to a few seconds (a measure of a piece of music) or even several hours (a movie or an opera).

Spectrograms and other alternatives to the waveform display do not require zooming to help address

some of the questions such as whether a signal is tonal or noisy, but these approaches can rarely replace a waveform display completely. Enhancements to standard waveform displays, such as a coloring according to the strongest frequency component, reduce some need for zooming but only show a very coarse view of the details of the signal.

To improve upon these solutions, this work proposes to synthesize a "quintessence waveform" that mimics the actual content of the given signal on a timescale which is slowed down by several orders of magnitude, see Figure 7. The resulting waveform resembles greatly magnified snippets of the original waveform that have been strung together. The extraction, however, is continuous, without borders between snippets, so that it also reveals changes on short time scales, useful for identifying short elements such as noise bursts. This waveform can be used alone or can be graphically superimposed on the regular waveform display, which helps to accentuate the amplitude envelope.

This display leverages the audio engineer's knowledge of waveforms. The basic approach can be likened to the focus-plus-context approach in information visualization, in which one attempts to present details and an overview at the same time. The quintessence waveform, however, shows generalized detail and does not require panning a lens across an overview image.

The quintessence waveform is generated through an extreme pitch-shifting process. To better use the limited space of the display, amplitudes and frequencies are subjected to nonlinear mappings.

This paper is structured as follows: Section 2 discusses related approaches. Section 3 describes the basic approach, the fine-tuning of which through nonlinear mappings of amplitudes and frequencies is subject of Section 4. Section 5 presents visual comparisons of different settings of the proposed method and of other approaches to this visualization problem. Section 6 concludes this paper and gives an outlook on future work.

2. RELATED WORK

Whereas the first digital audio workstations supported only oscilloscope-style waveform displays, many ways of enhancing or replacing this basic visualization have been proposed and implemented since.

A basic variation is to represent the sample values on a logarithmic scale [2]. This helps to indicate the perceived loudness and renders pianissimo passages more legible. A standard choice to indicate the evolution of the frequency content over time is a spectrogram [2, 4]. More advanced approaches [3, 14] aim to reproduce the look of a score, which requires finding and joining all partials of a single note.

Acoustic parameters extracted from the audio content can be mapped to color and turned into displays consisting of vertical stripes along a horizontal time axis such as in Timbegrams [6] and related methods [10]. These can also be used to color the inner part of the regular waveform display [5, 13].

Former work by the author and his coworkers [9] lists use cases for waveform displays and addressed specific ones of these such as showing reverberation tails and the perceived acoustic contribution of a track to the final mix. That paper proposes a seamless loupe as a focus-and-context display for waveforms.

On a higher level, one can show the structure of an audio file. For instance, a self-similarity matrix can be turned into a square image in which the repeating parts of a piece of music can be spotted easily [8]. Another option is to draw arcs that connect related parts of the time axis [15].

3. BASIC APPROACH

The vital step in the generation of the quintessence waveforms consists of a drastic pitch shift downward. This is handled through a phase vocoder with FFT analysis and oscillator bank (sum of sinusoids) synthesis [1]. When generating audible signals, it is more efficient to use an FFT also for synthesis. In the application at hand, however, the number of output samples can be heavily reduced, as the frequencies are very low. Thus, an oscillator bank is more efficient here. The examples in this paper use a downsampling factor of 100, so that the quintessence waveforms are generated at 441 Hz for CD-quality input.

The FFT window has to be long enough to accommodate the deepest frequencies to be analyzed. Otherwise, such oscillations would be treated as DC

components in the FFT and hence would be reproduced verbatim, that is: at their original frequency, without pitch-shifting. Hence, the FFT window is set to 4096 samples. Given the sampling rate of 44,100 Hz, the FFT bins' frequencies lie approximately 11 Hz apart so that also the lower end of the audible range of frequencies is safely covered. A hop size of 32 samples turned out to be low enough to create clean waveforms even for fast sweep signals.

Phase vocoders are known to lead to "phasiness" in the output signal. This artifact is caused by cancellations between the synthesis results of adjacent FFT bins. With the quintessence waveforms, it leads to spuriously low amplitudes, which is not tolerable. Hence, the phase angles of the output signal generated from adjacent FFTs bins must be adjusted to mimic those in the input signal. Miller Puckette's "Phase-Locked Vocoder" [11] and more sophisticated approaches to phase-locking [7, 12] address this problem.

The visualization is not as critical in terms of precision as is an acoustic rendition. It turned out that already a straightforward solution for phase-locking works for the visualization, see Figure 1. If the FFT bin number k has a lower power than its neighbor with number $n = k - 1$ or $k + 1$, the phase angle of the sine oscillator corresponding to k is set to the phase angle of the sine oscillator corresponding to n plus the difference of the phase angles between the bins k and n in the FFT analysis. This way, the sine waves generated from both bins have the same phase difference as the FFT analysis bins have.

To accelerate the rendering on the screen, most audio production software precomputes envelopes of the waveforms for coarse zoom levels and stores these as files. In a similar vein, the quintessence waveform can be precomputed and then stored. Thanks to the 100:1 downsampling enabled by its low frequencies, storing this waveform requires less than one kilobyte of memory per second in the prototype.

A full use of the method, however, requires synthesizing the quintessence waveform according to the current zoom level. In this case, only the analysis part can be handled as a precomputation. The amplitudes and the pitched-down phases resulting from that precomputation have to be stored. In the prototype, this amounts to $2 \times 2048 \times 441$ floating-point values per second, necessitating strong compression.

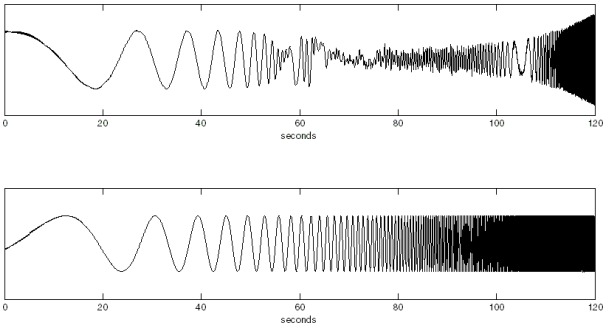

Fig. 1: Phase-locking to the stronger neighbor (bottom) already suffices to get rid of the most eminent "phasiness" artifacts (top). The input signal is a sine sweep from 20 Hz to 20 kHz, pitch-shifted linearly so that a frequency of 440 Hz becomes a period length of 20 pixels.

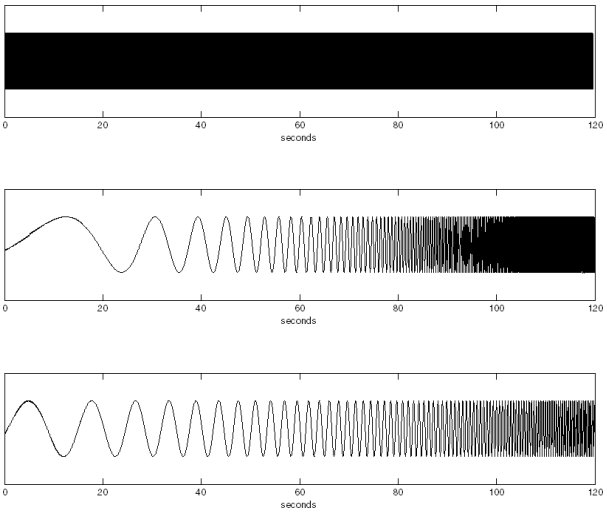

Fig. 2: A regular waveform plot of a logarithmic sine sweep from 20 Hz to 20 kHz results in a uniform block, which hardly reveals anything about the content (top). A linear pitch-shift multiplies all frequencies by a constant, in this case mapping 440 Hz to a period 20 pixels (middle). A nonlinear mapping defined by an power function can be used to map the audible range of frequencies to visual period lengths that are easily visually discernible (bottom, see text for data).

The quintessence waveform is an easily discernible line, unlike the regular waveform, which appears in all but the most detailed zoom levels as a densely filled area. Hence, it is possible to place a light image of the original waveform behind the quintessence waveform, thus providing even more information to the viewer. This combination of the two waveforms can be seen in Figures 5, 6, and 7. The background waveform is rendered as a white shape on a gray background and not the other way around, which would be typical. This way, the black quintessence wave clearly stands out from its white envelope formed by the regular waveform.

4. MAPPING PITCH AND AMPLITUDE

Once the pitch shifter is in place, one can vary the mapping of frequencies and amplitudes between input and output. By no means do they simply have to be multiplied by constant factors. This enables further enhancements of the display's legibility.

The audible range of frequencies encompasses almost three orders of magnitude. If an oscillation of 20,000 Hz was rendered as narrow as possible while still being discernible on the screen, one period of it would have to be 2 pixels wide. When frequencies would be mapped linearly, an oscillation of 20 Hz would then have a period length of 2000 pixels, which is about the screen width of current computer displays. Hence, it looks sensible to reduce this range.

A massive reduction of the range of period lengths can be obtained as follows: A frequency of 20,000 Hz is mapped to a period length of 2 pixels; a frequency of 440 Hz is mapped to a period length of 20 pixels; all other frequencies are mapped by fitting an exponential function through these two given points. Hence, the frequency ratio of one octave is reduced from $2:1$ to a ratio of $r \approx 1.5:1$, see the bottom part of Figure 2. To accomplish this, the pitch shifter has to turn each incoming frequency f to $f_1\, r^{\log_2(f/f_0)}$. Here, f_0 and f_1 are appropriate constants; an input of f_0 results in an output of f_1. In total, the pitch shifter has to multiply each input frequency f by $f_1\, r^{\log_2(f/f_0)}/f$. Noting $r = 2^{\log_2(r)}$ one finds that the frequency factor is

$$\frac{f_1}{f}\, 2^{\log_2(r)\,\log_2(f/f_0)} = \frac{f_1}{f} \left(\frac{f}{f_0}\right)^{\log_2(r)}.$$

The period length of a quintessence wave does not depend on the zoom level, see Figure 3. Thus, the user can learn which period length on the screen corresponds to which acoustic frequency. Such a memorization is not possible with regular waveforms, as their period lengths depend on the zoom level and hence vary greatly.

Similarly, the enormous audible range of amplitudes can be compressed for the display. First consider a sample value of a regular waveform. One option would be to replace the sample value $x \in [-1, 1]$ by $(1 + \frac{1}{3}\log_{10}(|x|))\mathrm{sgn}(x)$ if $|x| > 10^{-3}$ and else by 0. This would map a range of 60 dB to the number interval $[-1, 1]$. Such a purely logarithmic mapping requires a lower cutoff, however, such as the 60 dB in this example, because the logarithm can become an arbitrarily low negative number. In addition, the logarithmic mapping places too much importance onto short quieter passages such as background noise between sentences of recorded speech.

Considering this, it makes sense to choose a power function instead of a logarithm. A power function also corresponds to the computation of loudness in sone units: Increasing the sound pressure measured in Pascal by a factor of $\sqrt{10}$ doubles the number of sone. The relationship is merely the power function $y = x^{2\log_{10}(2)} \approx x^{0.6}$. Hence, the dynamic range of the quintessence waveform is compressed by replacing the amplitude a of each of the sine oscillators by $(ca)^{0.6}/c$ with the factor c adjusted to account for the contribution of the adjacent, phase-locked bins. The dynamics of the regular waveform in the background is transformed correspondingly by computing its power envelope $p(t)$ and then multiplying the signal by $\frac{\sqrt{p(t)}^{0.6}}{\sqrt{p(t)}} = p(t)^{-0.2}$.

5. VISUAL RESULTS

Figures 5, 6, and 7 show the graphical results for six wave files in three different settings: linear pitch-shifting, compressed mapping of pitch and strongly compressed mapping of pitch. For details, see the captions. All three employ an amplitude compression with an exponent of 0.6.

The audio file "celestaglocken" starts with a celesta and then continues with a glockenspiel. The two

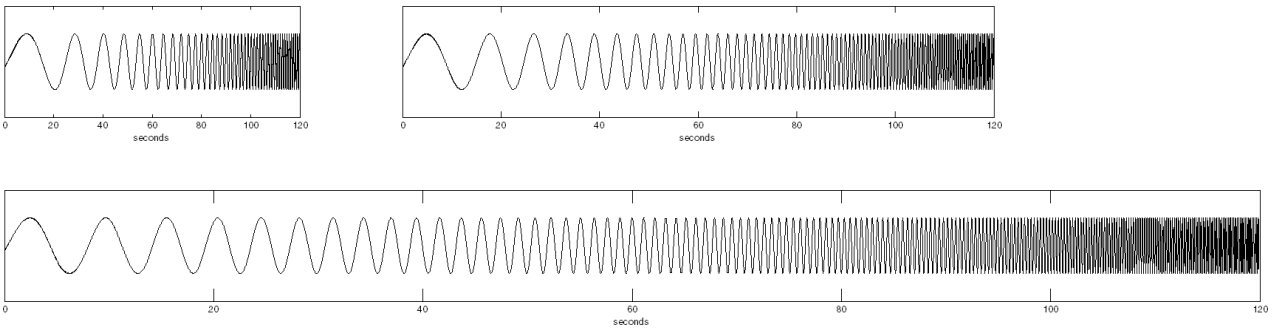

Fig. 3: Unlike a regular waveform, the quintessence waveform can map a given period length to a fixed period length in terms of screen pixels independent of the zoom level. All three screenshots show the same sine sweep as Figure 2 and employ the same mapping as does the bottom part of that figure.

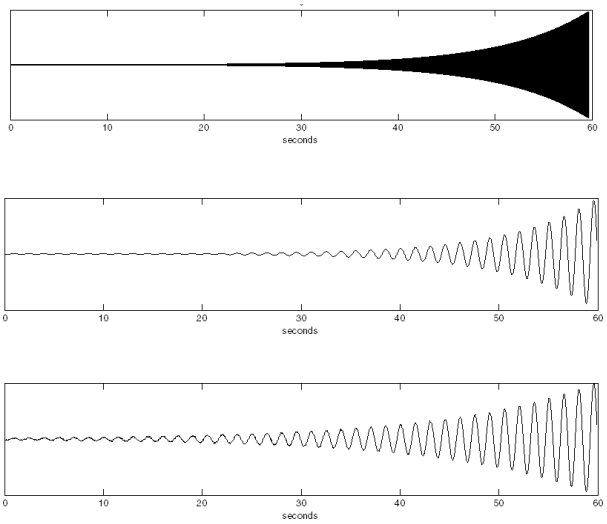

Fig. 4: A sine wave of 440 Hz is faded in from −60 dB to 0 dB over a time of 60 s, that means at a rate of 1 dB/s. The regular waveform display (top) hardly shows any content below −30 dB. In the pitch-shifted quintessence waveform (middle) at least a basic oscillation pattern becomes visible. Modifying the amplitudes by a power law enhances low-level signals in an unobtrusive way (bottom, see text for data).

nonlinear mappings clearly show the difference between the harmonic content of these two similar-pitched instruments. The bass line "fingeredbass" contains low frequencies that are brought out legibly by the linear mapping and by the strongly compressed mapping. The drum loop "hiphopdrums" comprises a bass drum sound (0 s) with a click at its end (0.25 s), a shaker sound (0.4 s) and a snare drum sound (0.5 s). These are discernible in the strongly compressed mapping. The French horn ensemble "hornsmellowbright" starts with mellow sounds and then changes to a bright timbre. This change is visible in both compressed mappings.

The file "oooeee" contains major scales sung by a male voice, first on the syllable 'o', then on the syllable 'e'. The difference in timbre is displayed well in all three settings. The low frequencies, however, are again handled best by the linear mapping and by the strongly compressed mapping. "speech" is a male voice saying "Dear Ladies and Gentlemen, I declare this exhibition to be opened." The quintessence waveforms clearly mark the two "s" and the "x" in these sentences. The third setting also brings out the "d" and the "th".

For a comparison with existing solutions, Figure 8 shows a variety of displays of the "speech" wave file in current audio software. On first inspection, the quintessence waveforms with the setting of Figure 7 is clearly more informative than all four options provided by Audacity and is on par with the patent-protected Comparisonics scheme.

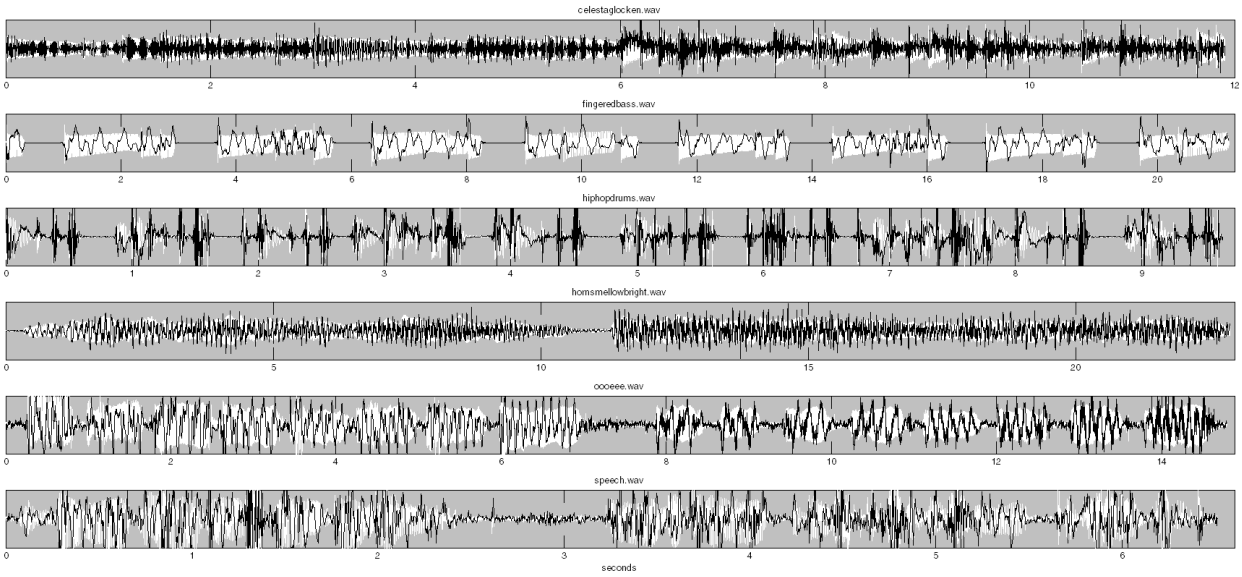

Fig. 5: Linear pitch shift with a period of 20 pixels corresponding to a frequency of 440 Hz. Amplitudes compressed by a power function with an exponent of 0.6.

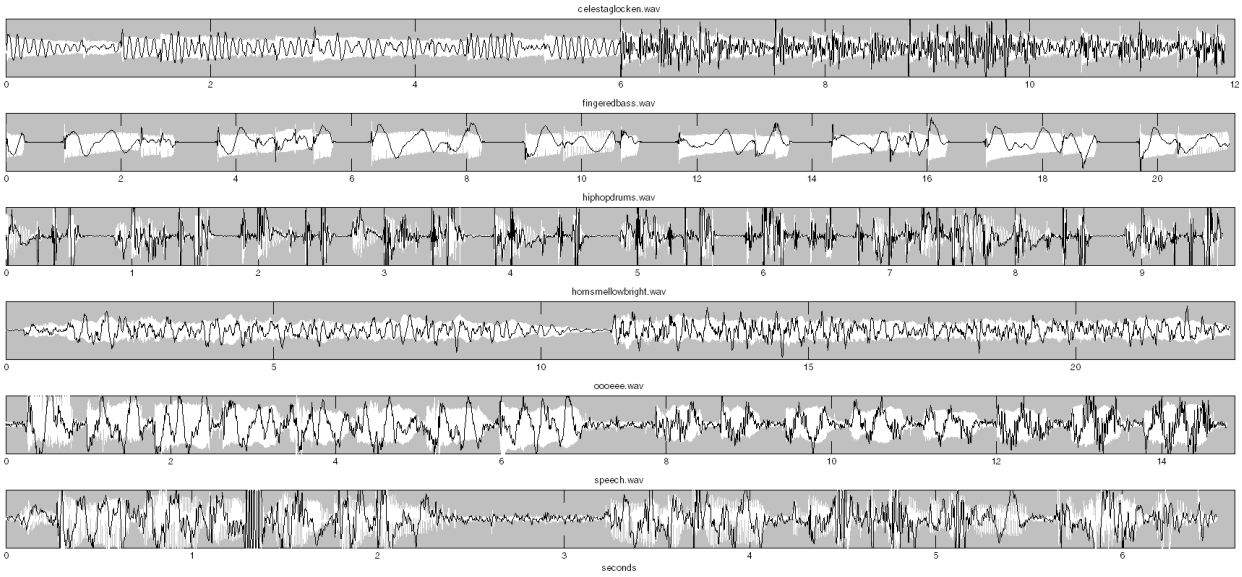

Fig. 6: Compressed mapping of pitch with a period of 20 pixels corresponding to a frequency of 440 Hz and 2 pixels corresponding to 20 kHz. Amplitudes compressed by a power function with an exponent of 0.6.

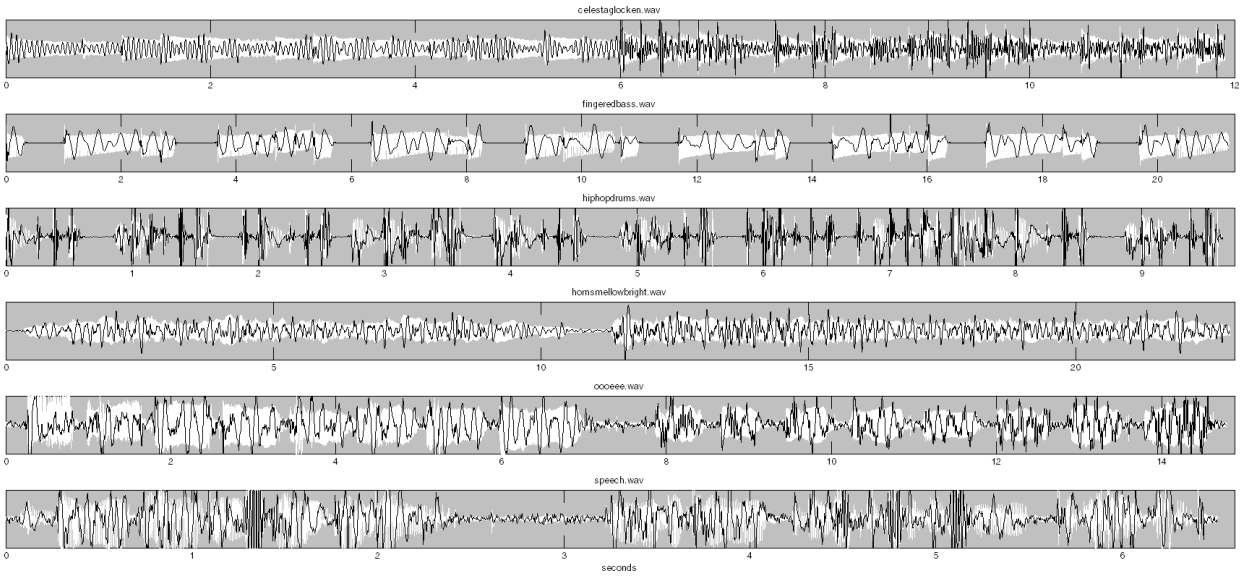

Fig. 7: Strongly compressed mapping of pitch with a period of 20 pixels corresponding to a frequency of 110 Hz and 2 pixels corresponding to 20 kHz. Amplitudes compressed by a power function with an exponent of 0.6.

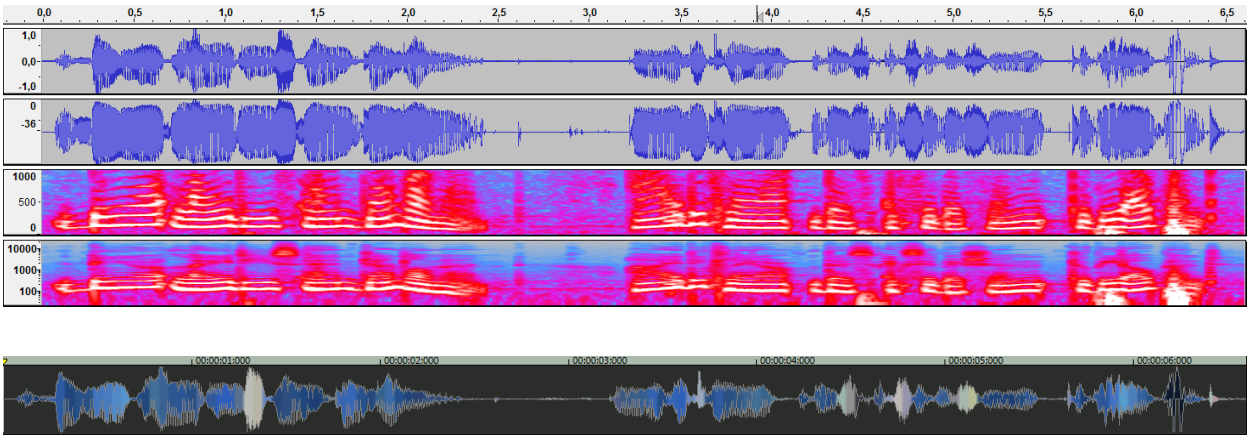

Fig. 8: The "speech" wave in four display settings of Audacity 1.3.12-beta [2]: waveform, waveform (dB), spectrum, spectrum log(f). The bottom show the Comparisonics coloring scheme [5] of Magix Samplitude 11.02.

6. CONCLUSION AND OUTLOOK

This paper has described a visualization method based on waveforms on compressed time and amplitude scales. Examples show that the content of single tracks is mapped faithfully and legibly.

However, there remain issues with the visualization. First, with complex polyphonic material the quintessence waveforms are as hard to interpret as are regular waveforms. Second, synthetic waveforms with strong partials such as sawtooth and rectangle—which occur rarely in recording practice—are not mapped to anything resembling their original shape, as not only the relative phases of the partials are lost but also their frequency ratio is compressed. Third, the method currently only handles monoaural waves. A multi-channel application would require phase synchronization between the different channels.

One option to address many of these issues could be to combine all partials belonging to each note in the mix, similar to advanced audio editing applications such as [3]. The loudness, pitch, and timbre of each note would then be used to create a stand-in waveform for that note in the visualization.

7. REFERENCES

[1] D. Arfib, F. Keiber, and U. Zölzer. Time-frequency processing. In *DAFX – Digital Audio Effects*, pages 237–297. John Wiley & Sons, Chichester, UK, 2003.

[2] Audacity Developer Team. Audacity. `http://audacity.sourceforge.net/`, accessed 2011-03-06.

[3] Celemony Software GmbH. Melodyne. `http://www.celemony.com/`, accessed 2011-03-06.

[4] Centre for Digital Music, Queen Mary, University of London. Sonic Visualizer. `http://www.sonicvisualiser.org/`, accessed 2011-03-06.

[5] Comparisonics Corp. Comparisonics waveform display. `http://www.comparisonics.com/`, accessed 2011-03-06.

[6] P. R. Cook and G. Tzanetakis. Audio information retrieval (AIR) tools. In *Proceedings of ISMIR*, 2000.

[7] A. Ferreira. An odd-DFT based approach to time-scale expansion of audio signals. *IEEE Transactions on Speech and Audio Processing*, 7(4):441–453, 1999.

[8] J. Foote. Visualizing music and audio using self-similarity. In *Proceedings of ACM Multimedia*, pages 77–80, 1999.

[9] K. Gohlke, M. Hlatky, S. Heise, D. Black, and J. Loviscach. Track displays in DAW software: Beyond waveform views. In *Proceedings of the 128th Convention of the AES*, 2010. Paper No. 8145.

[10] A. Mason, M. Evans, and A. Sheikh. Music information retrieval in broadcasting: Some visual applications. In *Proceedings of the 123rd Convention of the AES*, 2007. Paper No. 7238.

[11] M. Puckette. Phase-locked vocoder. In *Proceedings of IEEE ASSP Workshop on the Applications of Signal Processing to Audio and Acoustics*, pages 222–225, 1995.

[12] T. Quatieri, R. Dunn, and T. Hanna. A subband approach to time-scale expansion of complex acoustic signals. *IEEE Transactions on Speech and Audio Processing*, 3(6):515–519, 1995.

[13] S. V. Rice. Frequency-based coloring of the waveform display to facilitate audio editing and retrieval. In *Proceedings of the 119th Convention of the AES*, 2005. Paper No. 6530.

[14] S. Saito, H. Kameoka, K. Takahashi, T. Nishimoto, and S. Sagayama. Specmurt analysis of polyphonic music signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(3):639–650, 2008.

[15] H.-H. Wu and J. P. Bello. Audio-based music visualization for music structure analysis. In *Proceedings of SMC*, 2010. Paper No. 73.