Rethinking Audio Visualizations: Towards Better Visual Search in Audio Editing Interfaces

Evelyn Eika[1] and Frode E. Sandnes[1,2]

[1]Oslo and Akershus University College of Applied Sciences, Norway
[2]Westerdals Oslo School of Art, Communication and Technology
Evelyn.Eika@hioa.no, frodes@hioa.no

Abstract. Waveform visualization is a key tool in audio editing. However, the visualization of audio waveforms has changed little since the emergence of the first software systems for audio editing several decades ago. This paper explores how audio is visualized. This paper shows that the commonly used time-domain representation exhibits redundant information that occupies valuable display real-estate in most audio editing software. An alternative waveform visualization approach is proposed that exploits elements from the existing visualization conventions while enhancing features that are important in visual search through digital audio. Alternatively, the method is a means for making more efficient use of the display real-estate. The proposed method is discussed in terms of its suitability for various visualization situations.

Keywords: audio data, waveforms, visualization, audio editing, music software, spectrogram

1 Introduction

Audio editing is used in many domains such as academic acoustical phonetic research [1], music production and recording [2], live performances [3], DJ'ing [4, 5], radio production, and more recently in podcasting [6]. Various forms of audio editors exist and are easily available such as the open source Audacity audio editor [7]. Common to most audio editors is that audio is represented directly as the audio waveform, that is, the audio signal in the audio domain. The audio waveform has become an iconic representation with the appearance of a diverse mountain range mirrored across a lake (see Fig. 1). Users will immediately recognize an audio visualization as audio due to the visual signature of the visualization without needing an explanation.

There have been very few developments in audio visualization since the first emergence of the digital audio-visual editing software more than three decades ago. One noteworthy exception is the use of color-to-code frequency characteristics in the time-domain signal. In other words, audio segments with different frequency signatures or transients are represented using different colors or intensities also known as EQ color coding. Such coding is used in several pieces of state-of-the-art DJ'ing software such as Serato [8] and Traktor [9] to help the DJs more easily locate cue points.

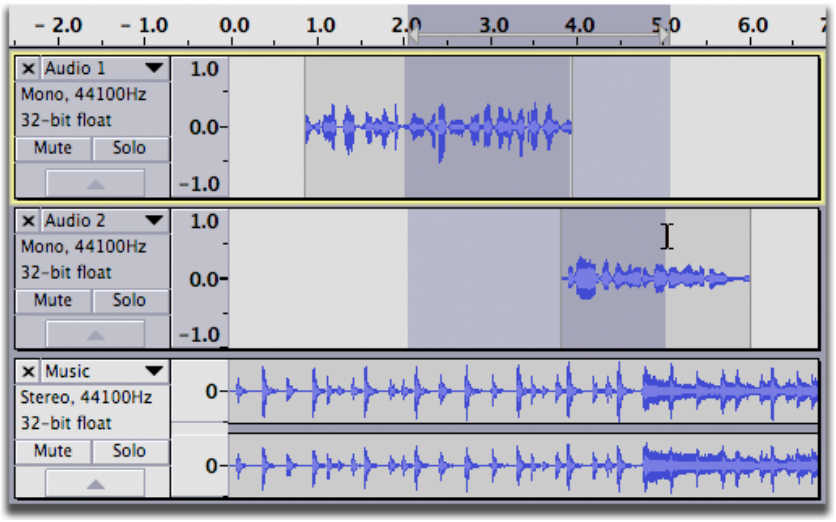

Fig. 1. Screenshot from the Audacity audio editor (reproduced from the User Manual, GNU General Public License). Multiple audio tracks are easily recognizable by their "mirrored mountain ranges". Potential cue points are recognizable as peaks in the signal.

One advantage of the color coding representation is that it is secondary to the time domain representation [10] that can help the user more quickly visually locate what they are looking for [11].

The visualizations of real-world audio appear as sequences of peaks and valleys mirrored above and below the timeline. We hypothesize herein that it is this mirroring effects that probably is the key feature that triggers the recognition of the visualization as an audio representation. Although this recognizable feature is a benefit, we argue that the mirroring of the signal above and below the time-line in most situations is redundant. Moreover, there is a limit to how many tracks that can be displayed simultaneously on a screen. We thus propose an audio visualization technique based on only displaying one side of the waveform. Special cases, in particular, DC information, are handled explicitly.

2 Background

The goal of the visualization field is to find the most effective ways to display data, being it numbers [12], texts [13], or volumes [14], and to facilitate user tasks. In the audio domain, VU-meters (volume unit meters) were used to visualize the signal level in early analogue recording equipment, and these analogue instrument meters were later replaced by LED-based VU-meters that are simple time-varying bar graphs.

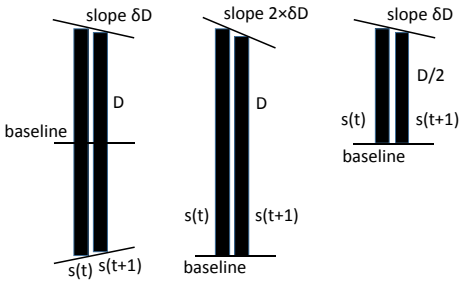

Fig. 2. Traditional waveform visualization mirrored above and below the axis (left), the proposed waveform visualization without mirroring (middle) and the proposed waveform visualization with half the width.

The emergence of audio editing software allowed the recorded signals to be displayed as a function of time, namely, the waveform. This representation has survived with minor changes for several decades. In addition to the already mentioned improvements, such as color-coding of frequency information [8, 9], beat mark annotations has also been explored [5].

Root mean square (RMS) amplitude plots are commonly used in speech analysis software such as Praat [16], which can show the intensity of the signal as a trend as the absolute signal intensity is averaged over a moving window. The method proposed herein has similarities to RMS amplitude plots, but the proposed approach does not mask rapid changes in the audio signal. Moreover, the proposed method is capable of showing DC offsets which information is lost with the RMS calculation.

Oscilloscopes are often used to visualize signals in the time-domain, and oscilloscopes have allowed waveforms to be inspected before graphical user interfaces were commonly available [17]. Still, they have been less used by individuals working with audio, probably due to the fact that audio equipment was not built with oscilloscopes.

Spectrum analyzers were also used with early specialized audio equipment, and used to show the intensity of the various frequency bands as snapshots in time. Visual audio software allowed these spectrograms to be plotted as a two-dimensional image where the frequency bands are plotted as a function of time. Saliency maximization of two-dimensional audio spectrograms [18] was proposed to improve the visualizations. Frequency time-series have also been visualized in three dimensions [19]. More sophisticated domain-specific visualization techniques exist such as tone plots [20], fundamental frequency plots, and formant plots, as used in phonetic research [1].

Mel-scale spectrums, chromagrams, periodograms [21], phase space plots [22], and self-similarity plots, used for discovering repeating patterns [23], are examples of other types of audio visualizations. DiskPlay [24], a system where an overhead projector is used to project an image onto a white time-coded vinyl record, can be classified as utilizing a context-specific audio visualization. Various colors are used to indicate parts of a track that is already played, areas to be played, unused time-codes, cue points, etc.

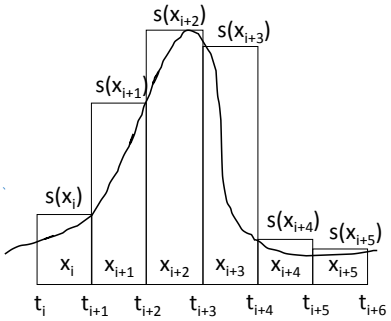

Fig. 3. Converting a signal from time-domain to pixel-domain by selecting the maximum value within each time window as a representative value for the audio signal.

In a slight variation on the same theme, the audio waveform is projected directly only to the white vinyl disk [25]. The waveform projection allows the user to access the audio directly via the turntable. The method proposed herein could be used to enhance such visualizations.

With increasing amounts of unannotated audio and video material, visualization is one valuable tool, in addition to query by humming [26], to help a user navigate large audio contents without the time-consuming task of going through hours, or even days, of audio material. There are several initiatives to make visualization tools for this problem domain [27]. Researchers have also explored domain-specific visual tools for manually transcription of music [28] and speech [29]. Another avenue of research includes large display audio visualization [30].

3 The Proposed Method

The audio visualization proposed herein assumes that the audio is viewed as a waveform, that is, the amplitude of the audio signal is plotted as a function of time. Moreover, it is assumed that the audio signal is not viewed at sample level or micro level but at macro level (in seconds). The resulting view is thus an aggregated form of the waveform that appears as the energy, or loudness, of the signal. It is this assumption of aggregated view that forms the basis for the proposed method since the detailed information at waveform level is lost when viewed at a more course time scale. At the more coarse-grained time scale, the energy signature appears mirrored above and below the time axis.

We thus argue for only showing one side of the waveform, effectively the absolute value, namely

$$w'(t) = |w(t)| \tag{1}$$

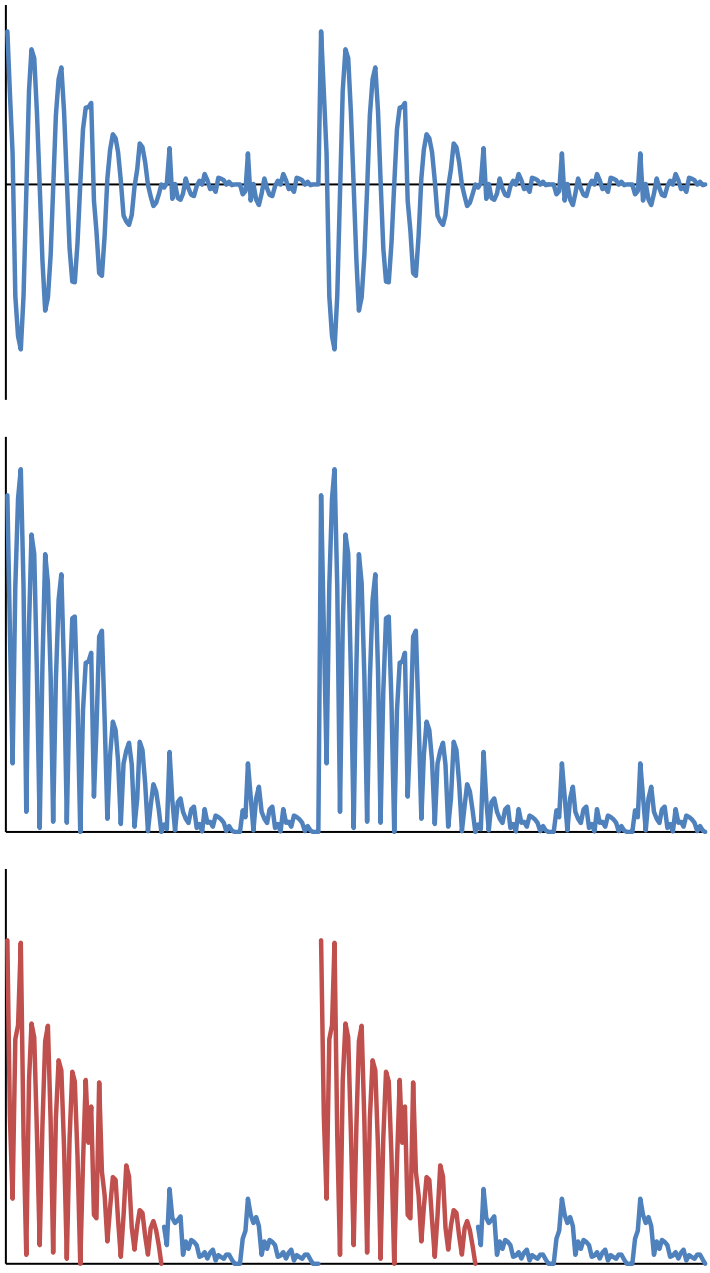

Fig. 4. The original waveform (top), absolute waveform plot (middle) and absolute waveform plot with colors denoting frequency (bottom). Red denotes a low frequency and blue denotes a high frequency.

where $w(t)$ is the signal amplitude at time t. Obvious benefits to this representation are that the differences between consecutive samples are doubled, and hence become more noticeable for the viewer (see Fig. 2 middle as opposed to Fig. 2 left). This difference enhancement is likely to contribute to better visual search in the waveform. Also, the proposed visualization requires just half the display real-estate compared to the traditional waveform with the same level of differences between consecutive samples since only half the height is needed when starting the sample at the axis, or half the width if the waveforms are displayed along the vertical direction. Yet, the proposed visual representation easily communicates the high and low intensity points in a signal. Thus, the simplified visualization facilitates simple search for particular feature points in the audio in a similar manner to traditional waveforms.

Further, to reflect the fact that we are looking at a waveform at a coarse-grained time scale, we convert the waveform w in the time domain t to the signal s in the pixel domain x as follows.

$$s(x) = \max(\mathrm{w}(t)), t \in [t_1, t_2] \tag{2}$$

Here t_1 and t_2 represent the start time and end time respectively, corresponding to a given display pixel along the time domain. The max function achieves a type of simple anti-aliasing effect. Fig. 3 illustrates the mapping.

Fig. 4 illustrates the proposed visualization in practice. Fig. 4 (top) shows a traditional waveform depicting an audio sample with a sequence regularly spaced hi-hats and two bass drums. Clearly, the hi-hats are a bit harder to spot than the two bass drums because of their different intensities. Fig. 4 (middle) shows the same waveform with the proposed visualization approach. We argue that this waveform makes the differences between the peaks and valleys more visible than the traditional waveform representation. The bottom plot in Fig. 4 shows the same waveform with frequency color coding. Here, red represents a low (warm) frequency and blue represents a high (cold) frequency.

3.1 DC offsets

One benefit of the traditional waveform display is that it allows DC, or direct current, signals to be easily spotted by the viewer such that they can be corrected and the sound quality improved. However, it is assumed that for most audio editing applications the audio is already DC-corrected, especially when working with third-party audio such as music in DJ software. Also, DC offset problems are probably less of a challenge with modern digital audio recording pipelines compared to analogue recording configurations.

Nevertheless, DC occurs naturally with certain sounds such as fuzz-guitars sounds. If working with audio that contains DC offsets, it is important to be aware of the DC levels especially when combining several sounds. We therefore propose to shift the signal according to the DC offset, namely

$$s'(x) = |s(x)| + DC(x) \tag{3}$$

where $DC(x)$ can simply be computed as a windowed average around the selected sample point, namely

$$DC(x) = \frac{1}{w}\sum_{i=-w/2}^{w/2} s(x+i) \tag{4}$$

Where w is the width of the averaging window. In other words, for each audio segment, the signal starts at the DC offset and ends at the peak. This concept is illustrated in Fig. 5.

The top plot in Fig. 5 shows the original waveform from Fig. 3 top offset with a sinusoidal time-varying DC as a traditional waveform. Fig. 5 middle shows the same signal using the proposed method. The shape of the DC offset is clearly seen from the bottom of the plot, and the peaks and valleys are visible on the top of the plot. Fig. 5 bottom shows the proposed method with frequency coding.

4 Conclusions

This paper has argued for an alternative, yet simple method for visualizing audio waveforms. The technique involves displaying the absolute value of the waveform in the simplest case. This magnifies the variations in the signal making it easier for users to search for particular cue points. Alternatively, the real-estate consumed by the traditional waveform plots can be halved without reducing the quality and magnitude of the audio signature. This is particularly useful for current digital audio processing software where the users may work with a large number of parallel audio tracks. The proposed technique is intended for working with coarse-grained time scales. If working at finer-grained time scales, it may be beneficial and more suitable to switch to the traditional waveform visualization. Such mode changes can easily be done automatically in an editor according to the selected time scale. In future work it would be interesting to explore the use of graph embellishments [31] to improve the communicative effectiveness of audio visualizations.

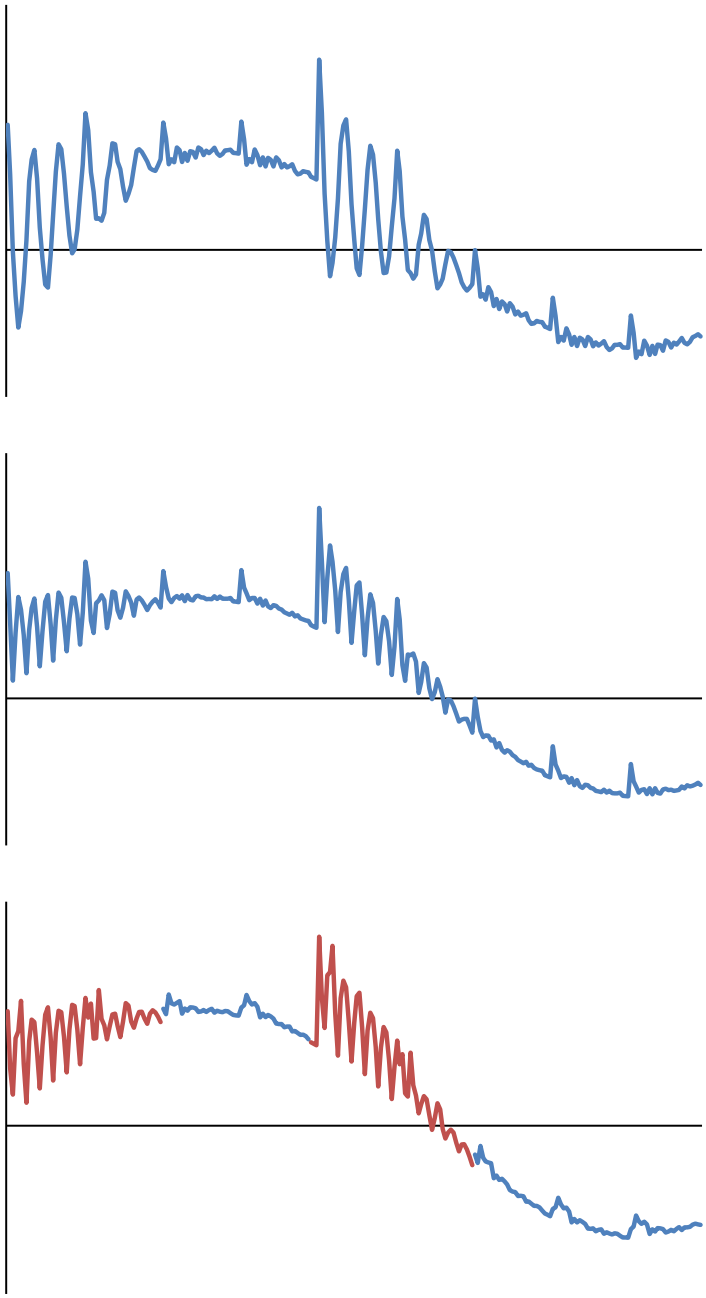

Fig. 5. The original waveform with DC-offset (top), absolute waveform plot with DC-offset (middle) and absolute waveform plot with DC-offset and colors denoting frequency (bottom). Red denotes a low frequency and blue denotes a high frequency.

References

1. Jian, H.-L.: On the tonal inventory of the Taiwanese language: a perceptual study. Journal of Taiwanese Vernacular 2, 28-50 (2010)
2. Hracs, B.J.: A creative industry in transition: the rise of digitally driven independent music production. Growth and Change 43, 442-461 (2012)
3. Roma, G., Xambó, A.: A Tabletop Waveform Editor for Live Performance. In: Proceedings of NIME, pp. 249-252 (2008)
4. Lopes, P.A., Ferreira, A., Pereira, J. A.: Multitouch interactive DJing surface. In: Proceedings of the 7th International Conference on Advances in Computer Entertainment Technology, pp. 28-31, ACM (2010).
5. Hansen, K.F., Bresin, R.: The skipproof virtual turntable for high-level control of scratching. Computer Music Journal 34, 39-50 (2010)
6. Copley, J.: Audio and video podcasts of lectures for campus-based students: production and evaluation of student use. Innovations in education and teaching international 44, 387-399 (2007)
7. Mazzoni, D., Brubeck, M., Haberman, J.: Audacity: Free audio editor and recorder. URL http://audacity.sourceforge.net (2005)
8. Serato, SERATO DJ 1.9.4 SOFTWARE MANUAL, downloaded from https://serato.com/downloads/files/145095/Serato+DJ+1.9.4+Software+Manual+-+English.pdf (2016)
9. Native Instruments, TRAKTOR manual, downloaded from https://www.native-instruments.com/fileadmin/ni_media/downloads/manuals/TRAKTOR_PRO_2_9_Manual_Englisch_2015_08.pdf (2016)
10. Luder, C.B., Barber, P. J.: Redundant color coding on airborne CRT displays. Human Factors: The Journal of the Human Factors and Ergonomics Society 26, 19-32 (1984)
11. Green, B.F., Anderson, L.K.: Color coding in a visual search task. Journal of experimental psychology 51, 19-24 (1956)
12. Sandnes, F.E.: On the truthfulness of petal graphs for visualisation of data. In: Proceedings of NIK 2012, pp. 225-235, Tapir Academic Publishers (2012)
13. Eika, E., Sandnes, F.E.: Authoring WCAG2.0-Compliant Texts for the Web through Text Readability Visualization. In: Proceedings of HCI International 2016, Universal Access in Human-Computer Interaction. Methods, Techniques, and Best Practices (eds: Margherita Antona and Constantine Stephanidis), LNCS Vol. 9737, pp. 49-58, Springer (2016)
14. Sandnes, F.E.: Understanding WCAG2.0 Color Contrast Requirements through 3D Color Space Visualization. Studies in Health Technology and Informatics 229, 366-375 (2016)
15. Andersen, T.H.: In the Mixxx: Novel digital DJ interfaces. In: Proceedings of CHI'05 Extended Abstracts on Human Factors in Computing Systems, pp. 1136-1137, ACM (2005)
16. Boersma, P.: Praat, a system for doing phonetics by computer. Glot international 5, 341-345 (2002)
17. Beckett, R.L.: Pitch perturbation as a function of subjective vocal constriction. Folia Phoniatrica et Logopaedica 21, 416-425 (1969)
18. Lin, K.H., Zhuang, X., Goudeseune, C., King, S., Hasegawa-Johnson, M., Huang, T.S.: Improving faster-than-real-time human acoustic event detection by saliency-maximized audio visualization. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2277-2280, IEEE (2012)
19. Misra, A., Wang, G., Cook, P.R.: sndtools: Real-time audio DSP and 3D visualization. In: Proceedings of the International Computer Music Conference, International Computer Music Association (2005)

20. Gómez, E., Bonada, J.: Tonality visualization of polyphonic audio. In Proceedings of International Computer Music Conference (2005).
21. Lartillot, O., Toiviainen, P.: A Matlab toolbox for musical feature extraction from audio. In: International Conference on Digital Audio Effects, pp. 237-244 (2007)
22. Gerhard, D.: Audio visualization in phase space. In: Bridges: Mathematical Connections in Art, Music and Science, pp. 137-144 (1999)
23. Foote, J.: Visualizing music and audio using self-similarity. In: Proceedings of the seventh ACM international conference on Multimedia, Part 1, pp. 77-80, ACM (1999)
24. Heller, F., Borchers, J.: DiskPlay: in-track navigation on turntables. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1829-1832, ACM (2012)
25. Heller, F., Borchers, J.O.: Visualizing Song Structure on Timecode Vinyls. In: Proceedings of NIME, Vol. 14, pp. 66-69 (2014)
26. Huang, Y.P., Lai, S.L., Sandnes, F.E.: A Repeating Pattern based Query-by-Humming Fuzzy System for Polyphonic Melody Retrieval. Applied Soft Computing 33, 197-206 (2015)
27. Kubat, R., DeCamp, P., Roy, B., Roy, D.: Totalrecall: visualization and semi-automatic annotation of very large audio-visual corpora. In: ICMI, Vol. 7, pp. 208-215 (2007)
28. Cannam, C., Landone, C., Sandler, M.: Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In: Proceedings of the 18th ACM international conference on Multimedia, pp. 1467-1468, ACM (2010)
29. Barras, C., Geoffrois, E., Wu, Z., Liberman, M.: Transcriber: a free tool for segmenting, labeling and transcribing speech. In: First international conference on language resources and evaluation (LREC), pp. 1373-1376 (1998)
30. Tzanetakis, G., Cook, P.: Marsyas3D: a prototype audio browser-editor using a large scale immersive visual and audio display. In: Proceedings of the 2001 International Conference on Auditory Display, pp, 250-254 (2001)
31. Sandnes F.E., Dyrgrav, K.: Effects of graph embellishments on the perception of system states in mobile monitoring tasks. In: Proceedings of CDVE 2014, LNCS Vol. 8683, pp. 9-18, Springer (2014)