Audio Engineering Society
Convention Paper
Presented at the 128th Convention
2010 May 22–25 London, UK

Track Displays in DAW Software: Beyond Waveform Views

Kristian Gohlke[1], Michael Hlatky[1], Sebastian Heise[1], David Black[1], and Jörn Loviscach[2]

[1]*Hochschule Bremen (University of Applied Sciences), D-28199 Bremen, Germany*

[2]*Fachhochschule Bielefeld (University of Applied Sciences), D-33602-Bielefeld, Germany*

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

ABSTRACT

For decades, digital audio workstation software has displayed the content of audio tracks through bare waveforms. We argue that the same real estate on the computer screen can be used for far more expressive and goal-oriented visualizations. Starting from a range of requirements and use cases, this paper discusses existing techniques from such fields as music visualization and music notation. It presents a number of novel techniques, aimed at better fulfilling the needs of the human operator. To this end, the paper draws upon methods from signal processing as well as computer graphics.

1. INTRODUCTION

Waveform views occupy a prominent space in the graphical user interface of digital audio workstation (DAW) software; often the entire interface is constructed around a central waveform view. The waveform display eases audio editing by serving a number of high-level goals such as visual track identification and the recognition of recording errors. The user of a DAW approaches these high-level goals through low-level tasks such as visual identification of note onsets and releases, passages with clipping, and changes in level over time.

We have collected a number of these and similar use cases that frequently occur when working with a DAW, including cases that are *not* supported by current commercial workstation software, such as identifying psychoacoustic masking between tracks, determining the tonality, finding similar parts on the same track or on different tracks, detecting hardly audible but potentially far-reaching technical issues such as subsonic components, phase cancellations, spotting errors in parallel takes of the same music, or checking the settings of dynamics processors and equalizers.

Apart from examining these use cases, we investigate display requirements. These comprise allowing a quick overview of a constantly increasing number of tracks, mapping the important features not to hue but to luminance, being able to scale the display both vertically and horizontally, and being able to precompute summaries to be stored alongside the actual audio files for quick visual reference and minimized overhead runtime.

A certain part of these mentioned use cases and requirements has already been addressed in research such as TimbreGrams [7] and The Sonic Visualizer [4]. In rare cases, standard audio software offers an additional spectrogram view or uses coloring to enhance waveform displays. We examine how these and other visualizations address the use cases and the requirements that we have found. Then, we widen the scope and propose solutions based on methods stemming from music information retrieval, computer graphics, and information visualization.

Waveform displays could be enhanced to include additional audio features other than merely the digital signal level. The selection of such features and their visual weighting in the display needs to emphasize those possible points of interest that reflect the user's mental model of an audio track and the task at hand. This can include clearly showing audio features such as perceptual sample onset, spectrum, timbre, and inaudible issues such as a DC offset.

Furthermore, we sought inspiration from other domains such as standard and contemporary music notation. Similar to music notation, waveform graphics may be scaled, locally compressed, or otherwise spatially optimized to allow a visual emphasis of the most relevant features or sample segments, or to display several tracks in one line. Text editing tools, a further source of inspiration, offer options such as highlighting, underlining, and folding.

Harnessing the processing power of today's processors and video cards, audio signals could be analyzed on the graphics card and visual elements could be precomputed to further enhance the speed and usability of such new display modes. Hence, the effects of dynamics processors, equalization, and other effects applied to tracks can be quickly displayed alongside or in place of the original waveform.

To evaluate our proposed enhancements and modifications to the traditional waveform display, we demonstrate a number of prototypes.

2. USE CASES

The traditional waveform display has arisen as a simple method to visualize the content of a recorded track. At the time the first DAWs were introduced, this was possibly the only computationally feasibly method. Given the power of today's computers, it is time to revisit the tasks supported and *not* supported by such displays. This results in clues toward more effective options for visualization. Looking at existing products and looking at regular tasks, we came up with a list of use cases that can be subsumed under six categories. These range from issues concerning the complete piece of music down to issues on the level of audio samples. This breadth of phenomena to be displayed in a single visualization already indicates the complexity of this visualization problem.

High-level and low-level structure. The user of the DAW needs to grasp the global structure of the project: How many tracks are there? How many are active? How long is the playing time? Are there sections and/or repetitions? Is there an overall temporal evolution? Loudness, timbre, and tonality can be of interest here; for a radio drama, transcribed speech may be added; a movie sound production may benefit from an added filmstrip or a visual analysis. A global view of a DAW project may also include audio material that is not currently in use, for instance additional takes. In regular DAW software, such material tends to be hidden.

Real-time orientation. Given the structure, the user has to know where he or she is in the project, in particular when listening to the resulting audio. The current position should be clearly visible and be equally effective even at different zoom levels.

Spotting the culprit. Currently, mistakes for instance in pitch or timing or technical issues such as

DC offset, infrasound, clipping, inverted phase, or booming bass sounds are mostly chased down by listening to the audio output. An advanced system, however, may also find such issues and indicate them visually. This may include an automatic comparison of different takes or repetitions.

Single-track audio adjustment. To facilitate adjusting dynamics compression, equalization, reverberation and other effects, a visualization can show the effects of such settings. This may include before/after comparisons and different type of analysis tools such as spectrograms. This is relatively straightforward for insert effects; a send effect such as a reverberation unit will require special treatment as its output signal contains contribution from several tracks. A standard use case is to automate the evolution of a parameter over time, such as the track's level, its panorama, or the gain setting of an equalizer. This is typically represented through curves superimposed on a waveform display.

Overall audio adjustment. Most current DAWs focus on adjusting single tracks or possibly also grouped tracks. An advanced track display may, however, also support overall audio adjustments. For instance, it may show the sum signal and/or how the different tracks enter with different loudness and different degrees of spectral masking. Visualizations focusing on the spectrum instead of the waveform may be beneficial here.

Audio editing. The raw audio material may be cut and pasted, working with a single track or any number of tracks. Such tasks would benefit from a visualization of natural boundaries in the audio material. Current systems automatically snap at zero crossings of the signal; future systems may snap to note onsets or even to parts that are found to be repeating. Onset detection is already part of existing solutions for flexible speed and pitch; it may also be used for rhythmic correction. Audio editing involves processing sections down to a handful of samples. Already today, advanced systems offer access to single notes within polyphonic recordings to change both timing and pitch [3].

Currently, waveforms are the main visualization employed to address the above tasks. Nonetheless, their shortcomings are obvious. For instance, onsets in a polyphonic signal or even in a monophonic legato line are poorly visible. It is hard to read off the *perceived* share of each track in the final signal; and it is virtually impossible to identify mistakes in pitch or to make a judgment about the timbre if the waveform display is not zoomed in to show a small number of periods of the waveform.

3. DISPLAY REQUIREMENTS

A number of general guidelines applies to almost any visualization. Considering the mass of data that could potentially be represented, Tufte's [35] principle of using the "least visual difference" is of particular importance. Using as few visual objects as possible and employing just noticeable differences for instance in line widths or solid colors can lead to highly economic visualizations that employ a human's visual bandwidth to the fullest. Note, however, that this principle contrasts with the heavy use of eye candy—such as shadows in waveforms and photorealistic representations of (actual or fictional) studio electronics—in most DAW software.

A major concern when allocating visual bandwidth is to focus the display on the aspects that are important for the task at hand. This could mean changing the display depending on the zoom level or on the controls currently used. Care has to be taken, however, not to destroy the user's mental model of the audio project when changing the visualization.

As the visualization potentially stretches from individual samples of, say, 192,000th fraction of a second to several hours of a Wagner opera, it runs into problems known from geographic visualizations that encompass scales from planet Earth down to a pedestrian's view. How does one navigate such a space? How does one orient oneself? To help with these issues, the display may present both detail and context [19] simultaneously, for instance through magic lenses [30]. For a classification of these types and of zooming interfaces see [6]. Depending on the scale, different visualizations may be offered—given that the user still senses enough consistency and stays oriented. Given the large amount of data, the optimal use of screen space is imperative. Standard waveform displays contain lots of empty space (such as silent tracks) and redundant data (such as waveforms that are mirror symmetric about their zero line).

Whether or not the visualization has to be “intuitively understandable” is debatable. First, the mere notion of “intuitive” is fuzzy: For a sound engineer trained on studio hardware or standard DAW software, other types of visualization will appear intuitive than for a musician-producer with little formal technical education. Second, professional tools can afford a certain amount of training in order to be the more effective. For instance, it is hardly conceivable to demand of a medical ultrasound device that the images it produces be “intuitively understandable.” This antagonism between initial ease-of-use for the untrained and effectiveness as well as efficiency for the trained may sometimes be overcome by expert functions hidden from the regular user. In any case, it should be taken care that each type of visualization satisfies the principle of “recognition rather than recall” [28].

A very specific issue of any visualization employed in a DAW is to create a consistent, easily learnable (if not “intuitive,” as discussed before) and effective mapping between sound and graphics. This concerns single parameters (Should waveform be represented by the stored values, their square, the logarithm of their square or neither of those?) as well as an overall design. The type of mapping may be classified [24] as acoustical (based on the physics, subsymbolic (based on the human hearing system), or symbolic (based on the manipulation of symbols). To quote Giannalis and Smith: “Visual representations of sound such as time-domain and frequency-domain representations are based on physical approaches to sound understanding and cannot be used as intuitive conceptual metaphors for sound design.” [11]

Music typically contains many repetitions, slight variations and gradual changes. Hence, the mapping between sound and graphics may best be fuzzy in order to emphasize perceived similarities. If it had discontinuities, a slightly different repetition of the same musical phrase would otherwise be visualized too differently. This could happen for instance with chords being recognized and displayed as text: The chord that was classified as Am7 on its first occurrence may become C6 the next time, due to a slight change in the relative levels of the constituent notes. Visualizing recorded music through a classical score leads to even more ambiguities that could be resolved in one way or another. A fuzzy visualization may indicate the certainty of a recognition. The level of uncertainty in the detection could influence the visual representation, for instance by a change in saturation.

An obvious, but momentous technical requirement on the visualization is that must react in real time. Given the computational load from audio processing—which has to run with the least possible latency, yet free of glitches—the use of processing-intensive real-time audio analysis for the sole purpose of enhancing the user interface seems counterproductive. The processing power of modern computer systems continuously increases and can provide the basis for additional computations, in particular as audio analysis lends itself well for parallelization on multicore processors.

Still, efforts should be made to use the available resources with great efficiency. Current audio workstation software provides much room for improvement. To lower the computational load, users can “freeze” audio tracks, meaning that all effects and other edits applied on the track are being pre-rendered and can then be played back as a simple audio stream that is gentle on resources. This process can be automated to a certain extent by freezing such tracks automatically that have not been modified for a certain period of time. This may also be done on a smaller scale by automatic pre-rendering of short subsections of a track. Another option that enables heavy-weight computing tasks without putting significant additional load on the CPU has been demonstrated in previous work by some of us [20], where the graphics card is used for audio processing. This approach makes use of processing capabilities that are rarely exploited to their full extent in audio applications.

4. EXISTING METHODS

Seeking inspiration on methods for visualizations, we looked into existing tools and research projects concerning audio editing and retrieval as well as into visual representations outside this domain.

4.1. Visualization for Audio Editing and Retrieval

Commonly used waveform displays are rarely used up to their full potential. Few products demonstrate a slightly more flexible approach to waveform displays. Audio editing software may optionally represents the waveform logarithmically [2] to better indicate dynamics. A complete overview of an entire track and an additional zoomed-in view that scrolls in sync with the music may be presented in parallel [27]. The before/after effect of a dynamics processor can be shown by overlaying the two corresponding waveforms [18]. This is related to display and edit MIDI notes with the help of their actual waveforms [21].

Waveform displays are poor at indicating the frequency content of a recording. This issue has been addressed in two ways: First, sonograms [2] and related methods [4] can be used to display the content of an audio recording almost like a piano-roll visualization. Using a deconvolution in the Fourier domain with logarithmic frequency, fundamental frequencies and harmonic patterns can be separated well [32]. Second, one can use color patterns to indicate the timbre. To this end, acoustic features as known from music information retrieval can be extracted and mapped to color [7]. Even basic acoustic features deliver promising results [26].

A blended approach is to color waveforms according to the current spectrum, for instance by mapping high frequencies to the red color channel, midrange frequencies to the green color channel, and low frequencies to the blue color channel, as proposed by [31]. (Note that this runs counter to the regular definition of blue = high frequency, as in visible light.) Such visualizations can already be found in commercial DAW software [9].

Ways to map parameters such as the loudness and the pitch of pure tones to colors have been examined in a number of basic studies. One of these finds [12] that a natural mapping is to depict loudness by color saturation and pitch by color brightness.

On a higher level, the frequency content of an audio file can be used to deduce chords and/or deduct the evolution of tonality. A standard approach is to assign each frequency band of a spectral analysis to one of twelve bins that represent the notes C through B, irrespective of the octave. The histogram thus generated can be compared to empirical patterns found in music with known tonality. This may be visualized [14] through a cloud on a torus, through color intensity in parallel lanes, each representing a key, and though a stripe of colors each showing the current most probable key through its hue (one octave mapped around the color wheel), possibly using several parallel lanes with different time scales ranging from the complete piece of music to single beats.

The recognition of tonality is a major device to segment an audio recording automatically and to find repeated sections such as a chorus [17]. Other acoustic features to be used for this task comprise mel-frequency cepstral coefficients and different kinds of beat spectra. Often, the result is visually presented through the similarity matrix [8, 29]: a square image that shows how related certain sections are. Sections that are similar to each other stand out as squares in the diagram. Whereas most proposed segmentation methods required no user intervention, it may be beneficial to allow the user to mark two regions as different [25]. Automatic segmentation is already being applied in sound editing software [7]. It looks logical to combine this with the modes of typical sequencing software in which blocks of audio can be shuffled around.

The positioning of sound sources in a mix lends itself to a geometric visualization. This is employed for instance in 2D for wavefield synthesis [22] as well as in 3D, with other parameters being added to the visualiziation [13], possibly even indicating sound quality [10]. The motion of a sound source may be visualized through an animation—which requires time to run—and/or a graphical path—which does not clearly indicate the connection to the overall temporal structure. This in contrast to waveform displays and spectrograms, which allow to grasp the temporal evolution quickly.

4.2. Music Notation as Visualization

A classical score can be considered a form of visualized audio. Orchestral music often contains “track” information for a large number of instruments, voices, and sound effects of a wide range of timbres and playing styles. By applying techniques from both traditional and modern music notation, track displays in DAWs may be further refined to display pertinent and useful information at once.

Traditional music notation for orchestras is divided into parts for the individual players and those for the conductor. The parts of the individual players contain notes to be played, as well as instructions on how to approach timbre, dynamics, instrument-specific playing techniques, and location cues such as those for repetition or jumping to other points in the score. These parts are often meant to be read by a single player, but are often combined with those of other related instruments, such as French horns or violins divided into multiple subsections.

The horizontal arrangement of the notes on the page is often related to the length of each note—but only rarely not to an extent that the visual space that a note occupies would be proportional to its acoustic length. Because a note head occupying a set width on paper may represent note lengths ranging from a few millisecond to many seconds, visual compression and expansion occurs when the score is arranged. Accordingly, a line of mainly long notes may represent up to several minutes, whereas a line of short notes may represent only a few seconds. When reading the score, the instrumentalist is then only forced to rapidly scan lines of notes when necessary. In addition, short notes are signified by more note flags, while longer notes have less flags. This permits a quick glance of the business of the music in a certain section. In short, more ink on the paper denotes more activity. In addition to note lengths and size, dynamics and other marking permit the player to immediately conceive and anticipate changes in the way the music should be played.

The score for the conductor combines all of the individual parts into one, to give him or her an overview of everything that is occurring at each point of time. Therefore, for scores containing a large number of instruments, information must often be combined. Instruments with similar scores are frequently combined so that the conductor needs only to glance at one line of notes to comprehend the actions of a group of instruments.

Concepts from music notation may help to design future DAW track displays. First, similar tracks may overlaid, just as conductor's scores combine the scores of similar instruments. In this way, the editor may easily obtain a feeling for the arrangement of a project and editing by allowing for repetitive actions to be performed simultaneously on similar tracks. Second, both single and individual tracks may be zoomed horizontally according to the amount of activity in the tracks in question. When music or effect changes occur at a slow rate, less track information might be shown on the screen to reduce side scrolling and excessive motion of the editor's eyes. More changes in the material would demand that more be shown on screen, although only when necessary. Third, the symbols used in music notation might also be adapted for DAW track displays. Because dynamic and timbre information are sometimes unclear using traditional waveforms, such markings from traditional notation might help supplement track displays so that editors could acquire timbre and dynamic information without even needing to view the waveforms.

Finally, concepts from more modern music notation and representation [33] may be advantageous. Most of these seem to converge toward abstract forms of spectrograms. Hence, it is only consequential that some software for music creation starts from 2D images that resemble spectrograms [36]. One may even apply 3D computer graphics to display additional data such as channel numbers [5]. A more abstract representation is employed in Rainer Wehinger's visualization of György Ligeti's composition Artikulation [37]. Here, similar sounds from Ligeti's electronic sound creation are notated as droplet-shaped comets, clouds of dots, and shades of color. Such a representation may also be of use to the DAW editor to visually categorize and discern between various sound sources.

4.3. Other Uses of Visualizations

Text editing software—as mundane as they seem to be on first sight—are optimized for efficiency. It is enticing to carry over the achievements made in this domain to DAW tools. This may concern features such as underlining potential errors, folding away details of a document, and keeping track of the editing history, for instance through marked additions and removals.

Most video editing tools share the timeline view of DAW tools. The usual thumbnail filmstrips, however, provide less orientation than do the waveforms in audio software. This is in part due to the bad temporal resolution of the thumbnails; there is only one thumbnail for each, say, 64 pixels on the timeline. And it is part due to the poor legibility of the thumbnail's contents; it may even be hard to tell the start of a clip from its end. Current video editing

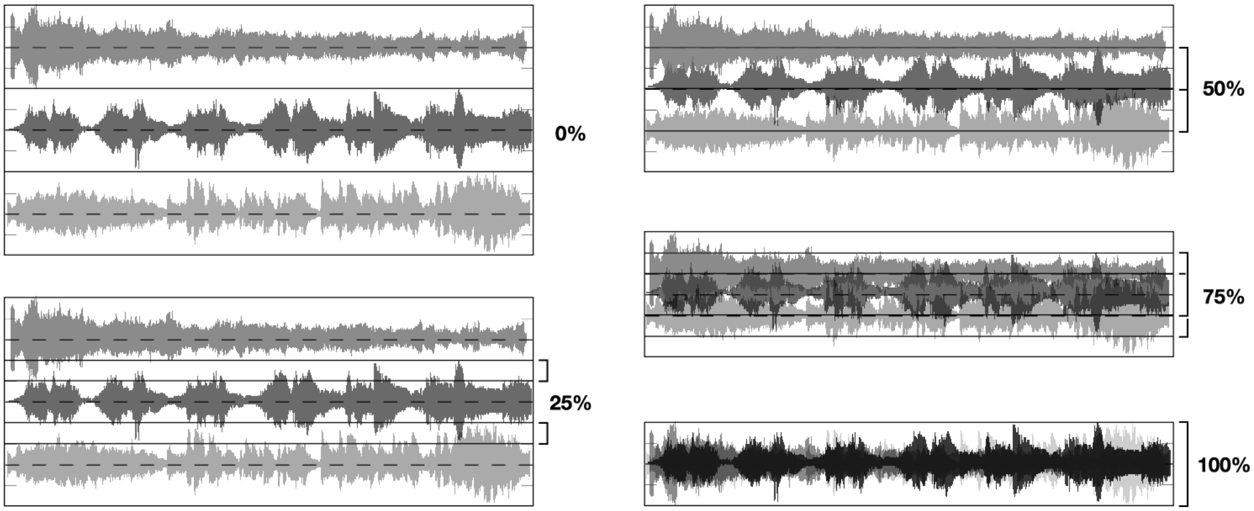

Fig. 1: Transparent overlays of several waveforms.

software tries to make up for this by speech recognition [1, 16]. A kind of captions would have obvious use for radio plays, podcasts, and other audio productions done on a DAW.

5. PROTOTYPES

We have created several prototype visualizations in an effort to find novel approaches for conveying information about the content of time-line based track displays. For this purpose, we have developed software that creates visualizations based on an analysis of the audio material. In addition, we have designed a number of screen mock-ups to discuss more visionary ideas.

5.1. Saving Screen Space

Whereas the size and the resolution of workstation displays increase steadily, a trend towards smaller form factors for mobile devices can also be observed. Many of these devices, such as netbooks or smartphones are capable of basic audio processing. Therefore, it seems relevant to develop space-saving, yet expressive visual user interfaces. Standard waveform displays, however, occupy a significant amount of screen real estate. To a certain extent, this might be tolerable on larger screens, but even there it is symptomatic that users tend to fold away and minimize tracks to free up space. The same interface on a smaller display would require excessive folding in order to minimize scrolling distances and to display at least a minimum number of tracks on the same screen.

Hence, it is imperative to use screen space more efficiently. This could for instance be done by sharing some space between adjacent tracks. At most coarse zoom levels, the outline of a waveform creates a comparatively simple shape. Several of such shapes can be presented with a certain amount of transparent overlap without losing any of their informational value. We have tried to identify an appropriate amount of overlap through a visual comparison chart, such as the one shown in Figure 1. This indicates that overlap rates of 25 percent and even more do not dramatically affect the display quality. Overlaps up to 75 percent are still legible on color displays, but should only be used with care. We envision the use of track overlaps as an alternative to visually scaling the entire interface. This method enables a different type of scaling where the size of the relevant information remains the same, while only the underlying layout grid is scaled. Overlapping waveforms could not only be used to save space on smaller screens, but also to achieve larger waveform displays on bigger screens without having to change the absolute height of the tracks.

Another way to layer tracks gets by with no transparency. It is based on the notion that the upper and

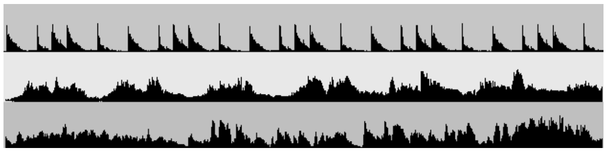

Fig. 2: Audio power displays without negative values.

the lower half of most waveforms are very similar at coarse zoom levels. If the waveform is mainly used to get a broad overview of the musical structure rather than for sample-precise editing tasks, the lower half does not necessarily have to be visible. This method allows overlap rates of up to 50 percent. However, a slight margin between the center line of a track and the top of the overlying track improves legibility. The mock-up prototype shown in Figure 2 uses an overlap rate of 45 percent.

5.2. Focus and Context

A bird's-eye overview of a DAW project can be maintained, while a detailed local view is provided simultaneously. For this purpose, we magnify the area left and right of the current playhead position and dynamically slide this window of heightened zoom along the audio tracks, see Figure 3. This view can be used for precision editing tasks. However, the scaling factor of the virtual lens is limited by a combination of factors. When the difference between the local zoom and the high level scaling becomes too large, the content highlighted by the lens might be no longer legible during playback as it dashes over the screen very quickly. Furthermore, the lens visually compresses a significant part of the track on both sides of the playhead. Thus, while emphasizing the immediate area around the playhead, the lens leaves less space for the remaining part of the track. This renders the approach less useful for mere playback, where a high-level overview of the area close to the playhead is often crucial. Still, this visualization can support sample-precise scrubbing and editing.

5.3. Audio Effect Visualization

In the spirit of the existing waveform displays [18] that visualize the gain reduction introduced by a compressor, we have created a software prototype that visualizes the tail of an artificial reverberation processor. The system employs a VST host to load

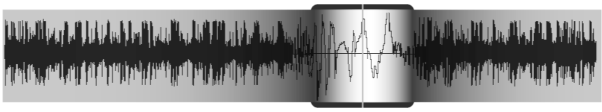

Fig. 3: A lens model creates an area of heightened detail around the playhead.

Fig. 4: Graphical overlay of an artificial reverberberation tail on the original waveform.

any reverberation plug-in and extracts the decay curve of the processor's impulse response. From this and the actual waveform'e envelope, it computes a transparent overlay image for the regular waveform display. Hence, the "smearing" through the introduced reverberation becomes visible, see Figure 4.

5.4. Contribution to the Sum

In a multitrack editing session it is often of interest to see how much a given track affects the overall result. This can be depicted by showing the final mix as a stacked diagram, see Figure 5. The overall height of the stack graph can be handled dynamically. As the user changes the level, the equalizer settings, or the effect parameters of a given track, the stack diagram is dynamically adapted to reflect the new constellation. Instead of displaying the relations between the mere audio power values of each track, such a method should account for the *perceived* loudness of each track in the mix.

5.5. Spectral Contribution

Apart from a percental contribution to the audio sum, as described above, each individual track's contribution over time in frequency space can be computed as well. The resulting image created by our software prototype resembles a spectrogram, even though it is limited in its resolution as it only displays two discreet states for each individual track.

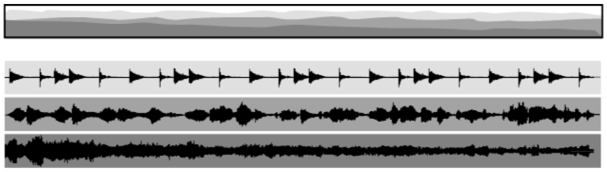

Fig. 5: Stack diagram showing the perceived loudness of each track in the mix.

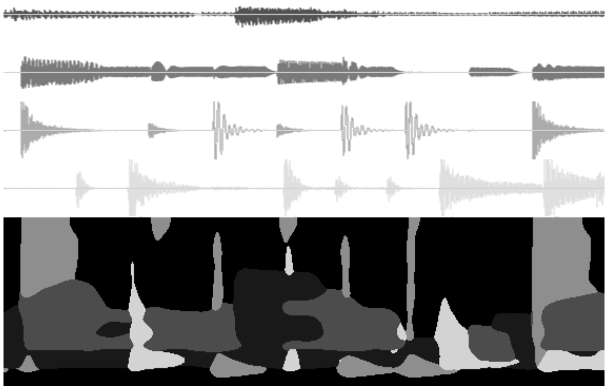

Fig. 6: Visual abstract representation of the contribution of four individual tracks to the spectrum of a mix.

As soon as one track's contribution to a certain part of the spectrum exceeds a user-definable threshold, the area of the spectrum is colorized in the track's associated color. In case the contribution of two or more tracks in the same part of the spectrum exceeds the threshold, only the color associated to the track with the highest contribution is drawn, thereby resembling a probable masking effect. The resulting visualization, see Figure 6, enables an easy understanding of the spectrum-based contribution of a number of single tracks to the mix. This can be used to create mixes in which the spectrotemporal overlap of the constituent tracks is minimal [23, 34] or the masking is minimized by appropriate panning [15].

6. CONCLUSION AND OUTLOOK

We have demonstrated a collection of novel approaches to provide a better visualization of the audio content. A series of functional and mock-up prototypes has been developed to examine different visualizations. The large amount of space that is occupied by waveform displays in current software products can be used to provide meaningful data about an audio track, data that goes beyond the bare audio power. This requires the use of techniques from computer graphics and the use of advanced audio analysis methods.

In most cases, the focus should not be on delivering the technically most accurate picture, such as for instance by showing a detailed spectrogram, but rather to remove most of the clutter and provide the user with a mental model of the acoustic content that is easy to grasp. Foremost, the visual track display should be based on the perception of the sound rather than the technical features. Nonetheless, the user should be able at any time to move toward a fine level of detail if a task so requires. Any data that is available at the time of editing a track—such as the difference before and after applying audio effects—can be harnessed for visualization.

Future work may address the display of similarity: Similar segments of audio should be presented visually similar. Which of the established methods is best to determine similarity may depend on the level of detail, ranging from single notes to movements of a symphony. A deep question concerning the user interface remains open: Should there be a range of different visualizations that are specialized for certain tasks and/or for certain levels of detail—or should one reach a compromise to cover all tasks and all levels of detail by one visualization? Regular waveform-type displays implement the latter approach, but rather by accident than by deliberate decision.

7. REFERENCES

[1] Adobe Sytems Inc. The future of video is searchable. `http://www.adobe.com/products/creativesuite/production/videosearch/`, accessed 2010-03-09.

[2] Audacity Developer Team. Audacity. `http://audacity.sourceforge.net/`, accessed 2010-03-09.

[3] Celemony Software GmbH. Melodyne. `http://www.celemony.com/`, accessed 2010-03-09.

[4] Centre for Digital Music, Queen Mary, University of London. Sonic Visualizer. `http://www.sonicvisualiser.org/`, accessed 2010-03-09.

[5] A. Chaudhary, A. Freed, and L. A. Rowe. OpenSoundEdit: An interactive visualization and editing framework for timbral resources. In *Proceedings of ICMC '98*, 1998.

[6] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41(1):1–31, 2008.

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

[8] M. Cooper, J. Foote, E. Pampalk, and G. Tzanetakis. Visualization in audio-based music information retrieval. *Comput. Music J.*, 30(2):42–62, 2006.

[9] C. Corp. Comparisonics waveform display. `http://www.comparisonics.com/`, accessed 2010-03-09.

[10] O. Delerue. Visualization of perceptual parameters in interactive user interfaces: Application to the control of sound spatialization. In *Proceedings of the 120th Convention of the AES*, 2006. Paper No. 6656.

[11] K. Giannakis and M. Smith. Auditory-visual associations for music compositional processes: A survey. In *Proceedings of ICMC 2000*, pages 12–15, 2000.

[12] K. Giannakis and M. Smith. Imaging soundscapes: Identifying cognitive associations between auditory and visual dimensions. In *Musical Imagery*, pages 161–179. Swets & Zeitlinger, Lisse, NL, 2001.

[13] D. Gibson. *The Art of Mixing: A Visual Guide to Recording, Engineering, and Production.* Artistpro, Boston, MA, 2nd ed edition, 2005.

[14] E. Gómez and J. Bonada. Tonality visualization of polyphonic audio. In *Proceedings of ICMC 2005*, 2005.

[15] E. P. Gonzalez and J. Reiss. Automatic mixing: Live downmixing stereo panner. In *Proceedings of DAFx-07*, pages 63–68, 2007.

[16] Google Inc. Automatic captions in YouTube. `http://googleblog.blogspot.com/2009/11/automatic-captions-in-youtube.html`, accessed 2010-03-09.

[17] M. Goto. A chorus section detection method for musical audio signals and its application to a music listening station. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1558–1794, 2006.

[18] Harrison Consoles. ESP waveforms. `http://www.harrisonconsoles.com/joomla/index.php?option=com_content&task=view&id=42&Itemid=53`, accessed 2010-03-09.

[19] H. Hauser. Generalizing focus+context visualization. In G.-P. Bonneau, T. Ertl, and G. M. Nielson, editors, *Scientific Visualization: The Visual Extraction of Knowledge from Data*, pages 305–327. Springer, Berlin/Heidelberg, D, 2006.

[20] S. Heise, M. Hlatky, and J. Loviscach. Aurally and visually enhanced audio search with soundtorch. In *Proceedings of CHI '09*, pages 3241–3246, New York, NY, USA, 2009. ACM.

[21] S. Heise and J. Loviscach. Waveforms, not bricks: a visually enriched MIDI editor. In *Proceedings of Audio Mostly 2009*, pages 53–56, 2009.

[22] Iosono GmbH. Iosono spatial audio workstation plug-in. `http://www.iosono-sound.com/technology/hardware-and-software/`, accessed 2010-03-09.

[23] P. Kleczkowski and A. Kleczkowski. Advanced methods for shaping time-frequency areas for the selective mixing of sounds. In *Proceedings of the 120th Convention of the AES*, 2006. Paper No. 6718.

[24] M. Leman. Symbolic and subsymbolic description of music. In G. Haus, editor, *Music Processing*, pages 119–164. Oxford University Press, Oxford, UK, 1993.

[25] M. Levy and M. Sandler. New methods in structural segmentation of musical audio. In *Proceedings of EUSIPCO 2006*, 2006.

[26] 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.

[27] Native Instruments GmbH. Traktor Pro. `http://www.native-instruments.com/en/products/dj/traktor-pro/`, accessed 2010-03-09.

[28] J. Nielsen. Enhancing the explanatory power of usability heuristics. In *Proceedings of CHI '94*, pages 152–158, 1994.

[29] J. Paulus and A. Klapuri. Acoustic features for music piece structure analysis. In *Proceedings of DAFx-08*, pages 309–312, 2008.

[30] R. Rao and S. K. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceedings of CHI '94*, pages 318–322, 1994.

[31] 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.

[32] 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.

[33] K. Stone. *Music Notation in the Twentieth Century: A Practical Guidebook.* W.W. Norton & Co, New York, NY, 1980.

[34] A. Tsilfidis, C. Papadakos, and J. Mourjopoulos. Hierarchical perceptual mixing. In *Proceedings of the 126th Convention of the AES*, 2009. Paper No. 7789.

[35] E. R. Tufte. *The visual display of quantitative information.* Graphics Press, Cheshire, CT, 2nd edition, 2001.

[36] U&I Software llc. Metasynth. `http://www.uisoftware.com/MetaSynth/`, accessed 2010-03-09.

[37] R. Wehinger. *Hörpartitur zu György Ligeti: Artikulation.* B. Schott's Söhne, Mainz, DE, 1970.