VISUALAUDIO-DESIGN – TOWARDS A GRAPHICAL SOUNDDESIGN

Lars Engeln
Chair of MediaDesign
Technische Universität Dresden
Dresden, Germany
lars.engeln@tu-dresden.de

Rainer Groh
Chair of MediaDesign
Technische Universität Dresden
Dresden, Germany
rainer.groh@tu-dresden.de

ABSTRACT

VisualAudio-Design (VAD) is a spectral-node based approach to visually design audio collages and sounds. The spectrogram as a visualization of the frequency-domain can be intuitively manipulated with tools known from image processing. Thereby, a more comprehensible sound design is described to address common abstract interfaces for DSP algorithms that still use direct value inputs, sliders, or knobs. In addition to interaction in the time-domain of audio and conventional analysis and restoration tasks, there are many new possibilities for spectral manipulation of audio material. Here, affine transformations and two-dimensional convolution filters are proposed.

1. INTRODUCTION

After Pierre Schaeffer's first experiments with bouncing records as a representative of the *musique concrète*, the availability of magnetic tapes glued together enabled later composers, such as Karlheinz Stockhausen, John Cage, or Edgard Varèse, to create loops. In the 1970s, spectral music was pioneered as a compositional technique using computer-aided analysis of acoustic music or artificial timbres at IRCAM with the Ensemble *l'Itinéraire* by composers such as Gérard Grisey and Tristan Murail. Murail himself has described spectral music as an aesthetic, not a style - not a set of techniques, but an attitude [1].

Nowadays, spectrograms are frequently used to analyze audio material. After applying audio effects, the spectrogram can illustrate the implications of the manipulation, for instance. However, processing and analysis tools are ordinarily separated. Although, manipulations within the visualization of analysed audio data allow a more comprehensible sound design (cf. [2]).

Popular spectrogram manipulations are time-scaling, transposition, restoration and compression. Apart from that, visual manipulation can also be used for more advanced spectral processing [3]. Especially since small spectral modifications are not perceived as unnatural or synthetic [4].

A spectrogram that is interpreted as a pixel-based representation can be manipulated with image processing to achieve a *Visual-Audio-Design* (VAD). This is an opportunity for a more comprehensible sound design, in contrast to directly editing the parameter of DSP algorithms.

According to Klingbeil [3], the following challenges occur in spectral editing:

- **Time and cost** – The analysis of signals must be computed in the shortest possible time. By efficient implementation, for example the STFT, and an increase of the computational power this goal is reached.
- **Synthesis problem** – Processed signals from the frequency domain must be transformed back for playback.
- **Control problem** – Exciting music is dynamic and consists of many finely tuned frequencies. Processing must therefore be equally finely granular.
- **Compositional problem** – In addition to subsequent editing, there are approaches to compose music directly in the frequency domain.

This creates new demands both on the software and on the composer, who has to deal intuitively with the composition of individual frequencies and their magnitudes.

There are tools and papers concerning generally visual manipulation of the frequency-domain [5, 3] and the timbre design [6]. Our research addresses the Control and Compositional problem. Thus, a workflow for sound *designers* to creatively manipulate and generate sounds is described.

1.1. Coherence of Visual and Auditory Perception

In recent years, studies have shown that the combination of auditory and visual stimuli can change and even improve human perception (see [7, 8]).

In particular, it was shown that multisensory convergence exists in low sensory processing phases [7] and exists for visual-auditive stimuli (see [9, 10]). Convergence does not only occur after extensive processing in unisensory brain regions. This low-level processing of coherent sensor inputs allows the improvement of visual and acoustic perception by simultaneous matching stimuli. Based on the early convergence it can be assumed that acoustic and visual stimuli have a positive effect on each other [11]. For example, multimodal objects were detected faster and more accurately than unimodal objects [12].

Furthermore, the relationship between color and sound was investigated with an empirical approach [13], as existing mappings were often inconsistent and unfounded. The works perform a mapping of tonality, loudness and timbre to hue, saturation and brightness in different constellations. Although, there is a strong correlation between loudness and saturation as well as tonality and brightness (cf. [13]).

2. RELATED WORKS

Spectral editing was often implemented as a kind of painting program (cf. [14]). It started early with *SpecDraw* [2], at which frequencies were filtered (rejected) with an eraser and rectangular selection and transformation of the frequency-domain. In *AudioSculpt* [5] polygonal and freehand selections provided more freedom for damping, enforcing, and duplicating spectrals, as well as for time-stretching [15]. *TAPESTREA* [16] made it possible for the first time to create sound spaces from different audio sources. *SPEAR* [3] abstracts the frequency-domain and uses a sparse representation of audio. Therefore the manipulation is mediated by a vector visualization (reprasentation of partials with lines) and not by a pixel visualization (sonagram) like the other works. Prehearing is done by STFT, and when exporting files in higher resolutions it is possible to choose between different implementations, such as the McAulay-Quatieri method [17] or oscillator banks. In *MetaSynth* image data is used as input besides the freehand drawing of shapes within the spectrogram [18]. Moreover, filters and transpositions with metaphorical manipulations of the frequency-domain with fluids [19] and with virtual reality [20] are proposed.

All works have a collection of user interface elements and interaction possibilities. Free zoom levels in time and frequency axis allow the user both a quick overview of the entire spectral space and the possibility to edit temporal details. A tool palette usually allows switching between different modes for the selection and modification of magnitudes.

Besides that, with *Sound Mosaics* [21, 22] a system for sound synthesis via influencing graphical variables was created. Thereby, timbre is described by three parameters: sharpness, compactness and roughness (sensory dissonance). Compactness classifies signals on a scale between complex tones and simple noise. The visualization uses color (hue, saturation and brightness), as well as texturial structure. In addition, there is the research field of *soundtextures* (compare [23]), which also describes a graphical synthesis that is sometimes done in the frequency-domain.

Moreover, the *Sonic Visualizer* [24] is a program for analyzing audio signals. The program has an external API and a plugin infrastructure. Thus, for example, psychoacoustic parameters can be calculated and various features like a beat-detection can be extracted. Also, the *ArtemiS SUITE* enables sound and vibration analysis in an industrial context and displays psychoacoustic parameters according to Zwicker and Sottek's hearing model.

3. VISUALAUDIO-DESIGN

Our VisualAudio-Design (VAD) is written in modern C++ and is based on libCinder using FFTW for analysis/resynthesis and openCV for two-dimensional convolutional spectral processing. Therefore, it is cross-platform compileable, but mainly it is developed for Windows.

VAD allows the user to load, analyze and play natural sounds as audio files in a canvas (see Figure 1). Through the analysis (FFT) the overtones (or generally the frequencies involved in the sound) can be visualized.

In addition to natural sounds, images of any type can also be loaded and converted into sound (phase estimation + iFFT). This allows graphical structures to be used as sound material.

The central part is the SpectralCanvas (a sonagram), in which sounds and effects can be freely transformed as nodes in groups and layers.

A sound object can be moved freely in the SpectralCanvas. A horizontal shift in time repositions the sound object and a vertical shift transposes it. Rotation can also be performed. A slight rotation causes a gradual glissando of the entire spectral range of the sound object. With strong rotations (approx. 90°) overtones become transients. Thereby, a sound rich in overtones becomes an impulse-like clicking (see Figure 2). Timestreching can be performed by stretching the node along the time axis. Stretching along the frequency axis contracts or expands the harmonics.

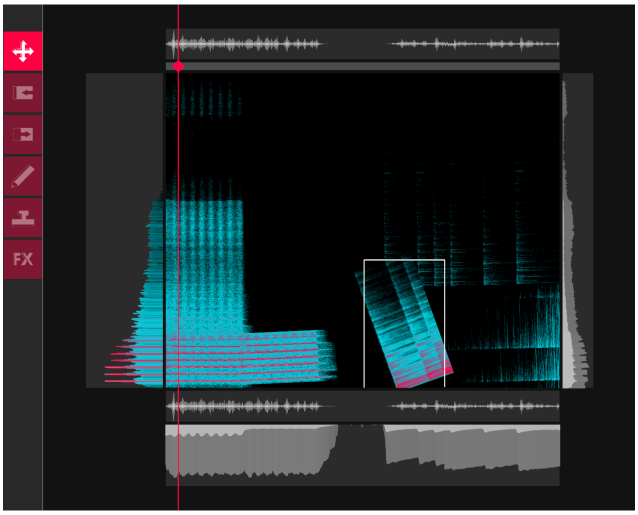

Figure 1: *Workspace of the VisualAudio-Design: the SpectralCanvas with its nodes consisting of sounds and effects (center), surrounded by widgets, and a toolbar (left).*

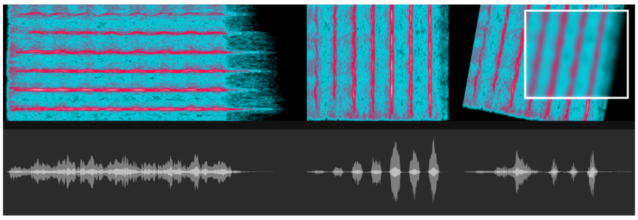

Figure 2: *Sound-nodes can be freely transformed. Thereby, a flute (left), is rotated by 90° (middle) or less (right). Effect-nodes can alter a certain area (see white rectangle (right) with blurring).*

In addition, effects equivalent to image processing can be used to distort the frequency domain to further alter the material. In this way, for example, a blur filter can be used to optically smooth the spectral space and thus audibly noise out the sound. The advantage over conventional audio effects over time is that individual frequency bands or individual overtones can be specifically assigned to an effect.

In this way, the VAD is suitable for creating sound surfaces as collages for creative use. A composer or sound designer is not limited to the mere temporal arrangement of sounds. New ways for musical expression can be found through transposition, rotation, overtone-exact effects, the use of graphical structures, and also by cutting & re-inserting individual frequency ranges/overtones.

3.1. System Design

The system is based on a scenegraph (the SpectralCanvas) of nodes and is highly modularized for extending with new features. At the same time, it is highly parallelized in order to guarantee a smooth

UserExperience with fast calculations in background worker. Basically, there are modules providing tools for the user to create or modify nodes in different ways, and widgets that can visualize additional parameters besides to the sonagram of the SpectralCanvas.

A tool is created as a module (inherits from ModuleBase-class) and is registered at the ModuleManager. This also creates the entry in the GUIManager for the toolbox. The active tool receives all interaction events that are performed with it on the SpectralCanvas. The tool also has direct access to the nodes of the SpectralCanvas and its spectral data. Thus a tool directly manipulates the nodes. A node can own spectral data itself (sound-node) or manipulate underlying spectral data in the hierarchy as effect-node. If a node has been changed, the corresponding layer and all layers above it are updated by the SpectralCanvas and are processed by the SpectralEngine. The SpectralEngine always keeps the time and frequency domain consistent with each other.

Multimodal input (mouse, key and touch at the moment) is preprocessed. Afterwards, the input event is recognized in the InteractionManager as gesture and propagated as interaction event. When starting an interaction, the GUI is handled first. If the GUI is not involved, the event is passed on to the WidgetManager. In this very moment the SpectralCanvas is treated as Widget. If the current interaction takes place on the SpectralCanvas, the event is passed to the active module. However, when interacting on a actual widget, the event is processed in the corresponding widget. If no one handles the event, so the interaction is done somewhere else in the workspace, then standard interactions for zooming and panning to navigate in the workspace are done.

3.1.1. Tools and Interaction

Each module provides a tool for the user, to manipulate the nodes or directly the frequency-domain of the SpectralCanvas. Several modules have been implemented but many more suitable modules are possible. Description of touch gestures according to *GeForMT* [25] are shortend in the following. For instance, a two finger swipe gesture is shortened to $2F_{swipe}$ and a gesture in which one finger holds while another finger performs a circular movement is shortened to $1F_{hold} + 1F_{circle}$.

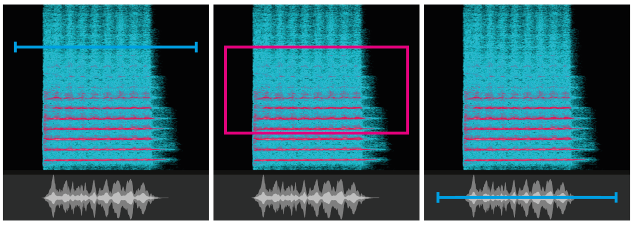

Figure 3: *SaveModule: selecting a time range (left) or an area of spectrals (middle) in the SpectralCanvas, or selecting a time range in the waveform widget aswell (right).*

- **load & save** – Loading of any audio and image files and simultaneous positioning by clicking ($1F_{tap}$) into SpectralCanvas or time-based widgets. The sound is always aligned to the 0Hz line. A time range for saving can be selected as a line in the SpectralCanvas or in the waveform widget. To export the frequency domain, an area above a certain threshold is drawn instead of a line. (see Figure 3)

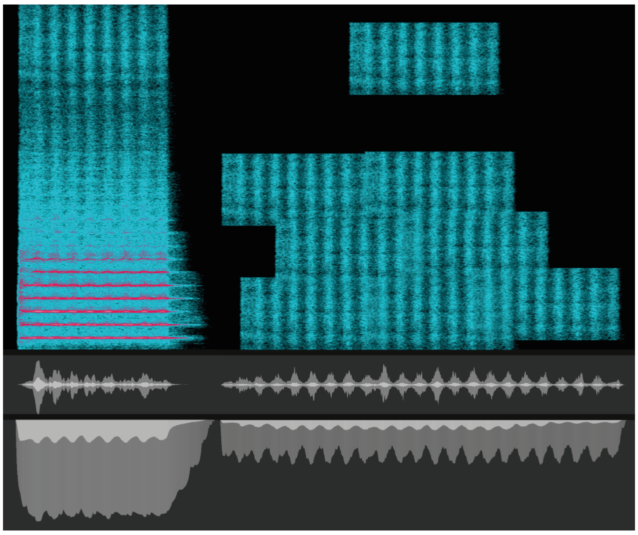

Figure 4: *StampModule: Selecting spectrals (left) and stamping it as new sound-nodes to the SpectralCanvas (right).*

- **transform** – Translation via dragging ($1F_{move}$), rotation & scaling via right-click drag ($1F_{hold} + 1F_{circle}$ & $1F_{hold} + 1F_{split}$), warping via manipulation handles.
- **stamp** – With the stamp-tool an area of spectrals is copied and stamped to a new location within the SpectralCanvas (see Figure 4). Soundtextures can be created.
- **draw** – Lines can be drawn representing sinusoids or transients. Drawing multiple lines via multitouch ($[1F_{move}]^*$) is possible. Therefore, navigational tasks with $3F_{move/sparse}$ are ignored. This module has high potential for drawing fundamental tones with instantaneously generated overtones to create lines with *natural* and *artificial* timbre.
- **effect** – Effect-nodes are treated like sound-nodes. They are created via normal selection.
- **loop** – VAD offers the possibility to create loops. The loops can cover the entire frequency and time range or, like the effects, individual frequency ranges. Individual loops can be started and stopped. In this way, a myriad of spectral synchronous and asynchronous loops can be played.
- **navigational tasks** – Panning via drag on workspace ($1F_{move}$ on workspace, or $3F_{move}$ anywhere) and zooming via scroll ($2F_{split/pinch}$ on workspace, or $3F_{sparse}$ anywhere).

3.1.2. Widgets

The widgets surround the SpectralCanvas (see Figure 5). This allows widgets to have one of the four orientations *(TOP, RIGHT, BOTTOM, LEFT)*. LEFT & RIGHT are frequency-based widgets and TOP & BOTTOM are time-based widgets. The same type of widgets can be arranged in arbitrary ways. In the following table each row can be modelled as a adaptive widget (see Tabel 1). For instance, if the waveform widget is dragged to a LEFT or RIGHT position it transforms into a spectrum. Advanced widgets for displaying and manipulating perceptual parameters such as sharpness, compactness, etc. are planed as future work.

time	frequency
timeline, time axis labeling	frequency axis labeling
waveform	spectrum
spectral power	long-term spectrum
current parameters	parameter distribution

Table 1: Relation of time based and frequency based widgets.

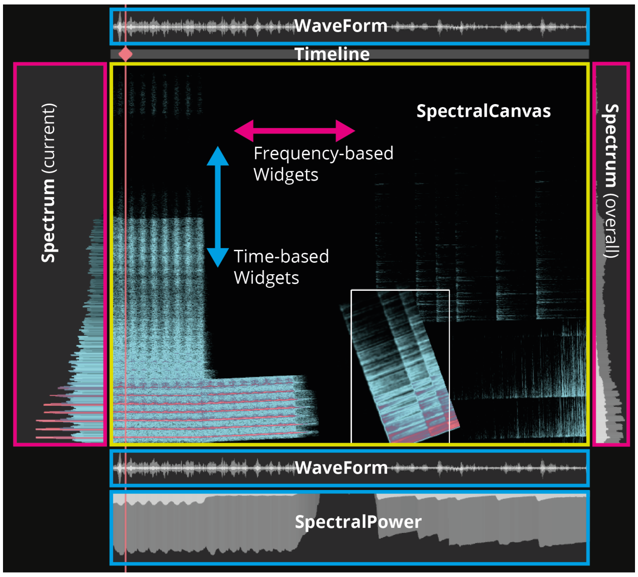

Figure 5: *Widgets are aligned horizontally or vertically to the SpectralCanvas.*

Widgets can bind themselves to events for getting the current play position, current spectrum, the whole frequency-domain, the whole time-domain, and the nodes of the SpectralCanvas. Each event will occur, if the corresponding data has been changed. At the same time, the data in the widgets can also be edited. An example is an equalizer in the spectrum.

The widget for spectral power and long-term spectrum are projections of the frequency-domain on the corresponding axis (see figure 5) and visualizes the minimum, maximum and mean values (min, max, avg).

3.1.3. Spectral Effects

A effect-node receives the underlaying spectral data as an event. Now, convolution kernels are applied. The easiest way is to just apply a kernel for blurring and sharpening, but conditional filter like a gain-threshold (Equation 1) can also be applied (Figure 6).

5x5 filter kernel for gaussian blur:

$$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

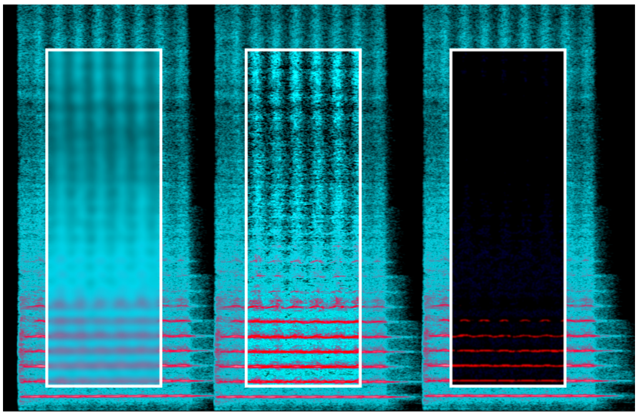

Figure 6: *Examples of effect-nodes: blurring (left), sharpening (unsharp masking) (middle) and gain-threshold (right).*

5x5 filter kernel for unsharp masking:

$$\frac{-1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & -476 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

$$f_{threshold}(spectral) = \begin{cases} spectral, & \text{if } spectral \geq threshold \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The unsharp masking is thereby derived from the gaussian blur and amplifies the high-frequency components of the neighbouring magnitudes in the two-dimensional space. The gaussian blur creates a more noisy and distant sound. On the other hand, the unsharp masking creates a rough and more direct sound. These kernels are structure-preserving, because harmonics and partials are not displaced. A short table shows the equivalent of time-domain effects for image processing frequency-domain effects (see Table 2). With this transfer, most conventional time-domain effects can be used directly in the VisualAudio-Design in a comprehensible way. Of particular interest are effects from the time-domain and image processing for which no transfer can be found. These are unique features.

audio	image
gate&limiter	tonal correction, black&white reassign
overdrive	tonal correction, white shift
compressor&expander	exposure lights&shadows
simple reverb	directional blur

Table 2: Relation of ordinary audio and image effects.

Additionally, some novel spectral effects are non-structure-preserving effects, like inflate and contract (see Figure 7). Those non-structure-preserving effects deform harmonics and transients, meaning that timbre and envelope are altered.

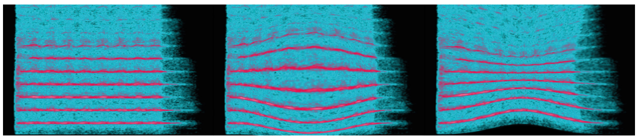

Figure 7: *Examples of non-structure-preserving/structure-invasive effects: none (left), inflating (middle) and contracting (right).*

3.2. Processing

The SpectralEngine has four worker threads for forward transformation, backward transformation, windowed time-data cumulation, and phase-estimation. The SpectralCanvas (scenegraph) has an instance of the SpectralEngine to process the overall results. In addition, every layer and every node have their own SpectralEngine. That offers the opportunity to instantly prehear only the node or layer soloistically. So there are at least $4 * (1 + \sum node + \sum layer)$ threads running, but most of them are idled.

To prehear a layer or the entire project, all spectral data of the nodes are merged. At first, nodes are merged to their layer and then the layers are merged at the same way to one consistent data layer. Merging in fact of the spectral data means, that the maximum value of all magnitudes and the average value of all phases are taken to reduce multiple spectral datasets to one.

To speed up processing, only the changed area of the corresponding and higher layers are recalculated. By transforming a node the changed area has to be processed. The whole changed area consists of the part where the node was before and the part where the node was moved to (see Figure 8). Because of overlapping windows, the influenced time range is *windowSize* many samples larger (Equation 4) than just the time position of the changed spectrums. Especially if an effect-node on top of a changed node is not fully covered in the changed Area ($A_{changed}$), then the half kernel size (see Equation 3) has to be added as well. In fact of multiple layered effect-nodes, for each effect Equation 3 has to be repeated. Only this range will be cumulated to the actual part of the audio buffer. For a smooth prehearing, the playback is double-buffered. While playing buffer A, new data is inserted to buffer B and after that the buffers are swapped.

$$A_{changed} = \sum spectrum_{changed} \tag{2}$$

if effect-node is not totally included in $A_{changed}$, repeat:

$$A_{changed} \mathrel{+}= \begin{cases} \|kernel\|, & \text{if effect is overlapping both sides} \\ \frac{\|kernel\|}{2}, & \text{if effect is overlapping one side} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where: $\|kernel\|$ = size of kernel

$$T_{influenced} = windowSize + hopSize * A_{changed} \tag{4}$$

3.2.1. Phase-estimation

The SpectralEngine decides depending on the strength of the manipulation, whether the phase values can be used further or whether they have to be re-estimated. The phase estimation is iteratively approximated with Griffen&Lim's Algorithm (GLA) [26] on its own thread in the background. After 5-10 iterations the root-mean-square error (RMSE) is already $\lessapprox 0.1$. Therefore,t he data is propagated to the rest of the system for the first time, so that the widgets are updated and a first prehearing is possible. In the background, the phases are approximated more and more, so that after 100 iterations (RMSE of some signals can already be $\lessapprox 0.05$) a further propangation takes place. Depending on the desired quality, further iterations are performed.

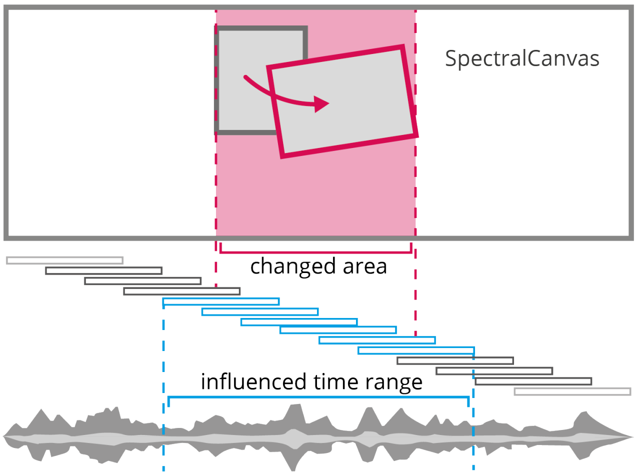

Figure 8: *Illustration of the changed area consisting of the old position (grey) and the new position (red) of a node, which is influencing a certain time range (blue).*

3.3. Challenges for Visualizations

Objective physical measurements are the basis for analysis, however, psychoacoustics deals with the subjective perception of signals (cf. [27]). Thereby, a *non-linearity* and a *non-orthogonality* occur. Sensory quantities often do not behave linearly to their corresponding acoustic quantities. Moreover, many sensory variables influence each other. For example, both frequency and volume have an influence on the perceived pitch.

Also, visual perception is non-linear and non-orthogonal. For the perception of the areas and volumes, the actual area or volume is underestimated [28, 29], although, for instance, the perception of lengths is directly proportional to the actual length of a line.

Spectrogram visualizations have a high dynamic range. Due to the visualization of a lot of sound nodes, there is a risk that the essential properties of a spectrogram are difficult to capture. Visual recognition is also complicated by noise. In the context of visual or image-based audio processing, however, a fast and simple inspection of sounds spectral space is necessary. One possibility for a improved visual inspection of a spectrogram is an abstraction of the frequency-domain. At the same time, the basic visual properties of the spectrogram, such as its dimensions, transients and harmonics, should be preserved.

When visualizing information, visual differences should be as subtle as possible, while being effective and meaningful [30]. Contrasts are defined as minimal and distinct. The number of distinctions can be increased by minimizing their differences. Using lower contrasts and differences, a potential visual disorder is reduced.

One approach is a sparse vectorized visualization of the overtones as in [31]. Another is the method presented here to extract layers of same decibels and visualize them in analogy to maps (see Figure 9). To achieve that, the magnitudes are quantized using

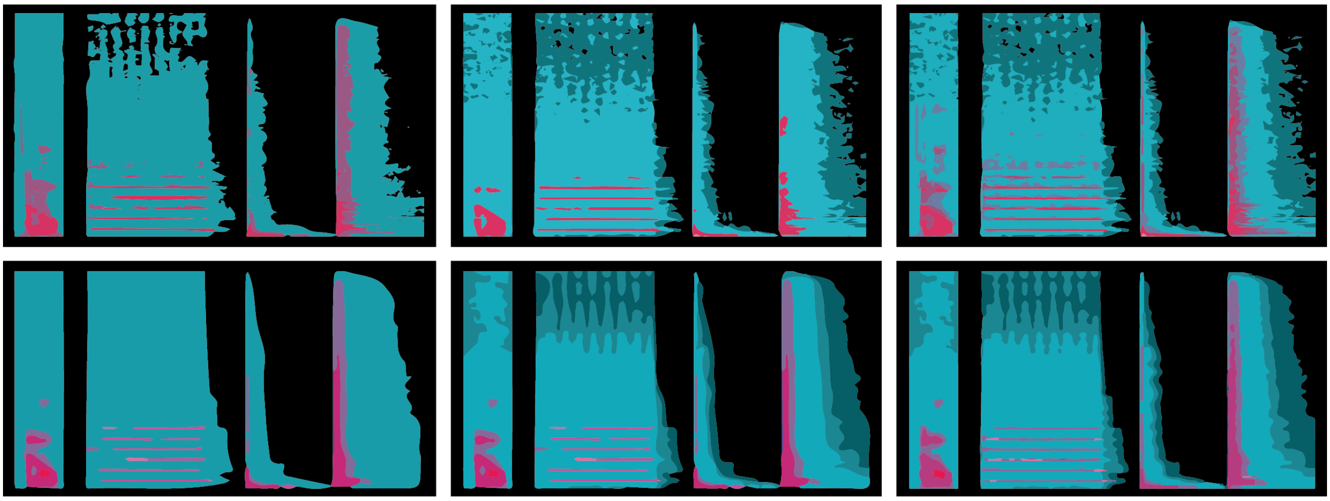

Figure 9: *Abstraction of the frequency-domain with different decibel thresholds: 3 threshold layer (left) and up to 6 (right), with no blurring (upper) and with blurring before thresholding (lower).*

$kmeans$. Here, the x largest clusters of similar magnitudes are determined. However, this means that each sound-node will have layers on different dB levels. Whereby these planes characterize the respective node particularly well. For an improved comparability of sound-nodes a $threshold$ is performed at specific dB level. For this, user-controlled exact levels can be generated. However, important information can be lost during visualization. Therefore, a combination is proposed. The $kmeans$ show the main clusters of a node and via $threshold$ uniform user controlled planes are shown over all nodes. For a smoother result where the plane edges are less noisy a prior blurring is done.

No more than about 20 layers with different color gradations should be used, as otherwise the distinguishability between the color gradations is made more difficult [32]. For the differentiation of two areas the indication of a line is sufficient, if needed at all. The line should have a hue which is in the color scale [32]. The distinguishable brightness values of colors depend on the corresponding hue.

3.4. User-centered Design

VAD highly focuses on UserExperience and ease-of-use. That is why it is designed and developed in a user-centered way. No formal study is carried out, yet. However, qualitative evaluations (expert interviews) with professionals and students from the fields of human-computerinteraction, sound design and music composition are conducted continuously in short design loop intervals.

3.4.1. User study

A brief user study was conducted. 26 subjects (20 male, 6 female) with average age of 26.12 participated. The subjects tested the VisualAudio-Design (VAD) with mouse only and had to perform transpositions of sound-nodes, rotations (arbitrary angles, but always including 90°), and applying effect-nodes (blur, sharpen, threshold) (compare Section 3.1.3). After performing these explicit tasks, they had time to create sound collages on their own.

Then they were asked with a 5-Likert scale whether VAD is creative, the resulting resynthesis is predictable, the interface and visualization is clear (or confusing) and whether it is easy to use. In addition, the user experience was tested with the *short user experience questionaire* (UEQ-S) [33] (with a 7-Likert scale).

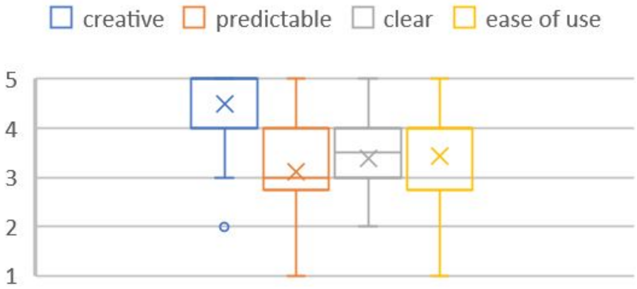

Figure 10: *Results of a brief user study with a 5-Likert scale (the higher the better): uncreative (1) – creative (5), unpredictable – predictable, confusing – clear, complicated – easy to use*

The results are indicating that VAD is highly creative (see Figure 10), while the predictability is moderate. The clarity and the ease of use of VAD is given. The UEQ-S is reflecting these indications. The user experience (hedonic quality) is more than average and the usability (pragmatic quality) is more than moderate (see Figure 11). This is a good starting point, but the interface to effectivly use the node-based VAD has to be improved in future work.

4. FUTURE AND ON-GOING WORK

Users of the qualitative evaluations wanted more intiutive touch gestures than the existing ones in order to operate VAD without GUI as mode switch. To achieve this, a complete GeForMT interpreter has to be implemented. Also gaze and pen should be included in the interaction.

At the moment it is only possible to select magnitudes via a rectangle selection. For this, freehand and polygon selections should be added, as stated in the related works, which follow content based features like harmonic overtones.

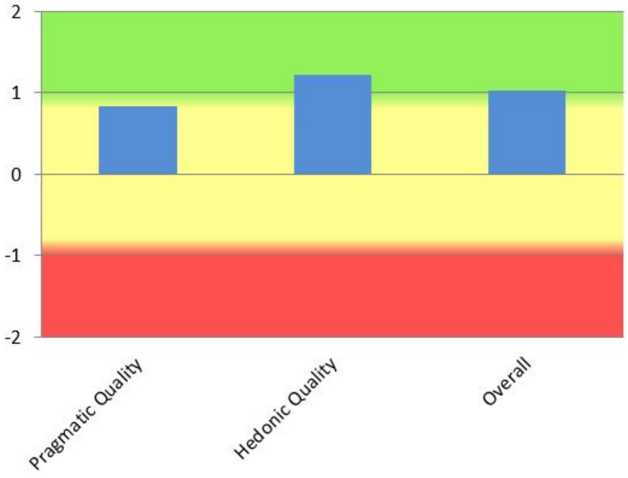

Figure 11: *Results of the UEQ-S test.*

The next stage of development will be formal studies. In particular, the coherence and influence of the visual on the auditory perception will be investigated for achieving a non-invasive *sound*-design.

The draw module has to be completely redesigned in order to paint meaningful timbre directly. In this sense also machine learning supported morphing between timbres should follow.

In addtion, great potential also has the merging of the nodes timbre. In this way, combined sounds are created by aligning the maximum harmonic components and/or the maximum spectral power (transient component) (see Figure 12).

5. CONCLUSIONS

In this paper the VisualAudio-Design was introduced, which is intended for the creative use for sound design and composition. The spectral manipulation is based on nodes within a scenegraph (SpectralCanvas) and its SpectralEngine, a highly multi-threaded processing kernel. In this context, basic tools for manipulating nodes and specialised effect-nodes for convolutions were presented. Additionally, widgets are used for analysis, advanced selection and interaction. As well as an abstraction of the frequency-domain was discussed.

6. ACKNOWLEDGMENTS

Thanks to all students who took part in practical and theoretical courses around VAD, testing it, or served as subjects, and thus contributed to the overall development.

7. REFERENCES

[1] Joshua Fineberg, “Spectral music,” *Contemporary Music Review*, vol. 19, no. 2, pp. 1–5, 2000.

[2] G Eckel, “Manipulation of Sound Signals Based on Graphical Representation-A Musical Point of View,” in *Proceedings of the International Workshop on Models and Representations of Musical Signals, Capri, Italia*, 1992.

[3] Michael Klingbeil, “Software for spectral Analysis, Editing, and synthesis.,” in *International Computer Music Conference (ICMC)*, Barcelona, Spain, 2005.

[4] John M Grey and John W Gordon, “Perceptual effects of spectral modifications on musical timbres,” vol. 63, no. 5, pp. 1493–1500.

[5] Niels Bogaards, Axel Roebel, and Xavier Rodet, “Sound analysis and processing with audiosculpt 2,” *International Computer Music Conference (ICMC)*, p. 1, Nov 2004.

[6] Allan Seago, “A new interaction strategy for musical timbre design,” in *Music and human-computer interaction*, pp. 153–169. Springer, 2013.

[7] Shams Watkins, Ladan Shams, Sachiyo Tanaka, J-D Haynes, and Geraint Rees, “Sound alters activity in human v1 in association with illusory visual perception,” *Neuroimage*, vol. 31, no. 3, pp. 1247–1256, 2006.

[8] Gemma A Calvert, Peter C Hansen, Susan D Iversen, and Michael J Brammer, “Detection of audio-visual integration sites in humans by application of electrophysiological criteria to the bold effect,” *Neuroimage*, vol. 14, no. 2, pp. 427–438, 2001.

[9] Charles E Schroeder and John J Foxe, “The timing and laminar profile of converging inputs to multisensory areas of the macaque neocortex,” *Cognitive Brain Research*, vol. 14, no. 1, pp. 187–198, 2002.

[10] Francesca Frassinetti, Nadia Bolognini, and Elisabetta Làdavas, “Enhancement of visual perception by crossmodal visuo-auditory interaction,” *Experimental brain research*, vol. 147, no. 3, pp. 332–343, 2002.

[11] Charles E Schroeder and John Foxe, “Multisensory contributions to low-level, unisensory processing,” *Current opinion in neurobiology*, vol. 15, no. 4, pp. 454–458, 2005.

[12] Marie-Helene Giard and Franck Peronnet, “Auditory-visual integration during multimodal object recognition in humans: a behavioral and electrophysiological study,” *Journal of cognitive neuroscience*, vol. 11, no. 5, pp. 473–490, 1999.

[13] Kostas Giannakis and Matt Smith, “Imaging soundscapes: Identifying cognitive associations between auditory and visual dimensions,” *Musical Imagery*, pp. 161–179, 2001.

[14] Jean-Baptiste Thiebaut, Patrick GT Healey, and Nick Bryan-Kinns, “Drawing electroacoustic music.,” in *ICMC*, 2008.

[15] Niels Bogaards and Axel Röbel, “An Interface for Analysis-Driven Sound Processing,” in *Audio Engineering Society Convention 119*, 2005.

[16] Ananya Misra, Perry R. Cook, and Ge Wang, “Tapestrea: Sound scene modeling by example,” in *ACM SIGGRAPH 2006 Sketches*, New York, NY, USA, 2006, SIGGRAPH ’06, ACM.

[17] T Quatieri and Rl McAulay, “Speech transformations based on a sinusoidal representation,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 6, pp. 1449–1464, 1986.

[18] E Wenger and E Spiegel, “Metasynth 4.0 user guide and reference,” *Redwood City, CA: U&I Software LLC. www. uisoftware. com/MetaSynth*, 2005.

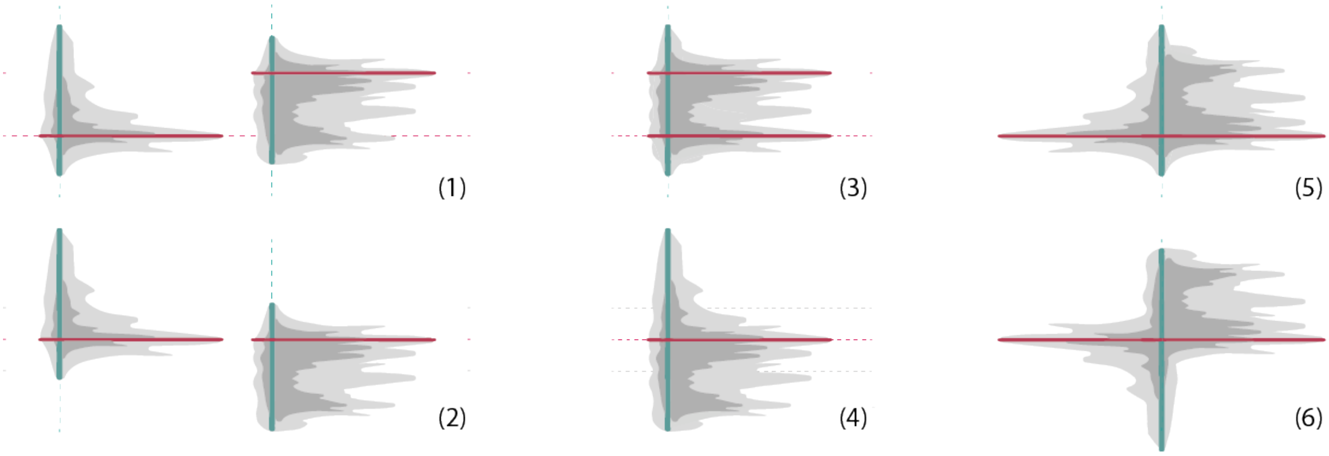

Figure 12: *Alignment of sound-nodes: (1) original position (no alignment), (2) harmonic alignment, (3) temporal alignment, (4) harmonic and temporal, and more complex harmonic & temporal alignments with mirroring of the first node: (5) vertical, (6) vertical and horizontal.*

[19] Lars Engeln and Rainer Groh, "AudioFlux : A Proposal for interactive Visual Audio Manipulation," in *Mensch und Computer 2017 - Workshopband*, M. Burghardt, R. Wimmer, C. Wolff, and C. Womser-Hacker, Eds., Regensburg, Germany, 2017, number September, Gesellschaft für Informatik e.V.

[20] Lars Engeln, Natalie Hube, and Rainer Groh, "Immersive visualaudiodesign: Spectral editing in vr," in *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*. ACM, 2018, p. 38.

[21] Konstantinos Giannakis, "Sound mosaics a graphical user interface for sound synthesis based on auditory-visual associations," 2001.

[22] Kostas Giannakis, "A comparative evaluation of auditory-visual mappings for sound visualisation," *Organised Sound*, vol. 11, no. 3, pp. 297–307, 2006.

[23] Diemo Schwarz, "State of the Art in Sound Texture Synthesis," in *Digital Audio Effects (DAFx)*, Paris, France, Sept. 2011, pp. 221–232.

[24] Chris Cannam, Christian Landone, and Mark Sandler, "Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files," in *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 2010, pp. 1467–1468.

[25] Dietrich Kammer, Jan Wojdziak, Mandy Keck, Rainer Groh, and Severin Taranko, "Towards a formalization of multi-touch gestures," in *ACM International Conference on Interactive Tabletops and Surfaces*. ACM, 2010, pp. 49–58.

[26] Nathanaël Perraudin, Peter Balazs, and Peter L Søndergaard, "A fast griffin-lim algorithm," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2013, pp. 1–4.

[27] Gareth Loy, *Musimathics: the mathematical foundations of music*, vol. 1, MIT press, 2011.

[28] Stanley S Stevens and Eugene H Galanter, "Ratio scales and category scales for a dozen perceptual continua.," *Journal of experimental psychology*, vol. 54, no. 6, pp. 377, 1957.

[29] James John Flannery, "The relative effectiveness of some common graduated point symbols in the presentation of quantitative data," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 8, no. 2, pp. 96–109, 1971.

[30] Edward R Tufte, Susan R McKay, Wolfgang Christian, and James R Matey, "Visual explanations: images and quantities, evidence and narrative," 1998.

[31] Michael Klingbeil, "SPEAR: Sinusoidal Partial Editing Analysis and Resynthesis," 2012, vol. 12, p. 3.

[32] Edward R Tufte, Nora Hillman Goeler, and Richard Benson, *Envisioning information*, vol. 126, Graphics press Cheshire, CT, 1990.

[33] Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski, "Design and evaluation of a short version of the user experience questionnaire (ueq-s).," .