SONNOTILE: AUDIO ANNOTATION AND SONIFICATION FOR LARGE TILED AUDIO/VISUAL DISPLAY ENVIRONMENTS

Zachary Seldess

Visualization Lab
King Abdullah University of Science and Technology
Thuwal, Saudi Arabia
zachary.seldess@kaust.edu.sa

So Yamaoka, Falko Kuester

Calit2 Center of Graphics Visualization and Virtual Reality (GRAVITY)
University of California, San Diego
La Jolla, CA, USA
syamaoka@ucsd.edu, fkuester@ucsd.edu

ABSTRACT

We present "Sonnotile", a multi-modal rendering framework to enhance scientific data exploration, representation, and analysis within tiled-display visualization environments. Sonnotile aims to assist researchers in the customization and embedding of sound objects within their data sets. These sound objects may act as way-finding markers within a media space, as well as allow researchers to attach and recall various sonic descriptions or representations of an arbitrary number of regions within a data set. In designing the software, our initial efforts have been centered on the challenges of sound "annotation" within large-scale pyramidal TIFF files.

1. INTRODUCTION

Large tiled-display walls, with tens of megapixels of display area, provide a unique environment for data exploration and analysis where extremely large and high-resolution datasets can be studied. However, although a large display can effectively present information detail while preserving its context better than a small desktop setup, the perception of information and tasks such as way-finding or search can become difficult precisely because of the large size of the environment, as well as the scale of the datasets which can be visualized. For example, a user generally knowing where a region of interest resides within an ultra-high resolution dataset does not at all guarantee that she will be able to find it quickly and easily. The sometimes overwhelming flow of visual information created by such environments can often distract users from the task at hand. Furthermore, multivariate datasets only compound the problem by creating cluttered visual representations.

The large physical and virtual media spaces typical in tiled-display walls provide an ideal environment for enhancements through the use of audio. Visualization research is concerned with the effective representation of data to enhance the user's insight and understanding of the information. In light of the above potential challenges to the visual display, we have the opportunity to leverage the strengths of the ears in improving way-finding abilities through audio cues, as discussed in [1], and in widening the perceptual bandwidth through simultaneous multi-modal data realizations, as discussed in [2], [3], [4], [5], [6], [7], [8], among many others. In this paper we present "Sonnotile", a framework to assist researchers in embedding simple sonic annotations and way-finding markers within their data sets.

2. TIFFVIEWER

Recently, applications enabling interactions with a large number of high-resolution images have been developed for large tiled-display walls [9]. TiffViewer [10], one such application, provides a large, unified workspace that spans across an entire display space provided by a high-resolution tiled display system. In TiffViewer, TIFF-encoded images are used for an out-of-core visualization technique. Basically, an image is visualized as a collection of small, TIFF-tile textures, tightly packed as a grid. Any TIFF-tiles that fall outside the current viewing volume are invalidated and recycled. In this way, many multi-gigapixel images can co-exist in the provided workspace, requiring only a fixed memory foot print.

The above application has been augmented to act as a visualization component to Sonnotile. During an interactive session, the state of the mouse pointer and modified images are encoded as OSC messages, and sent to the audio server. The encoded information includes: a) the name, dimension, and position of images; b) the id and state of the mouse button, and the position of the mouse pointer.

3. SONNOTILE

Sonnotile allows users to define and attach sound objects directly to an arbitrary number of images and to an arbitrary number of regions embedded within those images (Figure 1). These objects are created and manipulated in a hierarchical parent-child structure, allowing for a variety of complex logical operations to be initiated at various levels within an image's object structure. However, the amount of objects that can be tracked and sonified within a display environment is limited by the processing capabilities of the machine(s) being used. Therefore, in an attempt to grapple with issues of massive scale, as is common in tiled display environments, we have developed customizable processing load management methods within the software. Audio rendering for off-screen, inaudible, or non-essential objects can be dynamically muted, allowing for the "marking up" of many more objects within a space than could possibly be simultaneously rendered.

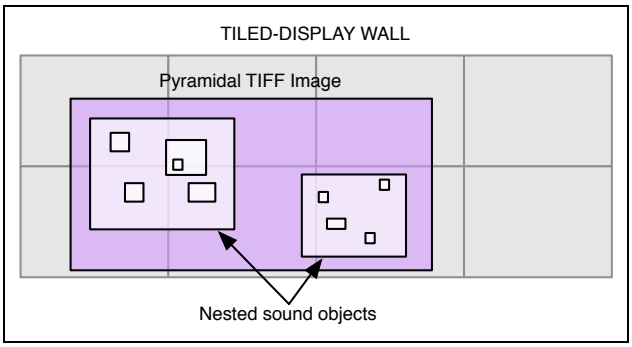

Figure 1: Nested sound objects within a pyramidal TIFF

Sound objects associated with an image, or attached to regions embedded within an image can contain two audio components: a "sound marker", and a "sound annotation":

SOUND MARKERS are looping sampled sounds that are attached to sound objects. These constructs are intended primarily as a way-finding tool within the media space, helping to alert and assist users in finding areas of interest within the display environment, as well as highlighting high-level similarities and hierarchies between annotated areas of interest within the data.

SOUND ANNOTATIONS are non-looping sampled sounds attached to sound objects that are actively triggered on and off by users (via mouse event, etc.). Sound annotations are intended to provide a useful method for storing and recalling specific details on selected regions within the data set, whether they be pre-recorded vocal narratives describing the data, more abstract symbolic sonification of other dimensions of the data not visually rendered, or various other methods of describing the data.

3.1. Configuration Conventions

Sonnotile uses a configuration file structure that stores sound object definitions in a hierarchical parent-child fashion. That is, it supports nested user definitions of sub-regions within sub-regions within sub-regions, etc. The plaintext configuration file contains five major areas to configure the system environment, sound object definitions, and their various behaviors (Figure 2).

Area 1 of the configuration file contains information on general software initialization, such as speaker count and physical locations in reference to the display environment, and other installation-specific issues such as the display resolution in pixels, whether or not the visual space wraps around on itself (as is often the case in immersive virtual reality contexts), or has boundaries (as in a flat tiled display wall), and all other parameters not related to the marking up of images within the space.

All remaining areas in the configuration file pertain to sound object definition and behavior. Features here are abstracted in a way that allow for easy reuse of code blocks amongst many different sound objects within a project.

Area 2 of the configuration file defines the names and hierarchical structures of sound objects within images, as well as references each object to an "audio profile" (described in Area 3).

Area 3 contains audio profile definitions. An audio profile includes sound file associations for an object's sound marker and sound annotation, as well as a reference to an "audio description" and a "usage description" (described in Area 4 and 5).

In **Areas 4 and 5** "audio descriptions" and "usage descriptions" are defined. Audio and usage descriptions contain details on a sound object's specific behaviors, such as loudness and fading characteristics, as well as human interface parameters such as the required mouse button to trigger playback of an annotation.

```
Setup { // GENERAL SOFTWARE SETUP

    displaySize = 13600 3072 // tiled-display wall width and height in pixels
    setSpeakers = 0. 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1. // speaker coordinates
    isWrapping = 0 // space is modulo or not
    decorrelation = 0 // 1 == real-time decorrelation , 0 == no decorrelation
    ...
}

SoundObjects { // NESTED SOUND OBJECT DEFINITIONS

    echogram {
        name = echogram.tif // file name of image
        audioProfiles = echogramAP// audioProfile associated with object

        // children regions embedded within image
        freeWaterMass {
            location = 0.01 0.12 0.17 0.758 // w h x y (in percentage of parent object)
            audioProfiles = freeWaterMassAP
        }
    }
    ...
}

AudioProfiles { // AUDIOPROFILES DEFINITIONS
    echogramAP {
        marker { // if no looping sound marker needed, leave this empty
        }
        annotation {
            soundFile = Thor_RedSeaSonogram.wav
            audioDescription = imageAD1
            usageDescription = imageUD1
        }
    }
    ...
```

Figure 2: Portion of a Sonnotile configuration file

3.2. Mapping Between Different Physical Environments

The location and size of each sound object embedded within an image is defined in reference to the Cartesian coordinate space in normalized x,y coordinates, with the origin located at the bottom-left corner of the image. We use normalized coordinates rather than pixel coordinates to allow for a more intuitive annotation terminology, and to avoid the burden of dealing with incredibly large coordinate systems that inevitably change from image to image.

In 3-dimensional media space, perceptual fading of an in-world sound object is often coupled to its distance from the camera, or virtual head. In the 2-dimensional media space of TiffViewer, we have adopted an analogous mapping that equates visual size to loudness (visual size being a byproduct of distance in 3-D space). Whereas in 3-D space, one might define a reference distance at which an object renders at full volume, coupled with a free-field or custom roll-off curve defining the fading behavior of the object, we have chosen to define fading

behavior in terms of the object's normalized area relative to the overall area of the display wall.

Fading behaviors of sound objects attached directly to images are defined by establishing a reference area that equates to “full” volume. When an object’s area is equal to this value, its audio signal will be attenuated by -0dB. A roll-off curve for attenuating the sound object as it gets smaller is then defined in reference to this area by providing a dB reduction per halved area. For example, given an object with a reference area of 1. unit and a roll-off value of -6dB, at 0.5 units the object would be attenuated by 6dB, at 0.25 units 12dB, at 0.125 units 18dB, and so on.

Fading behaviors of sound objects nested *within* regions of images are defined in the same manner, but with the addition of parameters establishing a secondary fade that occurs as objects grow beyond the above-mentioned reference size. Similar to the primary fade parameters, this secondary fade defines how the sound object will recede as other more deeply embedded sound objects grow in area and become sonically and visually foregrounded.

Using normalized areas to define fading behaviors provides what we have found to be the most intuitive method by which to deal with these important sonic parameters. However, tiled display walls come in a variety of sizes, and therefore this solution provides its own set of challenges when moving between environments, as the practical realization of these normalized areas can change drastically when defined in reference to different pixel resolutions and aspect ratios. To counterbalance the awkwardness of mapping annotations between different sized spaces, we have developed three "fade modes", which are associated with sound objects in the configuration file (Figure 3).

ABSOLUTE MODE: In absolute fade mode, all normalized areas are defined in relation to a reference display size. A reference display size is typically the full pixel resolution of the display wall on which the annotation was first created. When moving the project to a different sized wall, all normalized areas controlling fading behavior are then translated into actual pixel areas as experienced on the original display. This mode is useful, for example, when you want to audition the results of an annotation on your single desktop display or laptop screen, as if the display were one part of the entire wall.

RELATIVE MODE: In relative fade mode, all normalized areas are defined in relation to the overall display resolution of the current wall being used. That is, if a sound object is defined to be -0dB when its normalized area is 0.25 on a display wall with an 8000 x 4000 pixel resolution, that object will equal -0dB when its area is 0.25 in any display context, regardless of the size of the environment.

RELATIVE-BIAS: In relative-bias fade mode, as in relative mode, all normalized areas are defined in relation to the overall display resolution of the wall being used, but with additional adjustments applied to compensate for any divergence in aspect ratio between the display environment on which the annotation was made and the environment currently being used.

Figure 3 illustrates the effect of each fade mode on a sound object defined on one display wall, and then rendered on a smaller display wall.

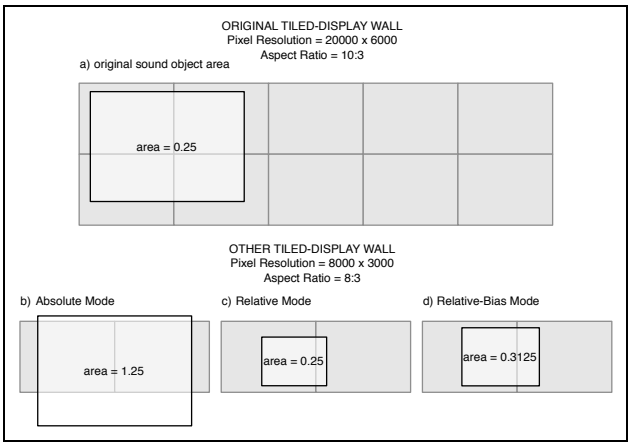

Figure 3: In a), a sound object is initially defined on a large tiled-display wall as having an attenuation of -0dB when its normalized area equals 0.25. b), c), and d) show the areas for that object that cause -0dB of attenuation on a much smaller display wall, when applying the absolute, relative, and relative-bias fade modes, respectively.

4. PANNING METHOD

One of the unique challenges in spatializing sound objects within a tiled-display environment is delivering a convincing representation of each object's physical size. Whereas it is often sufficient in virtual reality and gaming scenarios to represent spatial audio cues as point sources using a variety of well-established and effective panning algorithms (e.g. [11], [12], [13]), this approach can at times prove unconvincing when dealing with sound-emitting objects that span portions of large display walls, such as the 25-foot wall used at KAUST (Figure 4). Therefore, for this project we have developed a computationally efficient, variable-channel equal-power panning algorithm that renders width in addition to basic point-source location. As our display walls feature horizontal arrays of loudspeakers mounted above the displays, and due to the increased spatial sensitivity of the ears on the horizontal plane, our panner is specialized for one-dimensional configurations (including surround scenarios, and non-uniform speaker layouts). The panner therefore is designed specifically to render width and azimuthal location, but not elevation.

Figure 4: 40 megapixel, 25-foot wide tiled-display wall, with 11 Meyer MM-4XPD miniature loudspeakers

In order for the apparent source width and panning behavior to work well in combination with the visual display environment, it is required that all speaker locations be defined in the configuration file in reference to the display wall. This is done in normalized horizontal coordinates with 0 representing the left edge of the display and 1 representing the right edge. The panner input then accepts the left and right bounds of an object (rather than its centroid) as expressed in the display's normalized horizontal coordinates, and determines which speakers fall within the object's physical width, which fall just beyond its left and right boundaries, and which, if any, fall further away still. Weighting values are then assigned to each speaker based on its location in relation to the object's boundaries as follows: All speakers falling within the object's boundaries are assigned a weight of 1. The inner-most two speakers that lie outside the object's left and right edges are assigned weights between 0 and 1 depending on their distance from the boundary. For example, if x_L represents the location of the object's left boundary, l_o represents the location of the speaker lying directly to left of x_L, and l_1 represents the location of the speaker lying directly to the right of x_L, l_o is assigned the weight of $1-\left(x_L - l_o / l_1 - l_o\right)$. All speakers whose coordinates are further away than these most adjacent "outside" speakers are assigned a weight of 0. Equation (1) summarizes these four cases, where w_i is the weight, l_i is the location of ith speaker, x_L and x_R are the left and right bounds of the object, and N is the total number of speakers.

$$w_i = \begin{cases} 1, & x_L \le l_i \le x_R \\ 1 - \dfrac{x_L - l_i}{l_{i+1} - l_i}, & l_i < x_L < l_{i+1} \\ 1 - \dfrac{l_i - x_R}{l_i - l_{i-1}}, & l_{i-1} < x_R < l_i \\ 0, & otherwise \end{cases}, \quad i = 0, ..., N-1 \tag{1}$$

These weights are then normalized and translated into amplitude a_i by calculating the square root of each item in the array, as given in (2).

$$a_i = \sqrt{\frac{w_i}{\sum_{n=0}^{N-1} w_n}}, \qquad i = 0, ..., N-1 \tag{2}$$

The resulting amplitude values represent the scalars that are applied to the annotation sounds for each speaker in the configuration. A further scalar g is then globally applied to all speakers depending on the pre-defined fading behavior of the object and it's current normalized area. Equation (3) shows the final speaker signal $\mathbf{s}_i$ for a given sound signal $\mathbf{s}$.

$$\mathbf{s}_i = g \cdot a_i \cdot \mathbf{s}, \qquad i = 0, ..., N-1 \tag{3}$$

Figure 5 provides an example implementation of equations 1 and 2, as applied to a sound object located within a tiled display wall containing a six-speaker configuration.

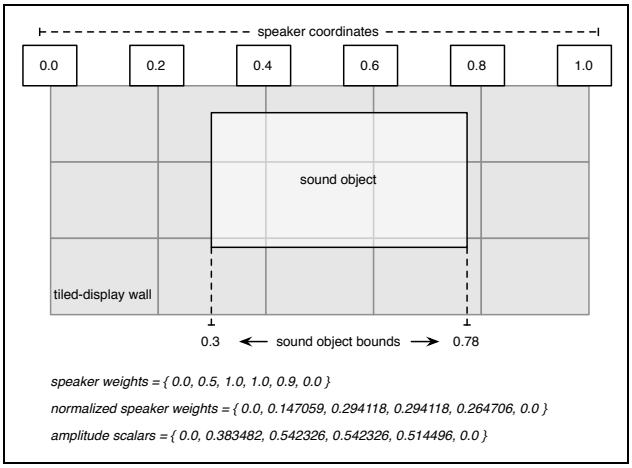

Figure 5: Non-normalized speaker weights, normalized speaker weights, and amplitude scalars for a sound object on a six-speaker display wall.

5. AUDIO DECORRELATION FOR APPARENT SOURCE WIDTH

After our initial experimentation with the panner, it became apparent that with certain speaker configurations, such as when using only two loudspeakers, or multiple widely separated loudspeakers, real-time decorrelation of the audio signals is required to produce a convincing sense of envelopment and apparent source width. After experimenting with several approaches, we have currently settled on a modified version of Bouéri and Kyirakakis' method for decorrelating audio signals by applying a random time shift to the twenty four frequency bands that correspond to the critical bands of the human ear, as described in [14].

Through subjective listening tests, we found that by reducing the amount of critical bands to which delays are applied from the full twenty four to only the top three bands (7700-9500 Hz, 9500-12000 Hz, 12000-15500 Hz), we could significantly decrease the computational load of the algorithm while experiencing minimal reduction in the effect of the decorrelation on the majority of the sounds used in our annotations. However, this method still remains far too computationally expensive in situations where hundreds of embedded sound objects need to be simultaneously rendered across multiple speakers in real-time. It is also worth noting that in our experience, in configurations where multiple speakers are placed equidistant and physically close to their neighbors, the perceptual improvements provided by audio decorrelation are not significant enough to justify the additional CPU load. We have therefore implemented decorrelation as an optional feature of the panner, which can be initialized in the software's configuration file, as well as dynamically enabled and disabled.

6. CONCLUSIONS

This paper has described a framework for sonic annotation and sonification of data in large-scale tiled-display visualization environments. We discussed the software's annotation design syntax, as well as a simple custom variable-channel 1-dimensional amplitude panner with optional real-time decorrelation for enhancement of apparent source width. Sonnotile is still in a very early stage of design, and as such in the future we plan to explore and implement a wide range of improvements and additional functionality.

Moving forward, we will continue to search for increasingly efficient real-time decorrelation algorithms for enhancing apparent source width. We also hope to augment the panning algorithm to deal with 2-D and 3-D speaker configurations, and to explore convincing rendering of off-screen audio cues in non-wrapping visual environments.

We plan to provide a system for designing custom "modules" that extend the software's core functionality. This will allow the software to support a wide variety of idiosyncratic future solutions, such as implementing a text-to-speech rendering engine, custom parameter mapping sonification approaches, real-time image annotation, and more.

Finally, it is our intention that future versions of the software will provide support beyond large-scale still image contexts, such as video playback and interactive animations, and 3-D virtual environments.

7. ACKNOWLEDGMENT

This ongoing project would not be possible without the support of Steven Cutchin, Thomas A. DeFanti, and all of our colleagues at California Institute for Telecommunications and Information Technology and the KAUST Visualization Lab. Thanks to Toshiro Yamada, Daniel Acevedo, and Jens Schneider for their technical and editorial assistance.

8. REFERENCES

[1] T. Lokki, M. Grohn. "Navigation with Auditory Cues in a Virtual Environment". *IEEE Multimedia*, vol. 12 (2), pp. 80-86, 2005.

[2] T. Bonebright, P. Cook, J. Flowers, N. Miner, J Neuhoff, R. Bargar, S. Barrass, J. Berger, G. Evreinov, W.T. Fitch. "Sonification Report: Status of the Field and Research Agenda," *prepared for the National Science Foundation by members of the International Community for Auditory Display*, 1997.

[3] A.O. Effenberg. "Movement Sonification: Effects on Perception and Action". *IEEE multimedia*, vol. 12 (2), pp. 53-59, 2005.

[4] M. Fernström, E. Brazil, and L. Bannon. "HCI Design and Interactive Sonification for Fingers and Ears". *IEEE MultiMedia*, vol. 12 (2), pp. 80-86, 2005.

[5] F. Grond, S. Janssen, S. Schirmer, T. Hermann. "Browsing Rna Structures by Interactive Sonification," *in Proceedings of the 3rd Interactive Sonification Workshop*, 2010, pp. 11-15.

[6] T. Hermann, T. Bovermann, E. Riedenklau, H. Ritter. "Tangible Computing for Interactive Sonification of Multivariate Data," *in Proceedings of the 2nd Interactive Sonification Workshop*, 2007.

[7] M. Rath, D. Rocchesso. "Continuous Sonic Feedback from a Rolling Ball". *IEEE Multimedia*, vol. 12 (2), pp. 60-69, 2005.

[8] R. Tünnermann, T. Hermann. "Multi-Touch Interactions for Model-Based Sonification," *in Proceedings of the 15th International Conference on Auditory Display*, 2009.

[9] K. Ponto, K. Doerr, F. Kuester. "Giga-stack: A method for visualizing giga-pixel layered imagery on massively tiled displays". *Future Generation Computer Systems*, vol. 26 (5), pp. 693-700, May 2010.

[10] S. Yamaoka, K. Ponto, K. Doerr, F. Kuester. "Interactive Image Fusion in Distributed Visualization Environments," *in Aerospace Conference 2011 IEEE*, 2011, pp. 1-7.

[11] T. Lossius, P. Baltazar, T. de la Hogue. "DBAP - Distance-Based Amplitude Panning," *in Proceedings of 2009 International Computer Music Conference*, 2009.

[12] V. Pulkki. "Virtual Sound Source Positioning Using Vector Base Amplitude Panning". *Journal of the Audio Engineering Society*, vol. 45 (6), pp. 456-466, 1997.

[13] J.C. Schacher, P. Kocher. "Ambisonics Spatialization Tools for Max/MSP," *in Proceedings of the 2006 International Computer Music Conference*, 2006.

[14] M. Bouéri and C. Kyirakakis. "Audio Signal Decorrelation Based on a Critical Band Approach," *in 117th Convention of the Audio Engineering Society*, 2004, pp. 28-31.