SPATIALIZED ANONYMOUS AUDIO FOR BROWSING SENSOR NETWORKS VIA VIRTUAL WORLDS

Nicholas Joliat, Brian Mayton and Joseph A. Paradiso

Responsive Environments Group
MIT Media Lab
Cambridge MA 02139
{njoliat, bmayton, joep}@media.mit.edu

ABSTRACT

We explore new ways to communicate sensor data by combining spatialized sonification with animated data visualization in a 3D virtual environment. A system is designed and implemented that implies a sense of anonymized presence in an instrumented building by manipulating navigable live and recorded spatial audio streams. Exploration of both real-time and archived data is enabled. In particular, algorithms for obfuscating audio to protect privacy and for time-compressing audio to allow exploration on diverse time scales are implemented. Synthesized sonification of diverse, distributed sensor data in this context is also supported within our framework.

1. INTRODUCTION

Homes and workspaces are increasingly being instrumented with dense sensor networks, encompassing many modalities of data, from environmental (e.g. temperature) to usage data (e.g. movement). While simple closed-loop systems exist for addressing specific problems (such as temperature control and lighting), rich, diverse sources of building data are generally balkanized within individual systems and cant be collectively explored. We are interested in creating a comprehensive interface that brings together all this complex information, allowing users to fluidly explore it in order to find patterns and connections between disparate kinds of data. Within a few years, sensor information will be agnostic to the specific systems that created it, mandating the ability to leverage such holistic sensor browsing systems in debugging and structuring ubiquitous computing [1] environments.

Graphical data visualization is a natural way to approach a problem of data display. Creating such displays entails challenges, chief among them being the reduction of rich multidimensional data to a two-dimensional display (a problem Edward Tufte refers to as 'flatland' in his seminal book *The Visual Display of Quantitative Information* [2].) In this work we explore the use of spatialized data sonification alongside animated 3D visualization for displaying rich spatial sensor data. Given that humans can localize sounds in three-dimensional space, spatialized sound is well suited to displaying three-dimensional spatial data, especially within a 3D virtual world. In addition, we can use sound to provide a more immersive and aesthetically compelling remote experience of a space. In particular, we use processed audio streams from the space, which both provide this immersive experience and effectively convey the usage of the space.

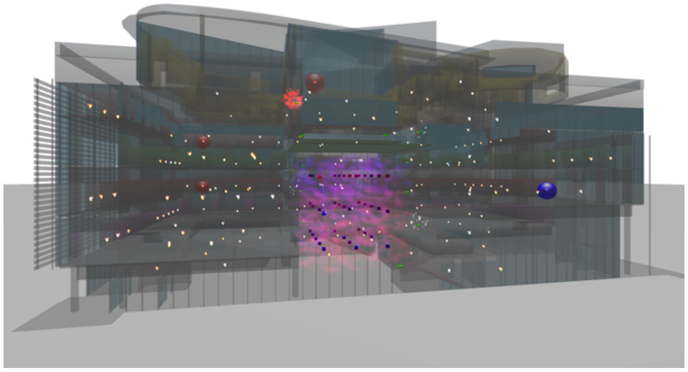


Figure 1: Full view of DoppelLab's representation of the Media Labs E14 Complex. Small flames visualize temperatures via their color; red and blue spheres visualize significant temperature anomalies; blue and purple fog and cubes visualize a dense array of temperature and humidity sensors in the building's atrium.

This work builds on DoppelLab, an ongoing project of the Responsive Environments group at the MIT Media Lab [3]. DoppelLab is a *cross-reality* virtual environment, metaphoring a building in the real world [4]. DoppelLab is populated with visualizations of diverse types of data, including temperature, humidity, motion, RFID tracking, geo-tagged tweets, and audio levels. Currently DoppelLab uses the new MIT Media Lab (E14) building as its test case; Figure 1 is a full-model screenshot of the interface. DoppelLab allows exploration of data both spatially, through a game-engine interface, and across time—real-time data is available, and archived data can be scanned over on a variety of time scales.

In this work, we build on DoppelLab to explore the potential of spatialized data sonification as a way to convey spatially-oriented sensor data. We explore two kinds of data sonification: synthesized sonification of non-audio sensor data (e.g. temperature, movement and presence of people) and the use of streamed and recorded audio as a more direct sonification. We particularly explore several issues that arise in the latter case: how to respect privacy while using recorded audio from a shared space, and how to allow users to efficiently explore recorded audio over long time

This work was sponsored by a grant from the Intel Corporation and the other sponsors of the MIT Media Laboratory.

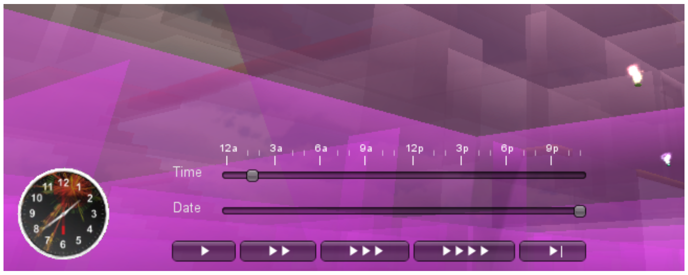


Figure 2: The GUI for DoppelLab's Time Machine functionality. Sliders allow coarse and fine selection of a new time to play back data and sound from; triangle buttons allow selection of playback speed for historical audio (2, 3, and 4 triangles denote speedup factors of 60, 600, and 3600, resp.), the 'forward' button returns to real-time data and audio.

spans (i.e., months). Few high-level tools exist for this kind of networked audio processing application, so our work also involved the design and implementation of this system.

The resulting system offers several ways to explore data through audio, using the existing DoppelLab user interface, hence the spatial characteristics of the audio playback follow the user's position in the virtual world. Seven microphone streams from the public areas in the Media Lab, obfuscated for privacy, are played back in the application, spatialized according to the locations of their recording in the real space. Spatialized audio streams are attenuated by the inverse square of their sources' distance from the listener, as according to the physical law. As with the other data in DoppelLab, the user may explore real-time or historical audio. In particular, the program allows playback of archived audio at faster-than-real-time rates, using an audio time-compression algorithm of our creation. To protect privacy of users in the space, we obfuscate audio data at the nodes where it is recorded. In addition to this recorded audio sonification, we have implemented a number of synthesized sonifications of other building data; this work is preliminary, but it allows us to explore the issues involved in sonifying spatially-oriented data, and doing so in the context of a 3D virtual world.

2. RELATED WORK

Our sonification leverages results of perceptual studies that show the effectiveness of sonification for understanding; particularly in the techniques of pitch mapping and spatialization. User studies have quantitatively shown the effectiveness of certain kinds of data sonifications for understanding features of graphs, both in seeing and blind users. For example, Brown and Brewster showed how blind users could listen to pitch mappings of time series data, and draw the data from memory with relevant features [5]. In Musical versus Visual Graphs, Flowers and Hauer showed that pitch mapping sonification was as effective as visualization for time series data [6]. Based on these and similar perceptual studies, Brown et al. published a set of guidelines for sonification of time series data, targeting blind users specifically [7]. They recommended pitch mapping, and for multiple variables they recommended stereo separation and allowing the user to change relative amplitude levels; we arrive at these latter two qualities through spatialization. Another study in favor of spatialization is Hunt and Hermanns Importance of Interaction [8], which discusses how sonification is most effective in interactive systems when it behaves like natural acoustic phenomena (e.g. when sound is produced by striking an object, a harder strike produces a louder sound.) Especially when exploring localized data, spatialization is one such phenomenon. My system performs spatialization on the client, rather than the server; this reduces latency, thereby making the interaction more realistic.

Previous studies exist on the use of spatialization for sonification, including its use for spatially oriented data, and in virtual environments. R. Bargar's work on interactive sound for the CAVE system is an early example of augmenting a graphical virtual environment with spatialized sound sources.[9] This system was restricted by performance issues which are no longer problematic; for example, it could only spatialize 4 sound sources; the ability to spatialize many more than that is essential for our ability to sonify large sensor arrays. Nasir and Roberts's Sonification of Spatial Data[10] is a survey of sonification work where either the data is spatial, the sonification uses spatialization, or both. One relevant conclusion is that spatialized sonification can enhance visualization of the same or related data. Andrea Polli's Atmospherics/Weather Works[11] is a striking and ambitious example of spatialized sonification of data with three spatial dimensions. It sonifies several weather related parameters, including pressure, humidity, and wind speed, sampled in a three-dimensional grid. All parameters were mapped to pitch, with different timbres differentiating kinds of data; spatialization was used to locate sounds according to sampling location. Other relevant work includes the spatialized sonification of EEG data[12].

Our system for preserving privacy in audio streams builds on several prior studies. The SPINNER project addressed privacy in distributed audio and video recording [13]. SPINNER uses an opt-in model for sharing of audio and video data, where building occupants wear badges, and recording is enabled only if a badge is present and the user's preferences are set accordingly. Recording may also be stopped manually. This comprises a viable model for privacy, but at costs of requiring badges and throwing away data from chosen time spans outright (as opposed to more selectively obscuring it). Other efforts focused on obfuscating the speech content of audio while preserving timbre. Chris Schmandt's ListenIn system, for example, used audio for domestic monitoring among family members or caregivers [14]. ListenIn scrambles audio by shuffling short buffers whenever speech is detected, aiming to make speech unintelligible, but otherwise preserve timbre. More recent work by Chen et al. alters vowel sounds in speech, and includes a user study which demonstrates significantly reduced intelligibility while leaving concurrent non-speech sounds recognizable [15]. These works do not discuss the question of whether a third party could process the obfuscated audio and restore intelligibility. In [14], this is less relevant because the application is meant to be a closed system where data is shared among a small number of individuals. In Minimal-Impact Audio-Based Personal Archives, Lee and Ellis address the issue of obfuscating audio in a way that would be difficult to reverse [16]. Their method is similar to the one in [14], but they reverse and crossfade short buffers of identified speech in addition to shuffling. They claim that given certain parameters (shuffling 50ms windows over a span of 1s, with "large" overlap between adjacent frames), reversing the obfuscation would be "virtually impossible". They note that this kind of obfuscation should leave spectral features in the audible range largely untouched, so that timbral analysis would not be disrupted. Lee and Ellis note that individuals using their system may wish to listen to old conversations without obfuscation. They discuss the possibility of turning off obfuscation for speakers who have given permission, possibly via voice recognition.

Our work is focused on time compression of a large body of audio data; other work on this topic exists, but most of it differs significantly from our project. One such area is time compression of recorded speech for efficient listening. In order to achieve this goal, relatively small compression ratios are used– generally less than 3. Products in this space date back at least to the mid-80's, e.g., Radio Shack's VSC-1000 VariSpeed Tape Recorder with all-analog pitch restoration. There exist analysis techniques, however, based on speech structure, which can guide compression more effectively than a general (non-speech-related) algorithm could; also, there are specific metrics for evaluation, based on speech comprehension [17], [18]. Another similar study is Tarrat-Masso's work in [19], which uses spectral analysis to guide time compression in music production. The analysis and resynthesis are decoupled, and the analysis isn't specific to musical inputs. The analysis is based on Image Seam-Carving, which involves calculating an energy map of the spectrogram of the audio, and then compressing most during times when there is the least amount of spectral change [20]. This is similar to the method that we will describe in Section 4, although we build on it in several ways, including using a perceptual weighting of the audio spectrum.

This work builds on the Responsive Environments Group's ongoing project, DoppelLab [3]. In this work, we are exploring ways to use sonification to complement 3D graphical visualizations and instill a sense of presence, hence to do so we integrate our implementation with the existing DoppelLab system. DoppelLab is a 3D virtual environment for browsing multimodal sensor data from the MIT Media Lab based around the Unity3D Game Engine. Visualizations of data such as temperature, humidity, and sound are situated throughout a graphical model of the Media Lab. Users can view the data from hundreds of sensors around the lab at once, or zoom in on one location to examine in more detail. Exploring historical data on different time scales is a focus of DoppelLab. By default DoppelLab shows real-time data, but it also provides an interface (pictured in Figure 2) for exploring past data. At the time of this writing, most of the data is archived for around one year. Users can go to a past time, and select the rate at which time will pass. Higher speeds allow the user to view large-scale patterns that would not be apparent in real time; for example, at the fastest speed of one hour per second, it is easy to see people arriving each morning and temperature anomalies every night when the building quiets down (indicating that the heating/air conditioning system has entered non-closed-loop setback).

3. PRIVACY

It is essential to this work that we use audio from the space in a way which respects the privacy of people in it. Our objective in that regard is that spoken language should not be intelligible in the streamed and recorded audio; nor should it be possible for a third party to derive intelligible speech from the audio our system provides. To this end, we process all sampled audio with an obfuscation algorithm which we will describe below, and run the obfuscation directly at the sensor node to prevent clean audio from propagating through the building network at any level. As stated in Section 1, our objectives in using actual audio are to recreate the ambience of the space, and to understand what sort of activity is going on (e.g. quiet discussion, cocktail party, cleaning), and anchor a sense of presence. To this end, our obfuscation algorithm should meet privacy requirements while preserving the timbre of the audio. A final requirement is that the obfuscation algorithm is computationally efficient; this is because it must run on the recording nodes of the system, which could be minimal computers or embedded microcontrollers.

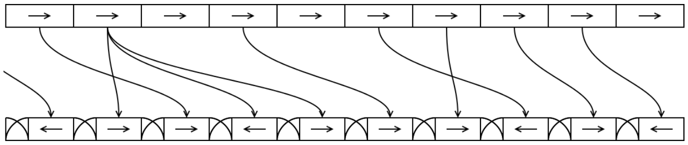


Figure 3: Visualization of the obfuscation process. In this example, we shuffle among the 4 most recent buffers, and reverse with probability 0.5.

3.1. Obfuscation Technique

Our algorithm works in a similar way to the one presented in [14], but attempts to improve on timbral preservation and irreversibility. To that end, we shuffle fewer, larger grains, instead of a larger number of smaller grains. To offset the increase in intelligibility that this causes, we randomly reverse some of the grains. We also significantly overlap and crossfade between sequenced grains, to create a smoother sound.

Currently, the algorithm is running using a grain size of about 200 ms, shuffling among the most recent 3 grains, and reversing grains with probability 0.6. These parameters were chosen on aesthetic grounds, and the result is unintelligible by observation, but the parameters should be revisited to ensure irreversibility. [16] suggests that more aggressive scrambling, i.e. shuffling among more grains of smaller size, while maintaining significant crossfades, would provide more of a guarantee against an eavesdropper reversing the algorithm.

This algorithm preserves the timbre of many environmental sounds. For example, the distinctive bell sound of the Media Lab's elevators is preserved– since this is a relatively constant tone which lasts several seconds, shuffling grains which are significantly smaller than a second has little effect on it. Significant changes in the tone of speech can be heard; laughter, for example, is often recognizable. Short percussive sounds, such as those from closing doors or a ping pong game, are affected by the grain reversal, but since reversal is randomized, if several such sounds occur, some of them are likely to play forward.

3.2. Deobfuscation

In the interest of privacy, it is important not only that our obfuscated audio is unintelligible, but that it is difficult to reverse the obfuscation procedure. Otherwise, a user could download and archive the audio streams and attempt to process them so that the original audio is intelligible; either by manipulating it manually using an audio editor, or by writing a program which analyses the audio and attempts to reconstruct contiguous speech passages.

Currently, we do not know of a way to prove that the obfuscation would be practically impossible to de-obfuscate, or even exactly what that condition would mean. We can make some conjectures. Reversal of the algorithm, if possible, would probably involve either spectrally analyzing grains and finding grain boundaries that best match up, or applying phoneme detection and then using a phonetic model to look for common phoneme sequences. Significant crossfading will make both of these approaches difficult, particularly the first one– with a large logarithmic crossfade, as we get near the edge of a grain, the spectrum of that grain will be mixed close to equally with the spectrum of the next grain; if the time scale on which syllables change is similar to the time scale of the crossfade size, it will be difficult to undo that corruption. If cross-fades comprise a large proportion of the duration of a grain, it will be more difficult for phoneme detectors to work.

Another property of the algorithm that we note is that if we randomly sequence grains from a set of the most recent N grains and assume that a given grain will be in the set during N sequencing intervals, then the probability of that grain never being played is $((N-1)/N)^N$. If we maintain a set of 3 grains, this probability is approximately 0.30; if the set has 10 or 20 grains, the probability is approximately 0.35 or 0.36. Thus, approximately one of every three grains will be dropped. Accordingly, for an algorithm reversing this process, only short contiguous sequences of grains would even exist, and then if those were all identified, the algorithm would have to recognize words with $1/3$ of the audio missing.

4. COMPRESSION

One of DoppelLab's main features is the ability to explore historical data on faster-than-real-time time scales (up to several thousand times) in order to understand larger-scale patterns. Our focus has been on using recorded audio data from the Media Lab to suggest generic activity within the space and enhance DoppelLab's sense of presence. Unlike many graphical visualizations, which can be sped up simply by fetching sequential data at a higher rate, audio data is not trivially sped up. Since we perceive audio data in terms of frequencies, speeding up the data by the kind of factors we deal with in DoppelLab (e.g. 60 to 3600) would bring the data out of the human range of hearing. We have designed and implemented an algorithm for speeding up audio data for this application that has two parts: one part uses granular synthesis to resynthesize audio at any speed without altering its frequency; the other part uses perceptually-based analysis to determine which audio is most interesting and should be prioritized. We experimented with both traversing the input file at a constant speed, and traversing at a variable speed, where more time is spent on audio sections with more interesting features.

4.1. Analysis

In time-compressing audio data, we would like to provide as much information as possible to the listener. To this end, we perform an audio analysis to attempt to determine which parts of the audio data are most interesting. We use this to guide our compression algorithm so that it compresses more aggressively on less interesting audio.

Of course, the question of which audio is more interesting is very open-ended. Given the exploratory nature of DoppelLab, we do not want to restrict the user to a specific type of event, such as human speech or activity. Also, we do not want to simply select audio that has more noise or activity, as this would misrepresent the data. For example, if an interval of time has both loud conversation and silence, we would like to show both. To accomplish this, we use the perceptually-based Bark frequency scale to obtain a compact representation of the audio features that humans perceive in greatest detail [21]. We then look at the amount of change in this representation over time, and bias the compression to preserve these times of transition, and apply more compression to times when the sound is more constant.

The analysis works by taking the FFT of successive windows of input audio, and then reducing the spectrum to a 24-dimensional Bark representation. We then compute the Euclidean distance between successive Bark vectors to estimate a magnitude of spectral change at each window. Intervals with greater spectral change are deemed more interesting. A more detailed explanation of the algorithm can be found in [22].

4.2. Synthesis

To time-compress audio without altering its pitch, we use a version of granular synthesis, as described in Roads' Computer Music Tutorial, known as "time granulation" [23]. In this variant, grains are sampled from an existing audio source. We use the notion of a "playhead", or an offset into the input audio, which determines where grains will be sampled from.

To compute the time-compressed audio, we move the playhead forward through the input file. To compute the version that compresses at a constant rate, we sample grains from the playhead at a uniform density throughout the input. To compute the perceptually-based compression, we sample grains at closer intervals during more interesting sections of audio, and more sparsely during less interesting sections. This procedure is described in much greater detail in [22].

4.3. Results

In this section, we show some data from intermediate steps and output from the time compression algorithms, then discuss the results of informal user testing.

Our process for variable-rate time compression involves a number of stages; Figure 4 shows example input data and intermediate data at successive stages. This example uses a compression ratio of 60. The input data are chosen as an instance where we have some interesting events which we would like to show in higher resolution; the beginning is mostly quiet, we have a few brief periods when people walk by and talk, and then near the end the beginning of a musical jam session in a nearby room can be heard. All four subfigures are on equivalent time scales, so corresponding data line up vertically. The first subfigure is a spectrogram showing the 60 minutes of input data; a spectrogram is a plot which shows successive windows of spectral representation over a longer signal; power is mapped to color. The second subfigure shows, for successive windows, the values of the Bark vector. The third subfigure shows the bark spectrum derivative metric over the same data; the fourth shows the playhead map, which indicates the location in the input file at which the granular synthesizer should be sampling from for each moment in the output file. Note the several more flat regions; these are moments of interest where the playhead spends more time.

Figure 5 shows spectrograms of the minute-long audio outputs from constant- and variable-rate time compression, given the hour of input data shown in Figure 4. The first image, predictably, looks

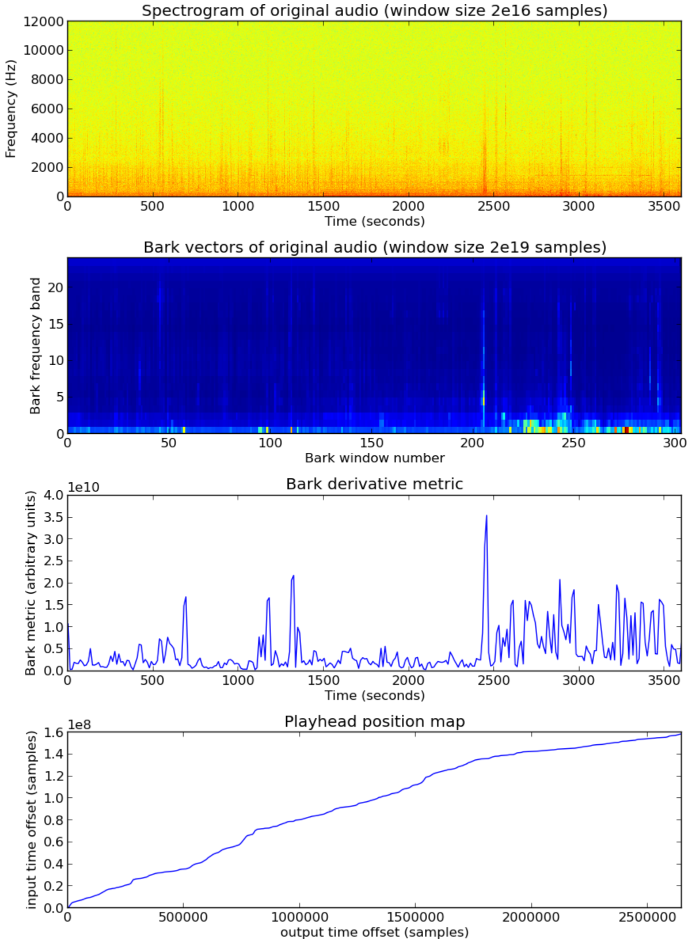


Figure 4: Analysis of 60m audio data: input and successive representations.

similar to the original data in the first subfigure in Figure 4, since we are compressing using a constant speed, and since the images don't have resolution to show the details in the hour-long sample that don't exist in the compressed version. In the second subfigure, as intended, we can see many of the same spectral features, but time is stretched or compressed at different moments in the audio. The "event" annotations point out a few moments where we can clearly see the same audio feature showing up in both outputs, based on the spectral characteristics. In particular, with "event B", we note that the event, which is brief, barely shows up in the first subfigure, and is even more difficult to see in the input data. In the second subfigure though (the variable-rate-compressed stream), the event is bigger and darker, and better distinguished from the peak immediately to the left of it. Similarly, with "event C", we see a pair of short, dark peaks, and some lighter peaks to their left; in the second subfigure, all this dense activity is more spread out in time. These two examples illustrate the variable-rate compression algorithm dwelling on events with more change, or with unusual frequencies, over relatively longer periods of time, thus rendering them with higher resolution--an effect that can be clearly noticed when listening to the audio. This is at the expense of more monotonous sections, like the several more static periods early in these spectrograms.

We informally evaluated the time compression program by testing it and having several colleagues test it. In testing it ourselves, certain events were recognizable through the time compression; for example, social gatherings and music. In comparing the constant- and variable-rate compression side by side, the variable-rate compression spread out the activity more. Also, certain short features were clipped in the constant-rate version and noticeably longer in the other one.

We then did an informal user test with two colleagues; Participant 1 was familiar with the project beforehand, and Participant 2 was not. For the test, subjects listened to four minute-long clips of time-compressed audio; output of both the constant- and variable-rate compression algorithms, run on 60 minutes of audio with compression rate 60, and then on 600 minutes of audio with compression rate 600. For each of the two compression ratios, subjects first heard the constant-rate output, and then were asked what they noticed; then they heard the variable-rate output and were asked what differences they noticed. The audio clips were both chosen from a weeknight at our lab, including conversations, a musical jam session, and custodians vacuuming. Of the constant-rate compression on 60 minutes, Participant 1 thought s/he heard indication of several-minute-long conversations $2/3$ of the way through, and the "usual background noise" of the Media Lab, with doors opening and closing, and a hint of something melodic towards the end. Participant 2 noted that it sounded like a room with conversations and silences, and that it sounded "natural", as opposed to sounding like a compression. Both participants reported hearing more activity during the variable-rate compressed audio. Participant 2 said that these sounded more "full", and that it sounded like there was a boost in mid-range frequencies. Participant 1 noticed that during the 10-hour compressed audio, he heard what sounded like a sequence of chords, which he speculated be the dominant harmonies of a series of songs; on hearing the variable-rate compressed version, he reported hearing more 'attack', and more of the transients, during the musical sequence.

This testing suggests that in general, the compression is successful in conveying main events that occur in our lab, as well as the general sonic character of the space. However, the difference

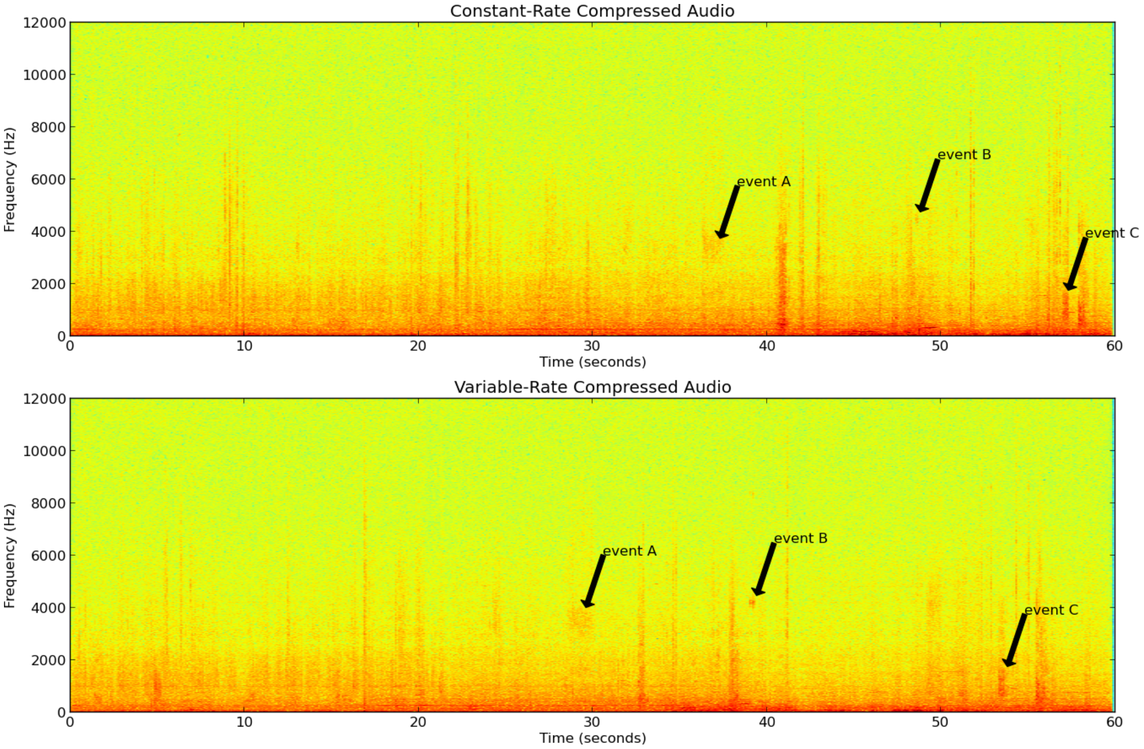


Figure 5: Comparison of outputs of constant- and variable-rate time compression. Compression ratio is 60; input is the input data from Figure 4

made by the variable-rate compression was more subtle; in some cases, users seemed to hear it as a change in timbre rather than something that enabled perception of new effects. Also, while comments suggest that the variable-rate compression is working by finding more activity, it is not clear that perceiving more activity is desirable; perhaps the distilling of audio to times of greatest change could be accompanied by a smoothing or averaging of timbre. A more substantial user study would better tease out the perceptual effects of different compression parameters (e.g., how extreme the bias towards more interesting audio should be).

5. SONIFICATION

In addition to exploiting sampled audio streams, we include some preliminary work on synthesized spatial sonification of other localized sensor data. The sonifications in this section are implemented using Max/MSP, which was sent data from the main DoppelLab client over Open Sound Control [24], [25]. We consider two broad categories of sonification: that of continuous data, such as temperature, where sensors have numerical values which vary over time, and event-like data, where sensors register discrete events as they occur.

5.1. Continuous Data Sonifications

DoppelLab incorporates streams of temperature data from a network of between two and three hundred temperature sensors around the Media Lab. Our temperature sonification picks a subset of those sensors and assign a sine-wave oscillator to each, spatialized at the corresponding locations. Temperatures are mapped proportionately to the frequency of the sine waves; both linear and logarithmic mappings of pitch were tried; the latter was more successful since a given temperature difference in degrees always corresponds to the same harmonic interval.

The resulting soundscape has a resonant, bell-like character. The lack of pitch quantization and the large number of voices give an effect reminiscent of later 20th century music such as the micropolyphony of Gyorgi Ligeti. While the continuum of pitches is at odds with the suggestion of discrete pitches in [7], it allows for nuanced presentation of this dataset, where at real-time or near-real-time speeds, data often changes very gradually.

5.2. Event-Like Data Sonifications

Several data types in DoppelLab are event-like, where discrete events are associated with a physical location and particular point in time. For this kind of data, we might use a sonification where we assign a transient note or sample to the events; this way, the density of sounds in time, or large-scale rhythmic structure, encodes large-scale patterns in the data. If data is sparse, the sonification can act as an alert to indicate the presence of new data. We tried sonifying two such types of data; RFID data (indicating the proximity of a badged individual to distributed readers) and Twitter streams (which graphically manifest in the office of the Tweeter as they arrive).

In both of these cases, we found the most salient data to sonify to be the individual's username. Since this is not quantitative data,

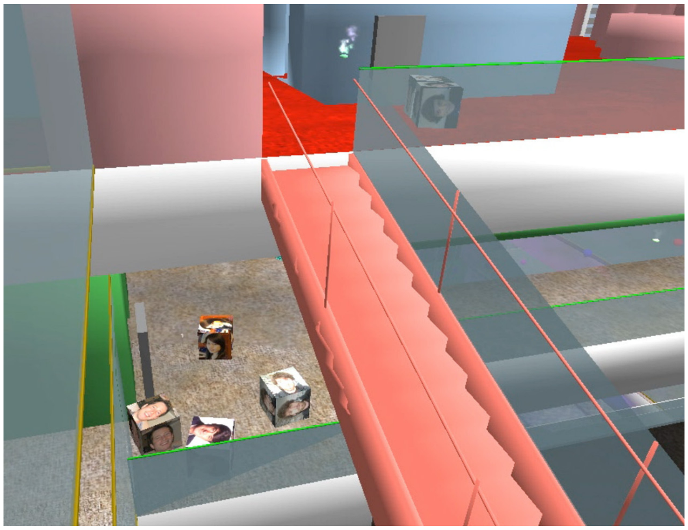
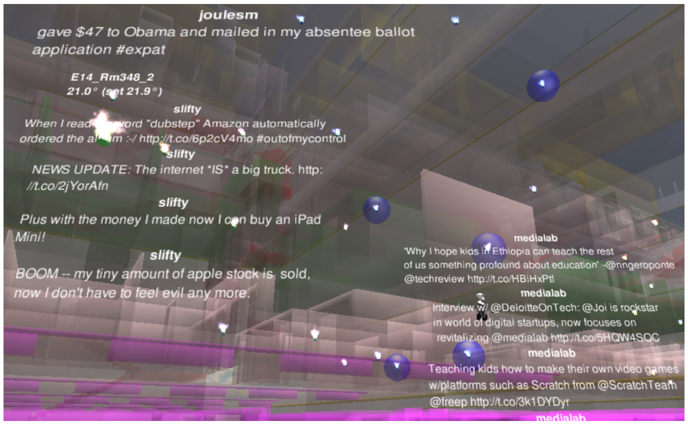


Figure 6: Cubes with faces on them appear when corresponding RFID badges are near sensors.



joulesm
gave $47 to Obama and mailed in my absentee ballot application #expat

E14_Rm348_2
21.0° (set 21.9°)

slifty
When I read the word "dubstep" Amazon automatically ordered the album :-/ http://t.co/6p2cV4mo #outofmycontrol

slifty
NEWS UPDATE: The internet *IS* a big truck. http://t.co/2jYorAfn

slifty
Plus with the money I made now I can buy an iPad Mini!

slifty
BOOM -- my tiny amount of apple stock is sold, now I don't have to feel evil any more.

medialab
'Why I hope kids in Ethiopia can teach the rest of us something profound about education' -@nngeroponte @techreview http://t.co/HBiHxPtl

medialab
Interview w/ @DeloitteOnTech: @Joi is rockstar in world of digital startups, now focuses on revitalizing @medialab http://t.co/5HQW4SOC

medialab
Teaching kids how to make their own video games w/platforms such as Scratch from @ScratchTeam @freep http://t.co/3k1DYDyr

medialab

Figure 7: Twitter streams are rendered in space near the office location of their authors. They update when new tweets appear.

it cannot be directly expressed by mapping it to a musical parameter. We used hashing to associate the username with the chosen parameters; the goal is not to 'encode' the username in a meaningful way, but to create an association, so that if a username appears frequently, the listener may learn to recognize the associated sound.

DoppelLab's RFID data provides a rough-granularity representation of where different lab members or visitors are in the building. Many members and visitors choose to carry a badge which has an RFID tag; if the tag is detected at one of the RFID readers around the lab, DoppelLab registers the event, and some associated data, such as a name, and if the tag owner has submitted one to the Lab-wide database, a photo. DoppelLab includes a visualization of this data, wherein a cube appears if a tag is registered in the corresponding location; if a photo is available, the photo appears on the cube's faces. Figure 6 shows a screenshot.

To sonify an RFID event, we used a simple, short bell-like tone. Usernames are associated with pitches; pitches are chosen among multiples of a base frequency, for a just-intonation effect; tones in a western scale could also be used. One important characteristic of the RFID data is that it has great variance in frequency of occurrence; on a normal day, around 10 RFIDs might appear, as most students and academic staff dont carry their badges; during semesterly sponsor meetings, however, hundreds can appear within an afternoon; this density is compounded in the case of faster-than-real-time playback. In these cases, the sounds from such events are triggered sequentially, so the effect is a fast rhythmic pattern.

DoppelLab also includes Twitter data [26]. Public tweets from lab members are aggregated, and the several most recent tweets are rendered in the visualization, situated according to the office location of the account's author. Figure 7 shows an example.

For the Twitter sonification, we use samples to denote events. In particular, we chose a set of bird call samples, primarily for aesthetic interest and to provide a clear metaphoric association between the sonification and the data modality in the presence of other sonifications. The greater sonic variety between the bird samples, as compared to the synthesized RFID sounds, made it relatively easier to associate the sounds with stream authors. On the other hand, this complexity can be problematic at the higher frequencies of events when running highly-compressed timestreams; the sound of many such extended samples playing simultaneously is difficult to sonically parse, although that may be partly because we were unfamiliar with avian sounds—birders, for example, are better at distinguishing particular calls in relative caucophony.

5.3. Results

We evaluated these sonifications through informal testing. Even though the work is preliminary, testing yielded a number of insights particular to using sonification to augment a 3D graphical environment.

With event-like data (Twitter and RFID), one of the main goals of the sonification is to give the user an awareness of events which they are not currently looking at (either because the user's eyes aren't on them or because the UI's camera is not on them.) These sonifications are successful to the extent that they indicate the frequency and general spatial distribution of events. One problem was that connecting a sonic event to the specific visualization it corresponded to was difficult. When a sound occurs, a number of RFID cubes or Twitter streams might be present near the location of the sound, and then it is not clear which of those events has just taken place. The Twitter stream visualizations are particularly problematic, because they are always shown; new data simply changes the content they show.

Thus, our tests indicate that it is difficult to learn these purely associative sonifications. One way to improve this situation might be to allow occupants to submit meaningful sonic signatures that would be more meaningful than arbitrarily assigned sounds. Another way would be to make the sonification more interactive through the GUI. In particular, if a user could click on a stream or an RFID cube and hear the associated sound, she might begin to learn the association between IDs and sounds.

6. SYSTEM

We implemented this sonification as a networked software system. Priorities were scalability, privacy, and responsiveness to user input. Audio obfuscation (see Section 3) is performed on the nodes where audio is recorded, before it is sent over the network. Obfuscated audio is sent to a streaming server, where clients which are looking at real-time data can download it. A central server also downloads the audio streams and archives them, and periodically computes time-compressed versions of the archived audio at several compression ratios (see Section 4). Normal-speed and time-compressed audio are served from a file server in minute long segments. We have implemented a native client program, which gets player position data from the DoppelLab game-engine client, downloads all audio streams at the desired time and speed, spatializes the streams, and plays them back. Our client uses the OpenAL library[27], to perform the spatialization using the inverse square law for intensity attenuation with distance. In [22], we provide more details on the server design, and some calculations demonstrating scalability and responsiveness.

7. CONCLUSIONS

Increasingly, the spaces in which we live and work are instrumented with sensors. To understand this increasingly dense and multimodal data, we will need increasingly powerful ways to display it. This work combines spatialized sonification of localized sensor data with interactive 3D visualization to demonstrate a new kind of immersive display. In addition, this work explores the use of recorded audio data as a timbral sonification of a space, and how to use extreme time compression to make this audio data understandable; these topics have seen little prior work. Finally, this work poses questions about data privacy that will become increasingly relevant, and takes steps toward finding a solution which best suits this new use of audio data.

This work touches on many rich topics, including audio analysis, compression, sonification, and privacy. There are various interesting possibilities for future work. The presentation of historical recorded audio would likely benefit from audio visualization (e.g., a scrubbable "waveform" above the time-scroll bars), to help users in navigating to particular features. Our exploration of sonification of non-audio data in DoppelLab was somewhat cursory and leaves much room for future work. One interesting area is sonification of associative or non-quantitative data, such as user IDs; making these sonifications interactive could improve learnability. The methods of audio spatialization used could also be varied and tested. In particular, DoppelLab allows users to toggle the visual transparency of the building representation; it would be natural to simulate audio attenuation through floors when the floors are graphically opaque. This work in general would benefit from more formal user studies with more participants; such studies could further investigate the quality of the time compression, or could focus on the quality of the obfuscation, or the overall usefulness of the sonification within the DoppelLab program. Our research group is also continuing this work by deploying a similar system in an outdoor area, so that users can observe changes to the natural habitat via audio streams and sensor data. The notion of privacy in dealing with recorded audio is essential to work in this area. We would like to explore more rigorous ways to ensure that audio obfuscation is not reversible, while preserving timbre and aeshtetics as much as possible.

The website[28] for this project includes audio and video material from the programs discussed here; this includes demonstrations of the time-compression, obfuscation, and the DoppelLab program as a whole.

8. ACKNOWLEDGMENT

We acknowledge the help and support of members of the Responsive Environments Group at the MIT Media Lab, particularly Gershon Dublon, as well as Dafydd Harries and Rob Ochshorn.

9. REFERENCES

[1] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.

[2] E. R. Tufte, *The Visual Display of Quantitative Information*. CT, USA: Graphics Press Cheshire, 1986.

[3] G. Dublon, L. Pardue, B. Mayton, N. Swartz, N. Joliat, P. Hurst, and J. Paradiso, "Doppellab: Tools for exploring and harnessing multimodal sensor network data," in *Sensors, 2011 IEEE*. IEEE, 2011, pp. 1612–1615.

[4] J. Lifton, M. Laibowitz, D. Harry, N.-W. Gong, M. Mittal, and J. Paradiso, "Metaphor and manifestation cross-reality with ubiquitous sensor/actuator networks," *Pervasive Computing, IEEE*, vol. 8, no. 3, pp. 24–33, July-Sept.

[5] L. Brown and S. Brewster, "Drawing by ear: Interpreting sonified line graphs." International Conference on Auditory Display, 2003.

[6] J. Flowers and T. Hauer, "Musical versus visual graphs: Cross-modal equivalence in perception of time series data," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 3, pp. 553–569, 1995.

[7] L. Brown, S. Brewster, S. Ramloll, R. Burton, and B. Riedel, "Design guidelines for audio presentation of graphs and tables." International Conference on Auditory Display, 2003.

[8] T. Hermann and A. Hunt, "The importance of interaction in sonification," in *In Proceedings of International Conference on Auditory Display (ICAD)*, Sydney, Australia, 2004.

[9] R. Bargar, I. Choi, S. Das, and C. Goudeseune, "Model-based interactive sound for an immersive virtual environment," in *Proceedings of the International Computer Music Conference '94*, 1994, pp. 471–474.

[10] T. Nasir, J. Roberts, *et al.*, "Sonification of spatial data," in *The 13th International Conference on Auditory Display (ICAD 2007)*. ICAD, 2007, pp. 112–119.

[11] A. Polli, "Atmospherics/weather works: A spatialized meteorological data sonification project," *Leonardo*, vol. 38, no. 1, pp. 31–36, 2005.

[12] G. Baier, T. Hermann, and U. Stephani, "Multi-channel sonification of human eeg," ser. Proceedings of the 13th International Conference on Auditory Display, W. L. Martens, Ed. Schulich School of Music, McGill University, 2007, pp. 491–496.

[13] M. Laibowitz, N. wei Gong, and J. A. Paradiso, "Wearable sensing for dynamic management of dense ubiquitous media," in *6th Int'l Workshop on Wearable and Implantable Body Sensor Networks (BSN 09)*, 2009, pp. 3–8.

[14] C. Schmandt and G. Vallejo, “”listenin” to domestic environments from remote locations,” in *Proc. the 2003 International Conference on Auditory Display*, Boston, MA, USA, 2003, pp. 853–856.

[15] F. Chen, J. Adcock, and S. Krishnagiri, “Audio privacy: reducing speech intelligibility while preserving environmental sounds,” in *Proceedings of the 16th ACM international conference on Multimedia*, ser. MM ’08. New York, NY, USA: ACM, 2008, pp. 733–736. [Online]. Available: http://doi.acm.org/10.1145/1459359.1459472

[16] D. Ellis and K. Lee, “Minimal-impact audio-based personal archives,” in *Proceedings of the the 1st ACM workshop on Continuous archival and retrieval of personal experiences*. ACM, 2004, pp. 39–47.

[17] N. Omoigui, L. He, A. Gupta, J. Grudin, and E. Sanocki, “Time-compression: systems concerns, usage, and benefits,” in *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*. New York, NY, USA: ACM, 1999, pp. 136–143.

[18] L. He and A. Gupta, “Exploring benefits of non-linear time compression,” in *Proceedings of the ninth ACM international conference on Multimedia*, ser. MULTIMEDIA ’01. New York, NY, USA: ACM, 2001, pp. 382–391.

[19] J. M. Tarrat-Masso, “Adaptation of the seam carving technique for improving audio time-scaling,” Master’s thesis, Pompeu Fabra University, 2008.

[20] S. Avidan and A. Shamir, “Seam carving for content-aware image resizing,” in *ACM Transactions on graphics (TOG)*, vol. 26, no. 3. ACM, 2007, p. 10.

[21] E. Zwicker, “Subdivision of the audible frequency range into critical bands (frequenzgruppen),” *The Journal of the Acoustical Society of America*, vol. 33, p. 248, 1961.

[22] N. D. Joliat, “Doppellab: Spatialized data sonification in a 3d virtual environment,” Master’s thesis, Massachusetts Institute of Technology, 2012.

[23] C. Roads, *The Computer Music Tutorial*. Massachusetts Institute of Technology, 1996.

[24] “Max/msp,” http://cycling74.com/products/max/, accessed: 2012-10-09.

[25] “Open sound control,” http://opensoundcontrol.org/, accessed: 2012-10-10.

[26] “Twitter,” https://twitter.com, accessed: 2012-10-03.

[27] “Openal documentation,” http://connect.creativelabs.com/openal/Documentation/Forms/AllItems.aspx, accessed: 2012-10-03.

[28] “Doppellab sonification web page,” http://resenv.media.mit.edu/sonification, accessed: 2012-10-15.