Spatialized Anonymous Audio for Browsing Sensor Networks via Virtual Worlds

Nicholas Joliat, Brian Mayton, and Joseph A. Paradiso

Responsive Environments Group
MIT Media Lab
# Table of Contents

- Existing work: DoppelLab
- Overview of DoppelLab Audio Behavior
- [DoppelLab Demo]
- Privacy
- Time Compression
  - Linear and content-aware methods
- Client implementation
- System implementation
- Future Work
Existing work: DoppelLab

- 3D virtual environment for browsing sensor data
- Temperature, humidity, sound level, movement, opt-in ID recognition, etc.
- Explore trends over larger time scales
Spatialized Anonymous Audio Overview

- Use recorded audio to show activity, immerse user
- Multiple recording locations, spatialized playback
- Privacy; obfuscation
- Realtime and Historical operation
- Time compression, linear and variable-rate

[demo: quick DoppelLab run-through]
Obfuscation I

● **Objective:**
  ○ hard to understand speech
  ○ preserve timbre (can we hear what's happening? how many people are there?)
  ○ hard to computationally unscramble

● **Prior art:**
  ○ Schmandt:
    ■ Shuffle recent buffers
    ■ Doesn't address reversibility
  ○ Lee, Ellis:
    ■ for certain parameters "virtually impossible" to reverse
Obfuscation II

- Shuffling, crossfading, reversing

- Algorithm runs at audio recording sites, so we don't transmit clear audio
- Computationally lightweight

[demo: obfuscation]
Time Compression

- Goal:
  - Speed up audio fast enough to hear on other time scales
  - Preserving speech is not an issue
  - Preserve timbre, and moments of interest
Time Compression: Algorithm

- Granular synthesis around a playhead
  - warp speed, preserve pitch
- Window grains, randomize size
- Move playhead faster than real-time
- Rendered offline, at ratios \{60, 600, 3600\}
Variable-Rate Compression

- Spend more time on moments of interest; compress monotonous audio more
- Bark metric bins frequencies according to critical bands of hearing
- Magnitude change in bark vector indicates activity in spectrum or amplitude
- Use that metric to control playhead speed
Variable-Rate Compression

- Playhead speed inversely proportional to Bark magnitude derivative
Linear vs. Variable Compression

Constant-Rate Compressed Audio

Variable-Rate Compressed Audio

[audio: two compression methods]
Client / UI

- Download audio streams, spatialize relative to avatar
- Spatialization is done using OpenAL, using physical inverse square rolloff
- Stream real-time, historical, or time-compressed data, according to DoppelLab time travel GUI input
System / Implementation

- microphones
  - obfuscation scripts
    - icecast streaming
      - archiver scripts
        - time compression scripts
          - archived audio
            - historical and time-compressed audio
            - real-time recorded audio
              - sensor data
                - doppellab server
                  - timing requests
                    - sensor data
                      - sonification max patches
                        - synthesized data sonifications
                          - user input
                            - data visualizations, GUI
                              - doppellab playhead and spatial data
                                - spatialization plugin
                                  - spatialized audio stream display
                                    - user
System / Implementation

- Obfuscated ogg/vorbis audio streams to central streaming server
- Archiving scripts save streams in one-minute increments in directory tree
- Time compressed-audio is pre-computed at 4 speeds
- Client asynchronously fetches sequences of one-minute files
- server (archiving, dsp): python, gstreamer, numm.
- client: c, OpenAL
Ongoing & Future Work

- Ongoing: tidmarsh (http://tidmarsh.media.mit.edu) will use similar spatialization and time-compression, for changing outdoor ecosystem
- Audio visualization in GUI (with DoppelLab or standalone)
- Test different time compression parameters
- Spatialization: add some physics (e.g. floors attenuate more than empty space)
- More rigorous study of privacy
- Better audio quality (better mics, dynamic range compression on stream)
We thank:

- Gershon Dublon and Laurel Pardue, DoppelLab developers with whom we collaborated
- Rob Ochshorn and Daf Harries, for advice and for numm software
- Mary Murphy-Hoye and Intel
papers, other media

http://resenv.media.media.mit.edu/sonification