Auracle

Voice-Controlled Interactive Audio on the Internet

The Auracle Team

- Max Neuhaus, project director
- Phil Burk, technical director
- Sekhar Ramakrishnan
- Kristjan Varnik
- Jason Freeman
- David Birchfield

in association with Akademie Schloss Solitude

Introduction

Auracle is a networked sound instrument, controlled by the voice. It is played and heard over the Internet. To participate, simply launch Auracle in your web browser, join an ensemble, and jam with other participants in real time.

What is Auracle?

- an architecture, not a composition
- a permanent web-based entity
- a sophisticated voice-controlled musical instrument
- a real-time collaborative environment

Key Goals

- accessibility: for a lay public, i.e. non-musicians and non-computer geeks
- transparency: users hear their contribution to the output sound
- complexity: remains interesting over extended periods of time
- self-evolution: changes over time in response to how users interact with it

Architecture

low-level analyzer

mid-level analyzer

high-level analyzer

ensemble model

Internet

server

synthesizer(s)

mapper

Low-level Analyzer

low-level analyzer

mid-level analyzer

high-level analyzer

ensemble model

Internet

server

synthesizer(s)

mapper

Low-level Analyzer

Goal: *Extract continuous characteristics of vocal input.*

- pitch
- first and second formants (“eee” vs. “ooo”)
- loudness
- voicedness (“shh” vs. “aaa”)

Low-level Analyzer

How We Do It: *Linear Predictive Coding*

- works on 40 ms blocks of audio data
- efficient real-time implementation in Java

Mid-Level Analyzer

low-level analyzer

mid-level analyzer

high-level analyzer

ensemble model

Internet

server

synthesizer(s)

mapper

Mid-Level Analyzer

Goal: *Break the audio into individual gestures.*

- find gesture boundaries
- optimize gesture data for transmission over the Internet
- compute statistics about each gesture to feed to high-level analyzer

Finding Gestures

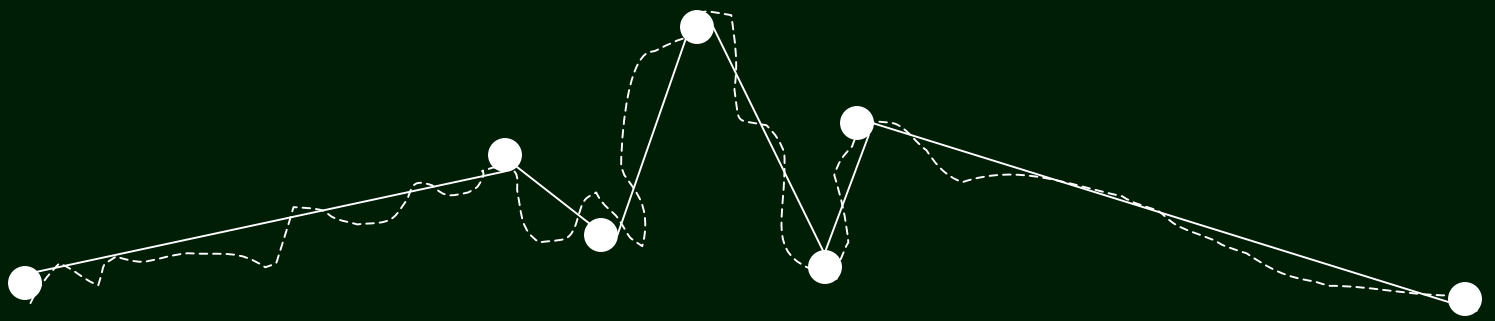
How We Do It: *Moments of silence are gesture boundaries.*

Future work: *Group gestures into phrases.*

Data Reduction

Problem: Too much data to transmit full gestures over Internet.

How We Solve It: Send minimum number of points that preserve shape.



Gesture Statistics

- inspired by research on emotion recognition in speech
  - Scherer (University of Geneva)
  - Yacoub, Simske, Lin, Burns (Hewlett-Packard)
- statistics on amplitude, fundamental, formants:
  - contour statistics (min, max, mean, standard deviation)
  - first derivative statistics
- audible durational statistics

High-Level Analyzer

low-level analyzer

mid-level analyzer

high-level analyzer

ensemble model

Internet

server

synthesizer(s)

mapper

High-Level Analyzer

Goal: *Use a small number of attributes to describe all possible gestures.*

- attributes are abstract
- e.g., 75% attribute “alpha”, 42% attribute “beta”, 91% attribute “gamma”
- learns from user activities over time

High-Level Analyzer

How We Do It: *Uses machine learning.*

- neural network
- statistical analysis
  (Principal Components Analysis)

Ensemble Model

low-level analyzer

mid-level analyzer

high-level analyzer

ensemble model

Internet

server

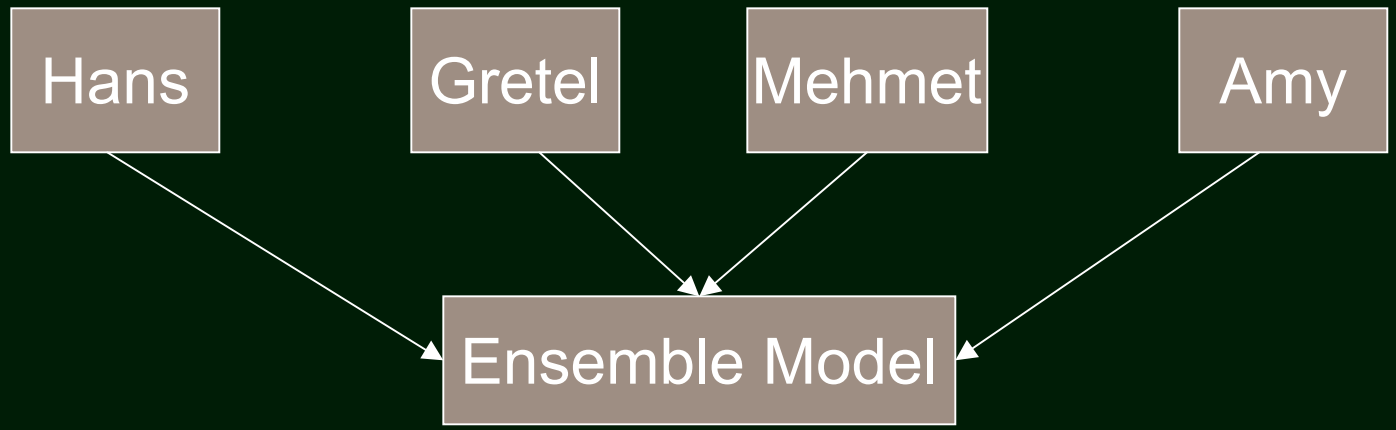
synthesizer(s)

mapper

Ensemble Model

Goal: *Keep track of all the players in the ensemble.*

- store the results of low, mid, and high-level analysis for each player in the ensemble



Server

low-level analyzer

mid-level analyzer

high-level analyzer

ensemble model

Internet

server

synthesizer(s)

mapper

Server

Goal: *Connect players together for real-time interaction.*

- distribute player gestures to participants
- enable users to log in and join an ensemble
- remember what users do (for debugging and evolution)

Server

How We Do It: *TransJam (by Phil Burk)*

- general-purpose server for online collaborative music applications
- does not analyze or synthesize data (merely a conduit)
- receives and transmits gesture data (no audio streaming)

Mapper

low-level analyzer

mid-level analyzer

high-level analyzer

ensemble model

Internet

server

synthesizer(s)

mapper

Mapper

Goal: *Convert gesture data into sound.*

- scale and transform data from ensemble model to control synthesis
- not trying to recreate original sound
- e.g., as input gets louder, output gets noisier

How We Do It: *pragmatic approach via ongoing experimentation and evaluation*

Synthesizer

low-level analyzer

mid-level analyzer

high-level analyzer

ensemble model

Internet

server

synthesizer(s)

mapper

Synthesizer

Goal: *Generate interesting range of timbres controllable by a variety of parameters.*

- transparent relationship between input gesture and output sound
- complex and engaging sound environment

Synthesizer

How We Do It: *A variety of synthesis techniques.*

- simulate physics of real instruments
- use JSyn (by Phil Burk), a fast and powerful Java synthesis plugin for web browsers
- development interfaces (via OSC) to Wire, MAX/MSP, SuperCollider, etc.

Managing Remote Collaboration

- Auracle team split between USA, Italy, and Germany
- ongoing interaction via:
  - meetings in Stuttgart, Internet conference calls
  - twice weekly “jams” on current development build
  - unstructured collaboration tools (mailing list, Wiki)
- online administrative tools (version control system, bug tracker, group calendar and task list)
- “eXtreme Programming” practices:
  - automated nightly integration builds and testing
  - group task-management and bug-tracking databases

Timeline

- April 2004: preliminary release
- July 2004: preview release
- September-October 2004: public launch

For more info:

http://www.auracle.org

auracle@auracle.org