INTRODUCTION TO COMPUTER MUSIC
SPECTRAL INTERPOLATION SYNTHESIS

Roger B. Dannenberg
Professor of Computer Science, Art, and Music

SPECTRAL INTERPOLATION SYNTHESIS

Roger Dannenberg
Marie-Helen Serra
Dean Rubine
Istvan Derenyi
Hank Pelerin
Ning Hu
Additive Synthesis and Table Lookup Synthesis

• Additive Synthesis:
  • Every partial has independent frequency and amplitude
  • Analysis/synthesis possible, but no simple parametric control

• Table-Lookup Synthesis
  • Relative amplitudes of all partials are locked in
  • Frequencies are all harmonic

Table Lookup Oscillator Review

```c
float table[513] = { /* some waveform */ };
double phase = 0.0;
void osc(double hz, float table[], float out[]) {
    double incr = hz * 512 / sample_rate;
    for (int i = 0; i < block_size; i++) {
        int iphase = floor(phase);
        double x1 = table[iphase];
        out[i] = x1 + (phase - iphase) * (table[iphase+1] - x1);
        phase += incr;
        if (phase > 512) phase = phase - 512;
    }
}
```
Spectral Interpolation

- Interpolate between tables
- Keep phases coherent so that interpolation is truly interpolation of spectra
- Restricted to harmonic spectra

Spectral Variation by Interpolation

- Reload tables with new spectra
- Relatively slow update (~20 tables/second)
- Where do tables come from?
Use Pitch and Amplitude to Compute Spectra

Measuring Spectra

- We use phase-vocoder style analysis on
- Crescendo at different pitches
- Yields 2-D (pitch, amplitude) tables of spectra
Attacks Are Too Rapid And Inharmonic

- Solution: use sampled attacks (~30ms)
- New Problem: How do you join attack to synthesized sound?
  - Cross-fade does not work – too many phase problems
  - Make attack long enough to settle to harmonics
  - Analyze phase of every partial at end of attack
  - Synthesize tables with matching phase
  - Splice with very short cross-fade (2ms) or none

WHERE DOES CONTROL INFORMATION COME FROM?
Traditional Synthesis Research

Control ➔ Synthesis Algorithm ➔ Sound

Score ➔ Sound

Compare

Traditional Synthesis Research

Control ➔ Synthesis Algorithm

Score ➔ Sound

Compare
SIS Research Approach

1. Score
2. Control
3. Synthesis Algorithm
4. Sound

ICM Week 12

Copyright © 2002-2013 by Roger B. Dannenberg
Divide-and-Conquer

1. Score
   - Apply Performance Model
2. Control Signals
   - Apply Synthesis Model
   - Audio

Research Model: Synthesis Refinement

- Score
  - Human Performance
- Audio
  - Parameter Extraction
- Control Signals
  - Apply Instrument Model
  - Audio
  - Human Comparison, Refine Instrument Model
Research Model: Control Refinement

A STUDY OF TRUMPET ENVELOPES
Introduction

• Envelopes:
  • are crucial to sound synthesis;
  • depend upon context.
• We claim:
  • envelopes largely determined by context;
  • envelope generation techniques can improve synthesis.

The Big Picture
Related Work

- Moorer, Grey, and Strawn ‘78
- Clynes ‘85
- Chafe ‘89
- Canazza, De Poli, Roda’, and Vidolin ‘97
- Sundberg, Askenfelt, and Fryden ‘83
- Key: consider context, study envelopes in musical phrases

An Experiment

**Question**: How do factors affect the center of mass* (first moment)?

*Why Center of Mass?*: simplest measure of shape.
What Did We Measure?

- Notes and envelopes defined by inter-onset times.
  - because end-of-note not well defined
  - “micro pauses” incorporated into data

Some Results

- Up-up phrases showed later center of mass than other combinations
- Large intervals had earlier center of mass than small intervals
- Legato articulation gave later center of mass than others (this should be obvious).
BACK TO SPECTRAL INTERPOLATION SYNTHESIS

Looking at “Real” Envelopes
Tongue and Breath

Envelope of a Slurred Note
Envelope of a Slurred Note

Toward an Envelope Model

- Breath envelope gives overall shape
- Tongue articulates beginning and ending
Amplitude Envelope Specification

Many shapes approximated by a “general” envelope function:

Computing Parameters

- Previous slide shows 9 detailed parameters
- Compute them from:
  - pitch (in semitones, according to score)
  - dur (in seconds, according to score)
  - begin-phrase (is this the first note in a phrase?)
  - end-phrase (is this the last note in a phrase?)
  - from-slur (is there a slur from preceding note?)
  - to-slur (is there a slur to the next note?)
  - direction-up (is this note higher than preceding note?)
Method

- Score provides “actual” parameters
- Study real performances
- Performance provides envelopes
- Find detailed parameters manually
- Generalize from observed trends

Example: \( tf \)

\[
\text{cond (from-slur)
  (setf takefrom (if direction-up 0.1 0.04)))
  (t
    (setf takefrom (+ 0.03 (* 0.01 (- log-dur)))))
\]

\[
\begin{array}{|c|c|}
\hline
\text{direction-up} & \text{takefrom} \\
\hline
\text{t} & 0.1 \\
\text{f} & 0.04 \\
\hline
\end{array}
\]

\[
\text{t} = 0.03 - 0.01 \times \log_2(\text{dur})
\]
Example: \((st, pr)\)

\[
\text{end-phrase} \rightarrow (0.02, 0.02) \\
\text{to-slur} \& \text{direction-up} \rightarrow (0.03, \text{dur} \times 0.2) \\
\text{to-slur} \rightarrow (0.02, 0.001) \\
o.w. \rightarrow (0.06 + 0.005 \times \log_2(\text{dur}), \\
0.03 – 0.01 \times \log_2(\text{dur}))
\]

Envelope from Score

- We use a 9-parameter envelope model
- One “reference” breath envelope is cropped and stretched (2 parameters + duration)
- Breath envelope is multiplied by attack and decay envelopes to derive final envelope
- Envelope depends on: pitch, pitch context, articulation, next note articulation, duration.
- Score-to-envelope mapping is hand-crafted.
Frequency Envelope

- Use analyzed frequency envelopes
- Capture natural frequency variation
- Pitch dependent
- No vibrato (yet)

Sound Examples

- CSIS Synthesis Example
- Another CSIS Synthesis Example
- CSIS (Without Accompaniment)
Summary

• Modeled from “real” performance
• Phrase-at-a-time
• Melodic direction (lookahead)
• Articulation (lookahead)
• Duration
• Pitch
• Currently rule-based, ad-hoc

Conclusions

• Envelopes are critical to music synthesis
• Statistically valid relationships between score parameters and envelope shape
• Breath + Tongue Model
• Study of musical phrases and notes in context is critical to future synthesis research.
So What?

• Based on these examples (and others), I claim we have an adequate “Synthesis Algorithm.”
• What happens if we return to the “Traditional Synthesis Research” model with the CSIS synthesis algorithm?
SIS With “Standard” Envelopes

- Amplitude, Time, Duration are correct
- Envelope shape is “correct”
- Timbre is correct
- Two Envelopes (better slurs)

- SIS model with computed envelopes

Discussion

- Since the only “wrong” ingredient is envelope shape, envelope shape must be critical for synthesis.
- This observation could only be made after we perfected (?) trumpet synthesis.
More Discussion

• It follows from our results that…

*Previous work on synthesis was wrongly assuming simple envelope templates are sufficient.*

• In other words…

*If you had a good synthesis algorithm, how would you know it?*

Still More Discussion

• Equal time for the physical modelers:

*My goal is to develop sufficiently rich instruments that musicians will want to learn how to control them.*

(paraphrasing Julius Smith)
ALGORITHMIC CONTROL OF SIGNAL PROCESSING

Roger B. Dannenberg

Generating Control and Audio Algorithmically

- Xenakis: GENDYN
- Roads: Microsound
- Granular Synthesis

- Signals (Sounds) controlled by Patterns
- Patterns controlled by Signals
Pat-ctrl

\[
pat-ctrl(make\-cycle\{(0.1\ 0.2)\}, \text{;duration}\nn\text{\small make\-cycle\{(0\ 1\ 2)\}} \text{;amplitude}
\]

• What is the duration of a sound returned by pat-ctrl?
Controlling Frequency with Patterns

define function pat-fm(
  durpat, valpat, pitch, dur)
begin
  with hz = step-to-hz(
    pitch + pat-ctrl(durpat, valpat))
  return pwl(0.01, 1, dur - 0.1, 1, dur) * 
    hzosc(hz + 4.0 * hz * hzosc(hz))
end

Using Scores

exec score-play(
  {{ 0 30 {pat-fm-note grain-dur: 8 spread: 1
     pitch: c3 fixed-dur: t
     vel: 50}}
  {10 20 {pat-fm-note grain-dur: 3 spread: 10
     pitch: c4 vel: 75}}
  {15 18 {pat-fm-note grain-dur: 1 :spread: 20
     pitch: c5}}
  {20 13 {pat-fm-note grain-dur: 1 spread: 10
     grain-dur: 20 pitch: c1}}}
Using Scores (2)

```lisp
exec score-play:
  {{ 0 30 {pat-fm-note grain-dur: 8 spread: 1
             pitch: c3 fixed-dur: t
             vel: 50}}
   {10 20 {pat-fm-note grain-dur: 3 spread: 10
            pitch: c4 vel: 75}}
   {15 18 {pat-fm-note grain-dur: 1 spread: 20
            pitch: c5}}
   {20 13 {pat-fm-note grain-dur: 20 pitch: c1}}}
```

• Key ideas:
  • Scores do not have to consist of "notes"
  • Packaging a complex behavior as a Nyquist instrument (a behavior with keyword parameters) supports hierarchical composition
  • Via scores, programs, even score-gen

Using Nyquist SOUNDs for Global Control

• Scores are fine for events
• What about continuous change?
• Example from before: Tendency Masks:
Accessing Sound Values

- Solution: use SOUNDs to specify global continuous evolution of parameter values
- To access a sound: sref(sound, time)
  - sound is any SOUND type
  - time is relative to environment, so time=0 means "now"
- Remember that while behaviors start "now", existing sounds have a definite start time

Template for Global Control using Sounds

define variable pitch-contour =
    pwl(10, 25, 15, 10, 20, 10, 22, 25, 22)
define function get-pitch()
    return sref(pitch-contour, 0)

define function pwl-pat-fm()
    begin
      ...
      make-eval({get-pitch}),
      ...
    end

play pwl-pat-fm()

Note: must be LISP expression
Contours in Score-Gen

begin
with pitch-contour = pwl(10, 25, 15, 10, 20, 10, 22, 25, 22),
ioi-pattern = make-heap({0.2 0.3 0.4})
exec score-gen(save: quote(pwl-score),
   score-dur: 22,
pitch: truncate(c4 +
   sref(pitch-contour,
   sg:start) +
   #if(oddp(sg:count), 0, -5)),
ioi: next(ioi-pattern),
dur: sg:ioi - 0.1,
vel: 100)
end
Examples (code_12.sal)

3D SOUND

Cues
Head-Related Transfer Functions
Room Models
Speaker Arrays
3D Sound

• Various cues to sense direction and distance
  • Inter-aural time delay (direction in horiz plane)
  • Spectral cues (elevation)
  • Reverberation : Direct Sound ratio (distance)
  • Spectral cues (amplitude, distance)
• Computer systems can simulate some of these cues

Head-Related Transfer Functions

• Richard Duda, “3-D Audio for HCI” at
  • http://interface.cipic.ucdavis.edu/CIL_tutorial/3D_home.htm

• Duplex Theory – Lord Rayleigh
  • Interaural Time Difference (ITD)
  • Interaural Level Difference (ILD)
    • ILD is frequency dependent
Cone of Confusion

Source: http://www.owlnet.rice.edu/~psyc351/Images/ConeOfConfusion.jpg

Elevation Cues

- Outer Ear or Pinna
- Pinna reflections create interference that is frequency-dependent
HRTF: Head-Related Transfer Function

- Measure effect of head on sound to each ear as a function of source direction: HRTF
- Apply HRTF to “position” a sound in space
- Interpolate HRTF for sound motion

HRIR – Head Related Impulse Response

- Horizontal Plane
- Median Plane
HRTF, Headphones, and Head Tracking

• Present different signal to each ear → headphones
• Sounds must change when you move your head → Use head tracking to update HRTFs
• Often combined with head-mounted display for VR

Room Models

• Ray tracing
• Audio has long wavelengths
• Early reflection models can convey room geometry
Doppler Shift

- Distance adds delay
- Change in delay creates doppler shift
- Can also simulate doppler shift using frequency modulation, etc.
- What about reflections?

Panning

- Linear panning:
  \[ A_{amp} = \frac{(45 - \theta)}{90} \]
  \[ B_{amp} = 1 - A_{amp}, \quad -45^\circ < \theta < +45^\circ \]

At 0°,
sum of power = \( A^2 + B^2 \)
= \( \frac{1}{4} + \frac{1}{4} = \frac{1}{2} \), but
at 45° or -45°,
sum of power = 0 + 1 = 1

- “Hole in the Middle” effect
Constant Power Panning

\[
A_{amp} = \frac{\sqrt{2}}{2} \times [\cos(\theta) + \sin(\theta)]
\]
\[
B_{amp} = \frac{\sqrt{2}}{2} \times [\cos(\theta) - \sin(\theta)], \quad -45^\circ < \theta < +45^\circ
\]

\[A^2 + B^2 = 1 \text{ for all angles}\]

Multi-speaker playback

- Generally use speaker placement for direction
- Use panning between speakers for direct sound, with Doppler shift
- As distance increases,
  - Intensity decreases
  - Reverb decreases more slowly
- One reverberator feeds all channels because reverb is diffuse, non-directional
Loudspeaker Orchestras

Francois Bayle (seen from behind) seated at the Acousmonium control panel in the Salle Olivier Messiaen, Maison de Radio France, Paris, in 1980. Photo courtesy GRM, from EMF website (http://emfinstitute.emf.org)

Wavefield Synthesis

• Large array of speakers to reproduce the wave front of an imaginary 3D or 2D scene
• Technical University of Berlin: 832 channels, 10cm spacing + woofers at 40cm

http://www.four-audio.de/en/references/wave-field-synthesis.html
Loudspeaker Orchestras

Francois Bayle (seen from behind) seated at the Acousmonium control panel in the Salle Olivier Messiaen, Maison de Radio France, Paris, in 1980. Photo courtesy GRM, from EMF website (http://emfinstitute.emf.org)

Summary

• Multiple cues for sound location, distance
• HRTF’s model direction cues, but require headphones
• Multiple loudspeakers provide multiple point sources, but usually in a plane and panning is a crude approximation of ideal perceptual cues
• Wavefield Synthesis solves many problems, but very expensive even for 2D
• Loudspeaker orchestra “accepts” speakers as legitimate sound sources rather than surrogates