



GRANULAR SYNTHESIS

A versatile synthesis technique

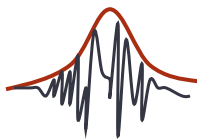


Summary

- Duration mismatch is a common bug in Nyquist programs:
 - Normalize durations to 1 and use stretch (~)
 - Explicit durations everywhere
- Smooth transitions – not just fade-in/fade-out
- Do not neglect control functions or copy over-simplified examples – your goal is expressiveness
- Global control spanning many sounds (notes) add expressiveness on a different time scale

Granular Synthesis

- Combine many “grains” of sound
- Grain is typically taken from a sound file
- Apply smooth envelope to avoid clicks



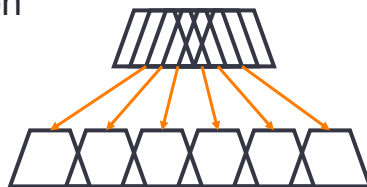
- Grains can overlap

Control

- Too many grains to specify each one
- Stochastic/Statistical control is common
- Dimensions:
 - Where to get grain: smooth progression or random
 - Resample grain? Fixed ratio or random in range.
 - When to play grains? Regular or random.

Things to do with Granular Synthesis

- Texture generation: contains spectrum but loses articulation, rhythm, identity
- Vocal mumbblings: grains can chop up speech to make speech-like nonsense
- Time stretching
- Or compression



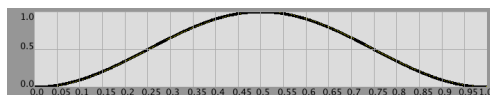
ICM Week 6

Copyright © 2002-2013 by Roger B. Dannenberg

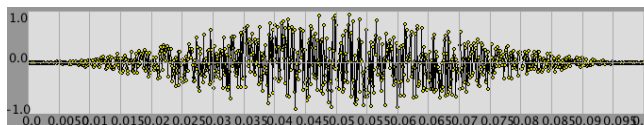
5

Implementation: Construct a Grain

```
function cos-pulse()
  return 0.5 * (1 + hzosc(1 / get-duration(1),
                        *sine-table*, 270.0))
```



```
s-read("filename.wav", time-offset: seconds, dur: d) *
(cos-pulse() ~ d)
```



ICM Week 6

Copyright © 2002-2013 by Roger B. Dannenberg

6