Sound: Perception, Physics, and Processing

N. C. State University

CSC557 ♦ Multimedia Computing and Networking

Fall 2001

Lectures # 06

Sound: Perception, Physics, and Processing

N. C. State University

CSC557 ♦ Multimedia Computing and Networking

Fall 2001

Lectures # 06

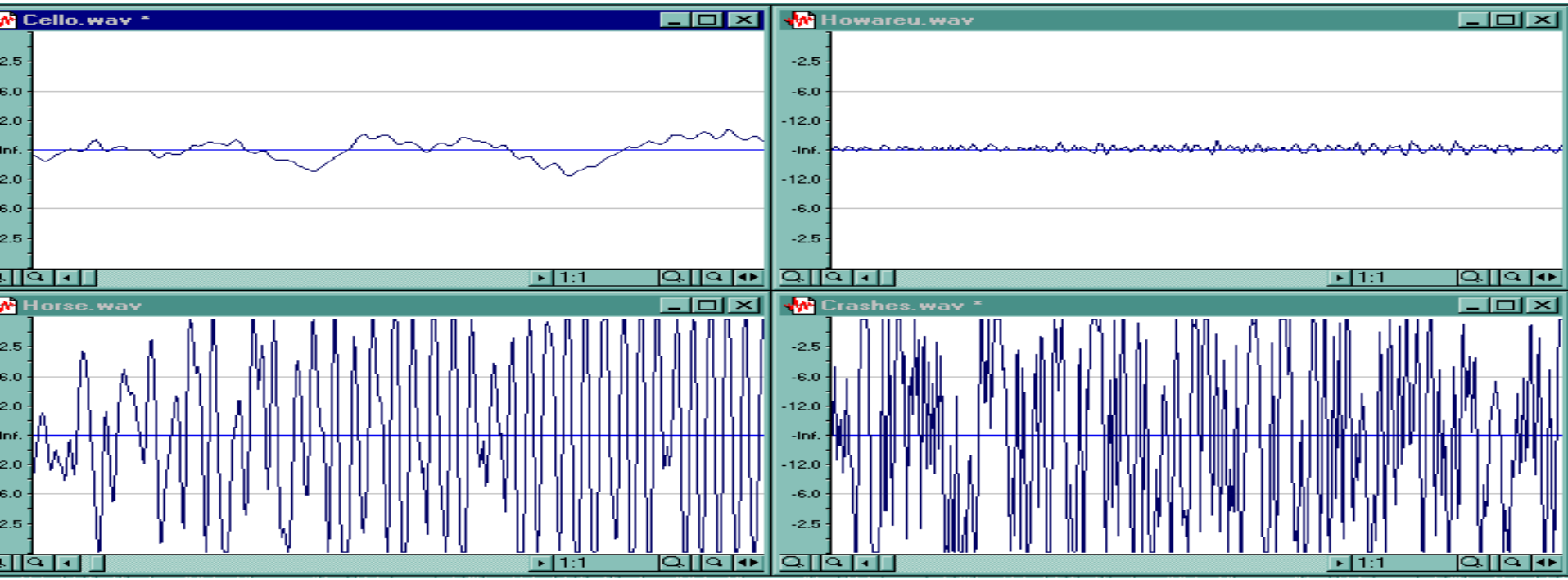
Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Questions / Problems / Announcements?

- ?

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Sounds As Signals



Cello, howareyou, horse, crash.wav

- Can be decomposed into distinct frequencies
  - Using the DFT!

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Physics: Volume

- Sound = pressure waves moving through the air
  - created by mechanical vibration
  - source of vibration: vocal chords, tuning fork, strings of a guitar or piano, diaphragm of a speaker, …
- *Volume* = power of the sound
- A microphone converts pressure variations into voltage variations

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Perception: Loudness

- 10X increase in power ⇔ 2X increase in perceived loudness
- dB measurements universally used
  - Since dB is a ratio, what is being compared to?
- 0dB = arbitrarily defined as the threshold of audibility
- 120dB = power level beyond which hearing damage may occur
- Sound level in a "quiet room" = 40-50 dB

Loudness1..6.wav and loudness70db.wav

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Digital Sound Quality

- How many bits are enough?
  - If SNR = 120dB, won't be able to hear the noise
  - 120dB → $\log_{10}(S/N) = 6$ → $S/N = 10^6$
  - # of bits to represent dynamic range of 10^6 = $\log_2(10^6)$ = 20 bits
- Illustration of digital quality
  - Which is more important: amplitude, or frequency?

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Perception: Pitch

- Pitch = "the" frequency of a note
  - ex: 440 cycles/sec (Hz) = A below middle C
- An "octave" in music = two notes, one of which is twice the frequency of the other
  - ex: 440 Hz and 880 Hz
- The western musical scale has 12 notes, or semitones, per octave
  - ratio of frequencies = 1, 1.06. 1.12, 1.19, 1.26, 1.33, 1.41, 1.50, 1.59, 1.68, 1.78, 1.89, 2
  - (ex: 200 Hz, 212Hz, 224 Hz, 238Hz, …)

Semitones.wav

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Pitch (cont'd)

- People can hear from roughly 20Hz to 20,000Hz
  - and can distinguish a 100th of a semi-tone pitch difference

Freq_10k_12k_14k_16k and 200_100_50_25.wav

Sweep20_280, 300_4k, 4.5k_20K.wav

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Relationship Between Loudness vs. Pitch

- Some pitches seem louder than others, even though they have the same power
  - i.e., our ears are more sensitive to certain frequencies
- A loudness isocontour shows the relative power needed at different frequencies to be perceived as the same loudness

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Loudness Isocontours

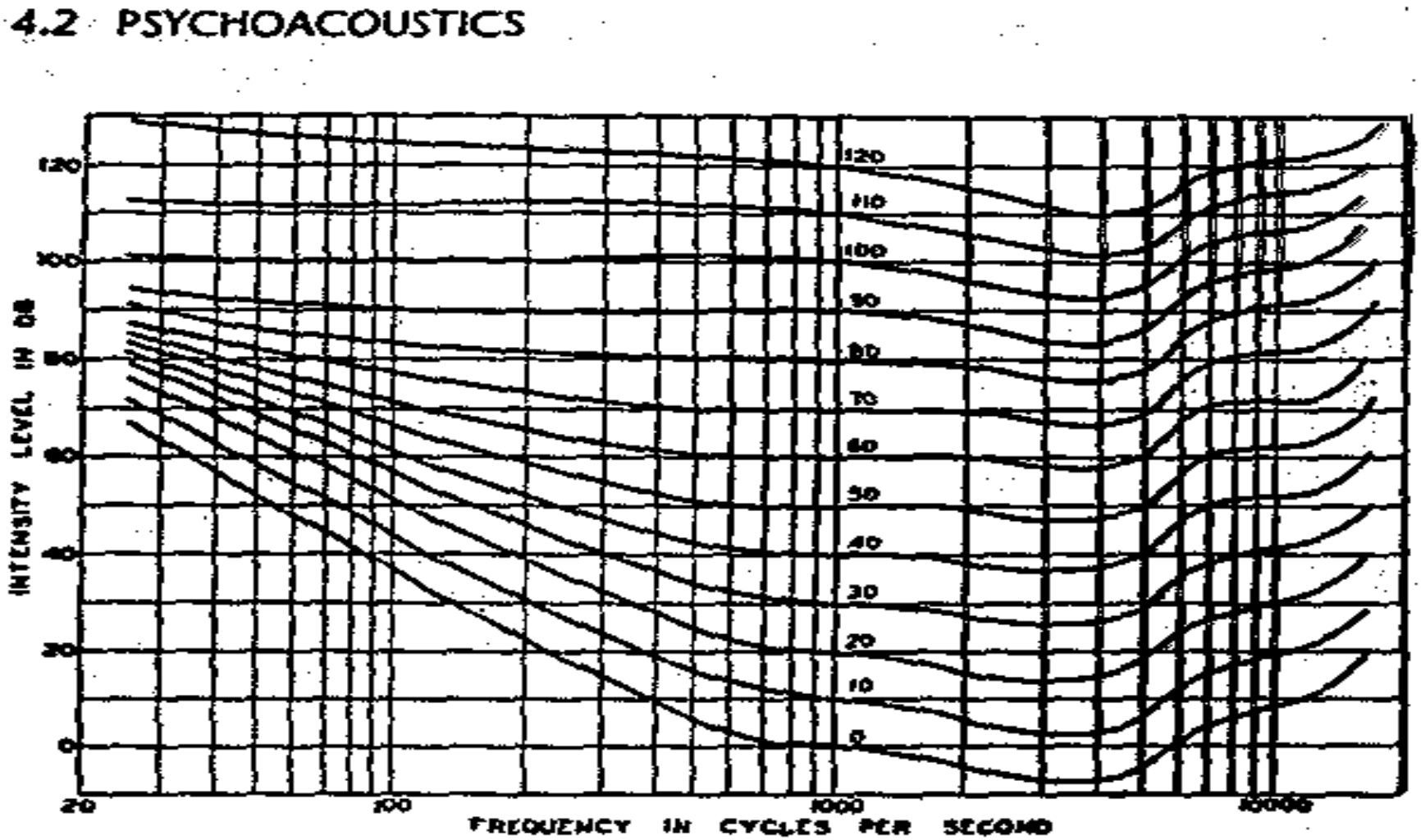


Figure 4.2 Perceived loudness. Each line shows a contour of equal loudness. (From Olson [9], p. 253, with permission)

Loudness_vs_pitch.wav

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Perception: Masking

- A high volume sound can "hide" a lower volume sound so it is not perceived
- The amount of hiding, or masking, is pitch dependent
  - similar pitches are masked more than very different pitches

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Masking (cont'd)

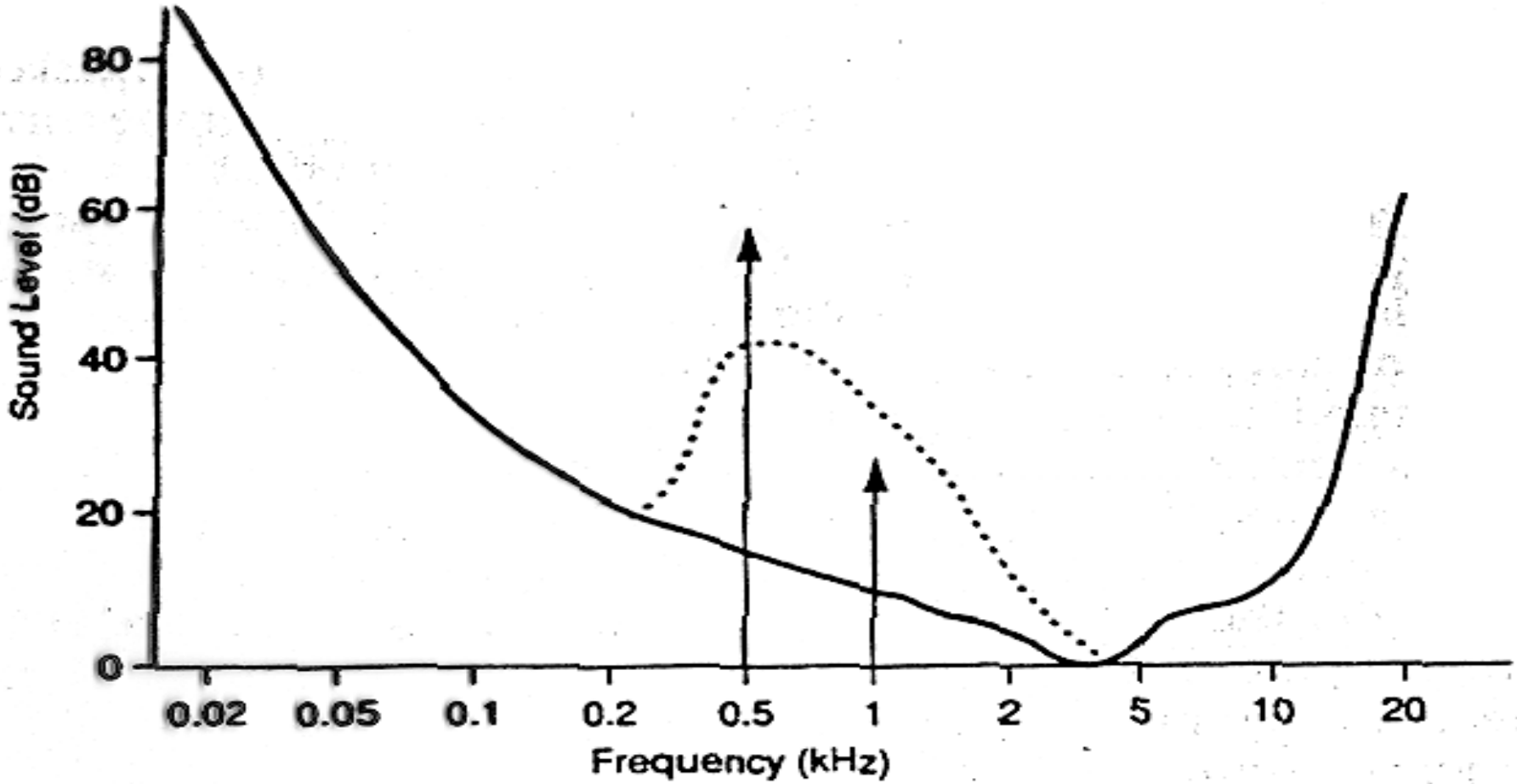


Figure 4.7 The threshold of audibility (solid line) shifted in the presence of a masker (left arrow). (Reproduced with permission from [19], p. 264)

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Perception: Tone

- Tone = characteristic sound of a particular audio source
  - also called "timbre"
- Harmonics = integer multiples of a fundamental frequency
- Tone is produced by spectral content -- what frequencies in what ratio

Timbre1.wav

Synthesis_fm and synthesis_wavetable.wav

Generate_sine, square, triangle.wav

Tone_200, 200_400_600_800.wav

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Perception of Pitch, Again

- Which frequency do we call THE pitch?
  - the fundamental, or lowest frequency, component
- Unpitched sounds don't have frequencies related in a harmonic way

Tone200, 200_400_600_800, 400_600_800.wav

Tone_noise.wav

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

The Effect of Sampling Rate

- What sampling rate is "good enough"?

Sample22, 11, 5, 2_7.wav

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Sound Processing: Notation

- X, Y, and Z are sound files sampled at l_X, l_Y, l_Z samples/sec, respectively
- E is a modulator, or envelope, signal whose values (real numbers) may range from -infinity to +infinity
- s_X, s_Y, s_Z, s_E = # of samples in files X, Y, Z, and E
- x_i, y_i, z_i, e_i are $i+1^{th}$ samples in X, Y, Z, and E
- f_X is a scaling factor (real number)
- m is the maximum possible digital value in PCM scheme for X, Y, and Z

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Notation (cont'd)

- Function `clip`(X,m)

```
for (i = 0..s_x-1)
    if x_i > m
        x_i = m
    else if x_i < -m
        x_i = -m
    endif
endfor
```

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Sample Values

- $x_0..x_4$ = <0, 50, 100, 20, -30>
- $y_0..y_4$ = <-40, -90, 20, 70, 10>
- $e_0..e_4$ = <2, 1, 0, 1, 2>
- m =128

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Editing Volume (Amplitude)

- Volume change
  - given scalar value f_X

for (i = 0..s_x-1)

$z_i = x_i * f_X$

endfor

`clip`(Z,m)

Example: f_x=2.0

$x_0..x_4$ = <0, 50, 100, 20, -30>

$z_0..z_4$ = <0, 100, 128, 40, -60>

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Normalization

- *Procedure*
  1. m_X = maximum absolute value of any x_i in X
  2. compute $f_X = m \ / \ m_X$
  3. then do `volumechange`

Example: $m_x = 100$, $f_x = 128/100$

$x_0..x_4 = \langle 0, 50, 100, 20, -30 \rangle$

$z_0..z_4 = \langle 0, 64, 128, 26, -38 \rangle$

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Fade

- Reduce volume from 100% to 0%

for (i = 0..s_x-1)
    f_i = 1.0 - (i / s_X)
    z_i= x_i * f_i
endfor
/ no clipping necessary */*

Example:

$y_0..y_4$ = <-40, -90, 20, 70, 10>

$z_0..z_4$ = <-40, -68, 10, 18, 0>

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Amplitude Modulate

- Amount of change determined by "envelope" E

for (i = 0..s_x-1)

$z_i = x_i * e_i$

endfor

`clip`(Z,m)

Example:

$y_0..y_4$ = <-40, -90, 20, 70, 10>

$e_0..e_4$ = <2, 1, 0, 1, 2>

$z_0..z_4$ = <-80, -90, 0, 70, 10>

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Noise Gate

- Simplest version: samples with absolute value below a threshold *r* are set to 0
- Improvements
  - only zero out if all values in a small interval less than threshold
  - fade to/from zero over a small interval, rather than replace with zero

```
for (i = 0..sx-1)
    if |xi| < r
        zi = 0
    else
        zi = xi
    endif
endfor
/* no clipping necessary */
```

Example: $r = 60$

$y_0..y_4 = \langle -40, -90, 20, 70, 10 \rangle$

$z_0..z_4 = \langle 0, -90, 0, 70, 0 \rangle$

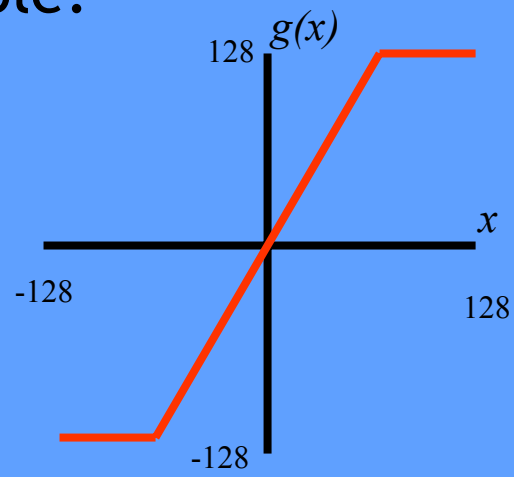
Copyrigl

Companding (Transformations on Amplitude)

- $g(x)$ = function which maps an input amplitude to a new value
- Also called "Dynamics"
- Applications

for (i = 0..s_x-1)
  $z_i = g(x_i)$
endfor
/ no clipping necessary */*

Example:



$x_0..x_4$ = <0, 50, 100, 20, -30>

$z_0..z_4$ = <0, 100, 128, 40, -60>

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Envelope

- A little tricky, but not too hard
  - left as an exercise for the reader

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Simple synthesis

- Given: generating function *g()* that creates samples from the parameters given
- Another exercise for the reader: generating functions for sine wave, square wave, and triangle wave

```
for (i = 0..s_x-1)
    z_i = g(i)
endfor
/* no clipping necessary */
```

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Mixing

- Adding two signals X and Y together, sample by sample, and clipping the result

for (i = 0..s_x-1)

$z_i = x_i + y_i$

endfor

`clip`(Z,m)

Example:

$x_0..x_4$ = <0, 50, 100, 20, -30>

$y_0..y_4$ = <-40, -90, 20, 70, 10>

$z_0..z_4$ = <-40, -40, 120, 90, -20>

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Additive Synthesis

- Procedure
  1. Synthesize two or more signals
  2. Mix

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Cross-Fading

- Smooth transition from one sound to another
- Procedure
  1. Fade X from 100% to 0%
  2. Fade Y from 0% to 100%
  3. Mix X and Y

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Echo

- Given delay *t* (in samples) and *f* (gain, decrease in amplitude)
- Procedure
  1. Generate a delayed, lower volume version of the input sound
  2. Mix this with the input sound
- (Simple) echo is a FIR filter with a single non-zero coefficient
- Multiple-tap echos (reflections)
  - Delays $t_0, t_1, \ldots$ and gains $f_0, f_1, \ldots$

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Echo (cont'd)

for (i = 0..s_x-1)
  if (t>i)
    $z_i = x_i$
  else
    $z_i = x_i + x_{i-t} * f$
endfor
`clip`(Z,m)

Example: t=2, f=.5

$y_0..y_4$ = <-40, -90, 20, 70, 10>

$z_0..z_4$ = <-40, -90, 0, 35, 20>

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Chorus

- Multiple echos, each with
  - very small delays
  - gain approximately = 1
  - pitch (frequency) shift
- Like of chorus of singers

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Reverb

- Mimics the environment → lots of reflecting surfaces
- Echos are themselves echoed → (IIR, or feedback) filter
- Also, surrounding environment filters the signal (absorbs some frequencies, reflects others)
- Result
  - Summation of all these filters

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Equalize

- Same as filtering
- Given cutoff frequencies, design the filter
- See lecture notes, and filtdesn.m

Copyrigl

Pitch Compress/Expand (*With* Duration Change)

- Compress (expand) pitch by factor *f* → same as expand (compress) time by factor *f*
- Method #1
  - just change the time/sample rate field in the file header
  - sound card will play back the file faster or slower!
  - but… the sample rate has changed (cheating?)
- Method #2 (hack)
  - Linearly interpolate the input samples to create new samples

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Pitch Compress/Expand (cont'd)

for ($j = 0 \ldots f*s_x-1$)

$i = j/f$

$n = i - \lfloor i \rfloor$

if $i > s_x-1$

$z_j = x_{sx-1}$

else

$z_j = (1 - n)*x_i + n*x_{i+1}$

endfor

/ no need for clipping */*

Example: $f=2$

$y_0..y_4$ = <-40, -90, 20, 70, 10>

$z_0..z_9$ = <-40, -65, -90, -35, 20, 45, 70, 40, 10, 10>

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Pitch Shift (*No* Duration Change)

- Convert to frequency domain (analyze, DFT)
- Compress or expand in the frequency domain
- Truncate the frequency representation, or pad with zeros, to achieve the desired new number of samples
- Convert back to time domain (synthesize, IDFT)
- The frequency content of sound is constantly changing over time
  - signal must be divided into small consecutive "snippets" of sound
  - process each snippet individually
- Equivalent to Time Compress / Expand (without pitch change)

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Vibrato

- Pitch shift in a time-varying way
- Pitch shift amount is specified by a modulator
- Again, process short segments

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

FM Synthesis

- Like simple synthesis, but the frequency being synthesized is varying over time
  - Variation is specified by another signal, or modulator

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Sources of Info

- [Buford94] *Multimedia Systems*
  - Chapter 4
- [Smith97] *The Scientist and Engineer's Guide to Digital Signal Processing*