Time and Pitch Scaling of Audio Signals

Prof. Dr.-Ing. Jana Dittmann
Andreas Lang

Otto-von-Guericke University of Magdeburg
Multimedia and Security

Thanks to: Stephan M. Bernsee www.dspdimension.com

structure

- MPEG 4 Structured Audio
- Pitch scale, techniques
- Simple Windowing

MPEG-4 Structured Audio

The Problem

- Takes long time to download sound over WWW
- Developed audio compression techniques
- E.g. Real Audio, MP3, Liquid Audio
- MP3 can't be streamed (10-15min download for a 5 min song), but quality is good
- RealAudio can be streamed, but sound like AM radio

Need technology that would allow high-quality sound to be streamed over modem.

Csound

- Special computer invented over last 20 years by Prof. Barry Vercoe
- Description of the "internal work" of a kind of synthesizer
- Use FM synthesis and/or wavetables
- Csound language two parts:
  - Orchestra: describes what all of their instruments should sound like
  - Score: describes how to use the instruments
- Is like MIDI, but orchestra: no analogy in the MIDI world
- MIDI: composer has no direct control over the sound
- Csound: composer specifies exactly what instruments sound like

NetSound

- Spring 1996 Michael Casey → Csound would be good high quality audio on WWW page
- Csound orchestra and score many hundreds of times smaller
- Mike, Adam Landsay, Paris Smaragdis and Eric Scheirer wrote script for Csound and WWW and a client program
- It is called NetSound
- It is not a perfect system: voice, natural instruments, streaming data

MPEG

- Media Lab sponsor Hughes Electronics saw demo of NetSound
- Hughes make a lot of money from MPEG-2 video standard (satellite broadcast Direct TV in USA)
- NetSound wasn't exactly the right solution for MPEG-4
- Decided revisit some of synthesis language issues represented in Csound
- → Structured Audio, NetSound did
- Result language was SAOL, designed winter 1996-1997

- Develop and maintain the SAOL (Structured Audio Orchestra Language) language model and computer programs

SAOL example

```
// This is a simple SAOL instrument that makes a short tone,
// using an oscillator over a stored function table.

instr beep(pitch,amp) {
  table wave(harm,2048,1);  // sinusoidal wave function
  asig sound;               // 'asig' denotes audio signal
  ksig env;                 // 'ksig' denotes control signal

  env = kline(0,0.1,1,dur-0.1,0); // make envelope
  sound = oscil(wave, pitch) * amp * env;
   // create sound by enveloping an oscillator
  output(sound);  // play that sound
}
```


structure

- Pitch scale / time scale
- Prerequisites
- FFT and Frequency resolution
- Windowing

Pitch, time scale

- Pitch Scaling
  - is a way to change the pitch of a signal without changing its length
- Time Scaling
  - is a way to change the lengths of a signal without changing its pitch

Pitch Scale examples

Examples:

	A	B
Original:		
Higher:		
Lower:		

Scale factors:

1.1	1.3
0.9	0.7

Time Scale examples

Examples:

	A	B
Original:		
Faster:		
Slower:		

Scale factors:

1.1	1.4
0.9	0.6

Ways to realize them

- Change the samplerate
- Modify the clock/speed from DSP chip
- Make a pitch/time scale calculation

Pitch/Time Scale - Core

- Fourier Transformation
  - Represents a time domain signal as mixture of sinus and cosine waves (and phases) in frequency domain
- Short Time Fourier Transformation (STFT)
  - Split the signal into smaller “frames”
  - Frames are only several milliseconds long
- Also used often Fast Fourier Transformation (FFT)

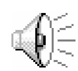
Techniques

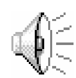
- Fourier Transformation
- Phase Vocoder
- Time Domain Harmonic Scaling (TDHS)

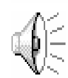
Comparison

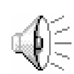
- Speed
  - Real time →TDHS unless SFTS representation of the signal already at hand
  - Different optimization techniques
- Material
  - Knowledge about the signal →optimize the algorithm accordingly
- Quality
  - TDHS gives better results for small time base and pitch changes but will not work well with most polyphonic material
  - Phase vocoder gives smoother results for larger changes and will also work well with polyphonic material but introduces signal smearing with impulse signals

Examples

Original sound wave: 

200% time scaled, Phase Vocoder: 

200% time scaled, TDHS: 

200% time scaled, MCFE: 

TDHS – Time Domain Harmonic Scaling
MCFE – Multiple Component Feature Extraction

Prof. J. Dittmann
A. Lang
WS 03/04

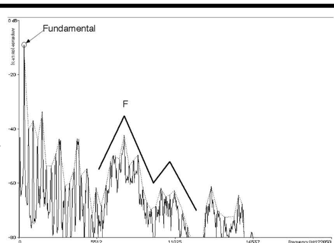
Formants

Definition:

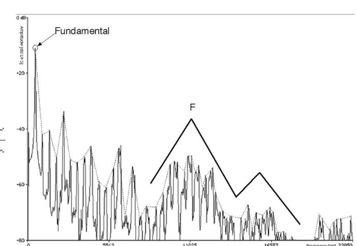
Formants are frequencies remaining more or less the same within a sound, independently from the pitch. They represent significant characteristics of the sound.

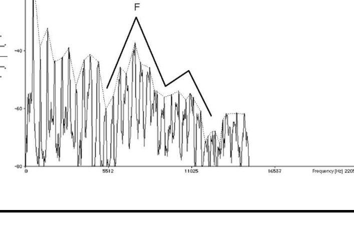
If for e.g. a voice sample is played at increased speed (see sampler next slide) on order to increase the pitch, the formants are scaled as well. The well known "Micky-Maus-Effekt„ which can be observed in this case is mainly due to this formatnt transformation.

Example



Female voice singing vowel “ah”





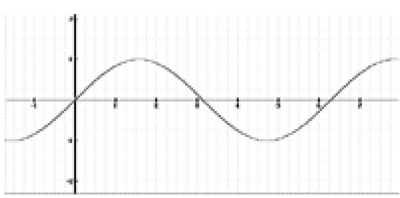
- Fundamental
- F → formants

structure

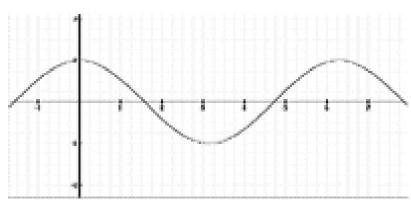
- Pitch scale / time scale
- Prerequisites
- FFT and Frequency resolution
- Windowing

Prof. J. Dittmann
A. Lang
WS 03/04

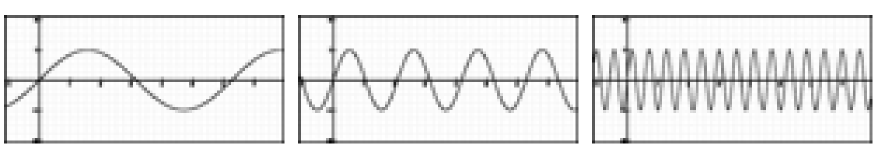
Prerequisites



Sine wave



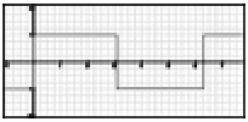
Cosine wave



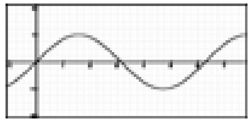
low middle high
frequency

Prof. J. Dittmann
A. Lang
WS 03/04

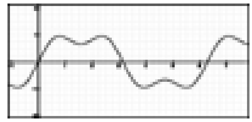
Understanding Fourier Theorem



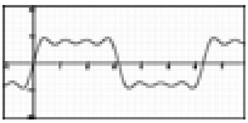
Our signal



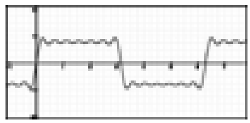
one sine



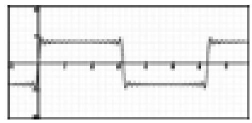
two sine's



four sine's



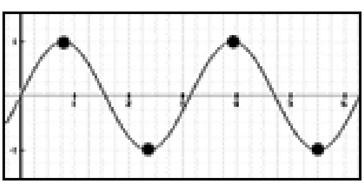
seven sine’s



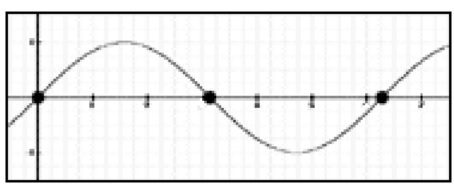
fourteen sine's

Prof. J.
Dittmann
A. Lang
WS 03/04

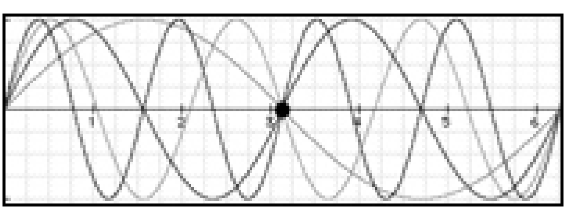
Sine waves in a computer



highest frequency



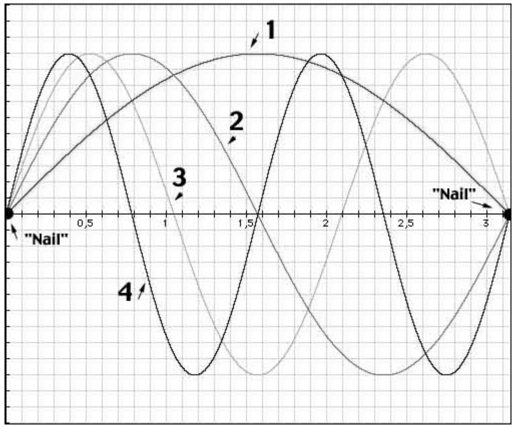
lowest frequency



Many sine waves go through one single point, so one point doesn't tell us about frequency

Prof. J.
Dittmann
A. Lang
WS 03/04

Partial sine waves



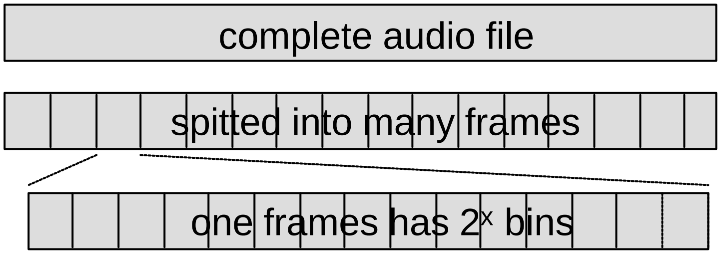
The first 4 partial sine waves

structure

- Pitch scale / time scale
- Prerequisites
- FFT and Frequency resolution
- Windowing

Prof. J. Dittmann
A. Lang
WS 03/04

FFT and Frequency resolution



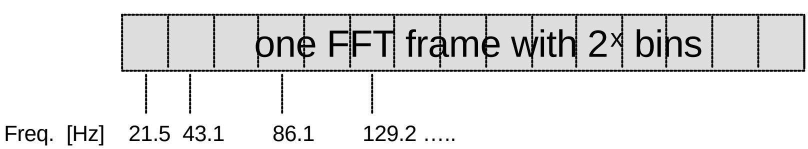
- Frame size is 2^x (typical …,1024,2048,…) to do Fast Fourier Transformation
- Frequency resolution for each bins is:
  SampleRate / FrameSize [Hz]
- Fast Fourier Transformation calculates: frequency and phase

Frequency and bins

Example: Samplerate: 44100 Hz

Framesize: 2048

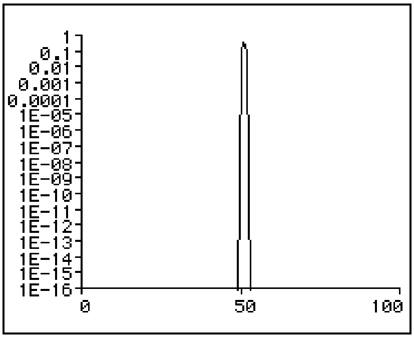
21.533 Hz between two bins!



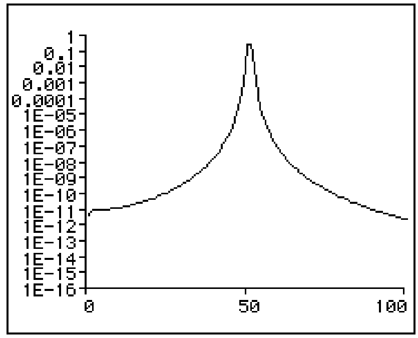
The pros and cons for FFT Framesize:

FFT Framesize	Frequency resolution	Time resolution
High	Good	Bad
Low	Bad	Good

Magnitude spectrum

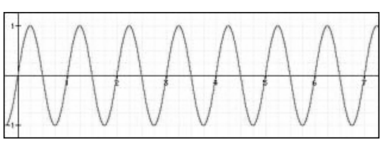


Magnitude spectrum of a sinusoid whose frequency is exactly centered on a bin frequency

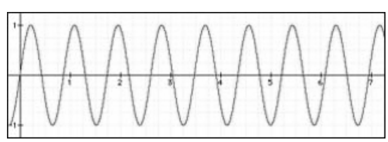


Magnitude spectrum of a sinusoid whose frequency is halfway between two bins

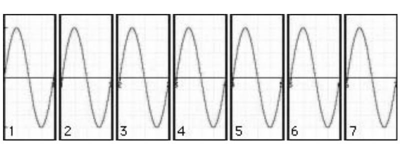
Phase



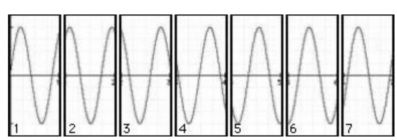
Sine frequency that is exactly that of a bin



Sine frequency that is not on a bin frequency



Now spitted into 7 frames, and each frame will be passed to our transform and be analyzed.



Now spitted into 7 frames, and each frame will be passed to our transform and be analyzed. We see a phase shift in each window.

Prof. J. Dittmann
A. Lang
WS 03/04

Estimated true frequency

Pass #2: 2416.025391 Hz (bin 112.2):

Bin number	Bin frequency [Hz]	Bin magnitude	Bin phase difference	Estimated true frequency [Hz]
110	2368.652344	0.022147	1.256637	2372.958983
111	2390.185547	0.354352	1.256637	2394.492187
112	2411.718750	0.974468	1.256637	2416.025391
113	2433.251953	0.649645	1.256637	2437.558594
114	2454.785156	0.046403	1.256637	2459.091797

Calculating frequency

Pass #1: 2411.718750 Hz (bin 112.0):

Bin number	Bin frequency [Hz]	Bin magnitude	Bin phase difference	Estimated true frequency [Hz]
110	2368.652344	0.000000	-0.403069	2367.270980
111	2390.185547	0.500000	0.000000	2390.185547
112	2411.718750	1.000000	0.000000	2411.718750
113	2433.251953	0.500000	0.000000	2433.251953
114	2454.785156	0.000000	0.112989	2455.172383

$$f = \frac{samplerate}{FFTsize} \bullet \left(BinNr + BinPhase \bullet \frac{1}{2 \bullet p} \right)$$

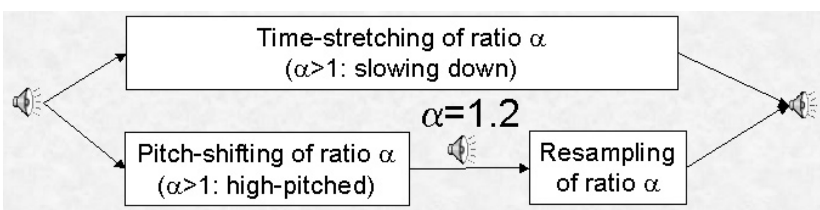
$$f = \frac{44100}{2048} \bullet \left(112 + 1.256637 \bullet \frac{1}{2p} \right) = 2416.02539 \text{ Hz}$$

Scaling the pitch

- Array with magnitude values
- Expect: -2dB 1000Hz pitch scaled by 0.5x
  → -2dB 500Hz
- Array true frequencies of sinusoids
- Interpolation or decimation (up-, down sampling)
- For magnitude: zero order hold interpolation

Prof. J. Dittmann
A. Lang
WS 03/04

Time scale



Original
SR=44.1kHz

Pitch scale a=1.2

SR=32000

- Simplification of time scale
- Do pitch scale and a resampling

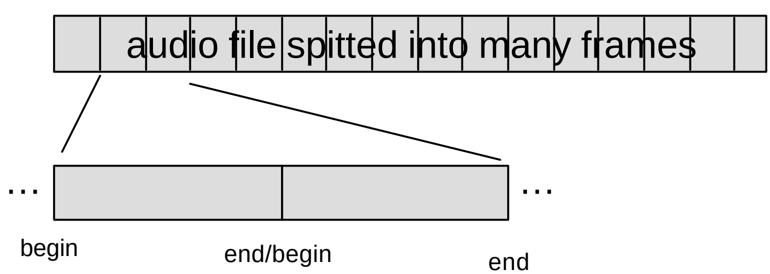
Prof. J. Dittmann
A. Lang
WS 03/04

http://www.iua.upf.es/activitats/semirec/semi-pallone/

structure

- Pitch scale / time scale
- Prerequisites
- FFT and Frequency resolution
- Windowing

Why windowing?



- FFT do not calculate the frequency in the first and last point
- You hear a kind of “klick”

Examples without Windowing

Wave Sound — Original

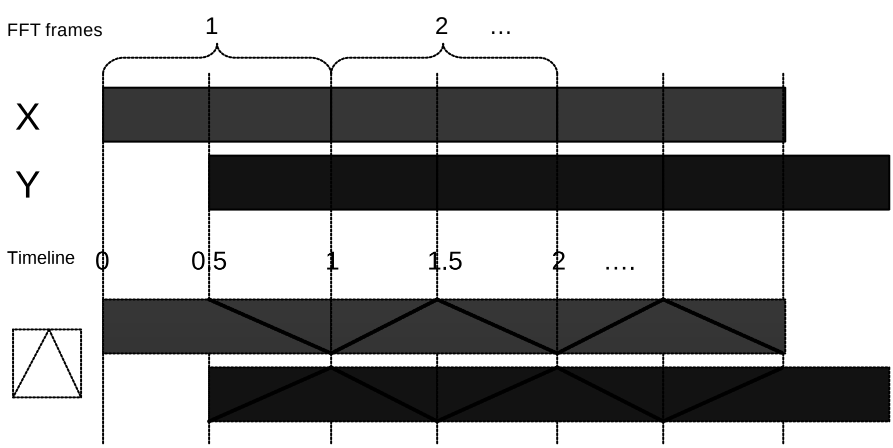
WS 1024

WS 2048

WS 8192

WS 65536

Principles of windowing



Blend over from “red – blue – red …”
Sum is “one” -1

$$\sum_{i=1}^{0} i * X_i + (1-i) * Y_i$$

Prof. J. Dittmann
A. Lang
WS 03/04

Windowing

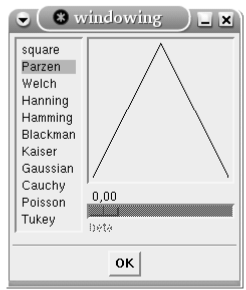
- Different types of windowing are possible:
  - Square
  - Parzen
  - Welch
  - Hanning
  - Hamming
  - Blackman
  - Kaiser
  - Gaussian
  - Cauchy
  - Poisson
  - Turkey

Windowing

- Different types of windowing are possible:
  - Square
  - Parzen
  - Welch
  - Hanning
  - Hamming
  - Blackman
  - Kaiser
  - Gaussian
  - Cauchy
  - Poisson
  - Turkey

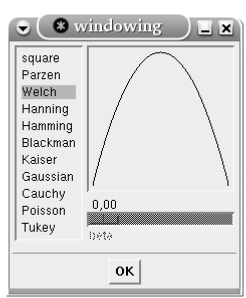
Windowing

- Different types of windowing are possible:
  - Square
  - Parzen
  - Welch
  - Hanning
  - Hamming
  - Blackman
  - Kaiser
  - Gaussian
  - Cauchy
  - Poisson
  - Turkey



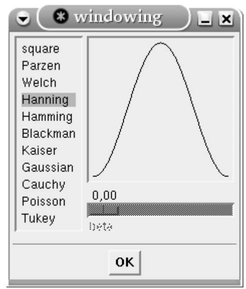
Windowing

- Different types of windowing are possible:
  - Square
  - Parzen
  - Welch
  - Hanning
  - Hamming
  - Blackman
  - Kaiser
  - Gaussian
  - Cauchy
  - Poisson
  - Turkey



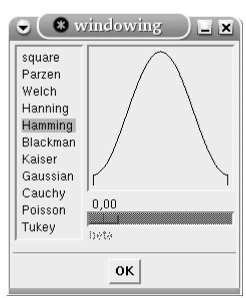
Windowing

- Different types of windowing are possible:
  - Square
  - Parzen
  - Welch
  - Hanning
  - Hamming
  - Blackman
  - Kaiser
  - Gaussian
  - Cauchy
  - Poisson
  - Turkey



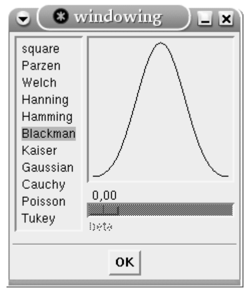
Windowing

- Different types of windowing are possible:
  - Square
  - Parzen
  - Welch
  - Hanning
  - Hamming
  - Blackman
  - Kaiser
  - Gaussian
  - Cauchy
  - Poisson
  - Turkey



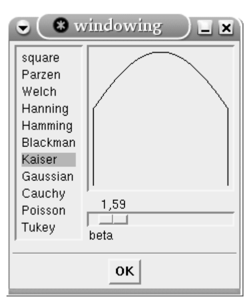
Windowing

- Different types of windowing are possible:
  - Square
  - Parzen
  - Welch
  - Hanning
  - Hamming
  - Blackman
  - Kaiser
  - Gaussian
  - Cauchy
  - Poisson
  - Turkey



Windowing

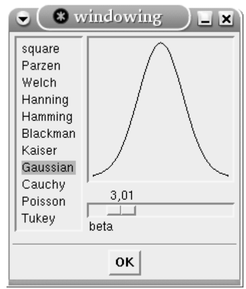
- Different types of windowing are possible:
  - Square
  - Parzen
  - Welch
  - Hanning
  - Hamming
  - Blackman
  - Kaiser
  - Gaussian
  - Cauchy
  - Poisson
  - Turkey



Prof. J. Dittmann
A. Lang
WS 03/04

Windowing

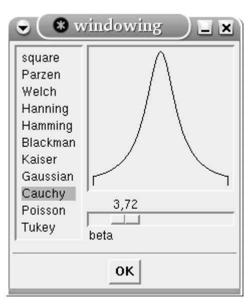
- Different types of windowing are possible:
  - Square
  - Parzen
  - Welch
  - Hanning
  - Hamming
  - Blackman
  - Kaiser
  - Gaussian
  - Cauchy
  - Poisson
  - Turkey



Prof. J. Dittmann
A. Lang
WS 03/04

Windowing

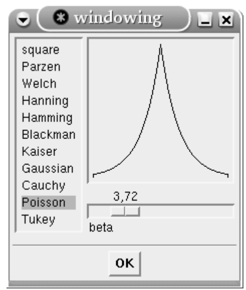
- Different types of windowing are possible:
  - Square
  - Parzen
  - Welch
  - Hanning
  - Hamming
  - Blackman
  - Kaiser
  - Gaussian
  - Cauchy
  - Poisson
  - Turkey



Prof. J. Dittmann
A. Lang
WS 03/04

Windowing

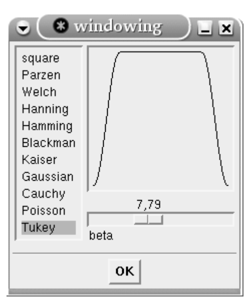
- Different types of windowing are possible:
  - Square
  - Parzen
  - Welch
  - Hanning
  - Hamming
  - Blackman
  - Kaiser
  - Gaussian
  - Cauchy
  - Poisson
  - Turkey



Prof. J. Dittmann
A. Lang
WS 03/04

Windowing

- Different types of windowing are possible:
  - Square
  - Parzen
  - Welch
  - Hanning
  - Hamming
  - Blackman
  - Kaiser
  - Gaussian
  - Cauchy
  - Poisson
  - Turkey



Prof. J. Dittmann
A. Lang
WS 03/04

Thank you!