Äänen spektrianalyysi graffaefektinä

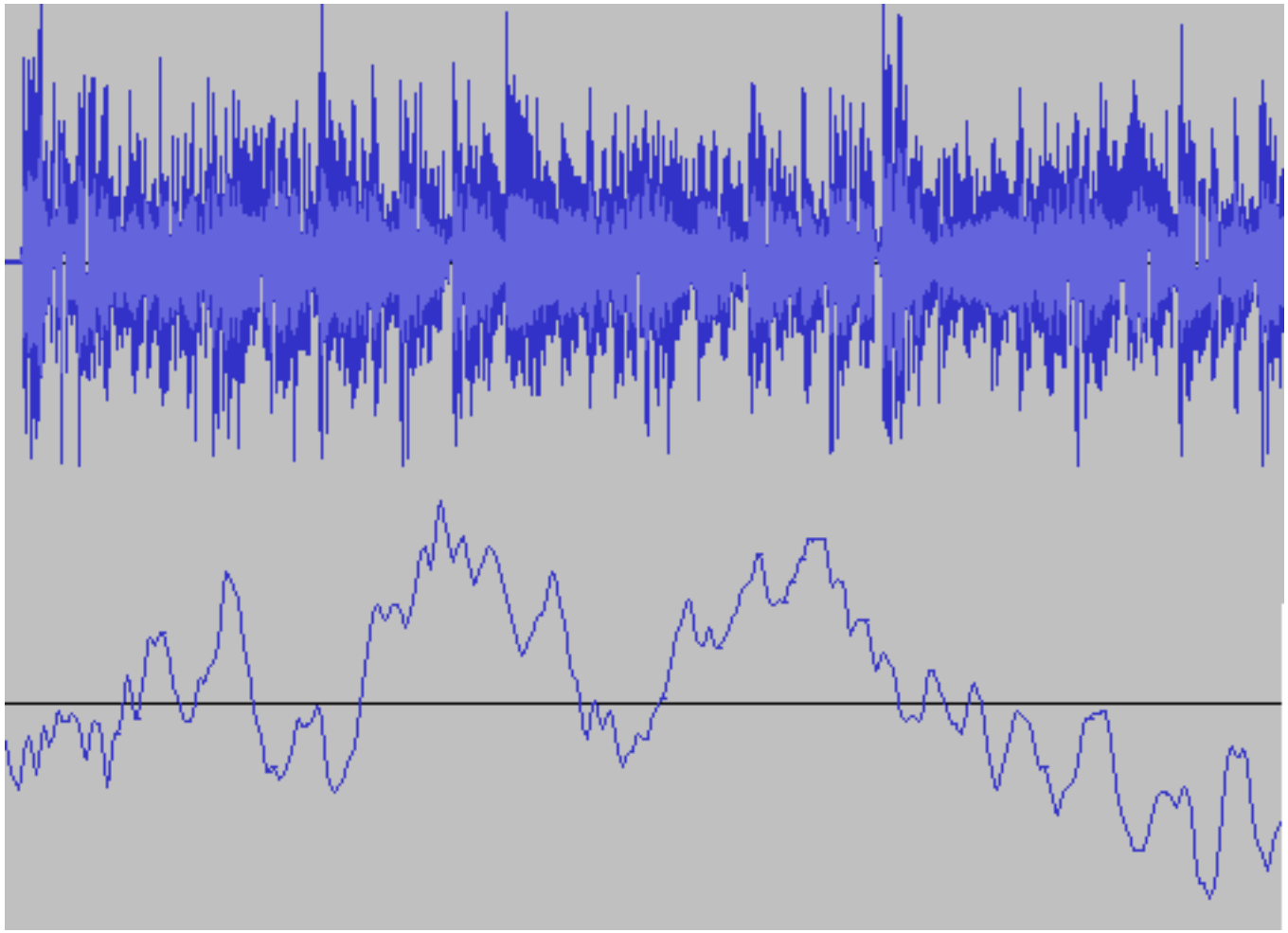
Konsta “sooda” Hölttä

14.6.2014

Spektrianalyysillä synkkaus

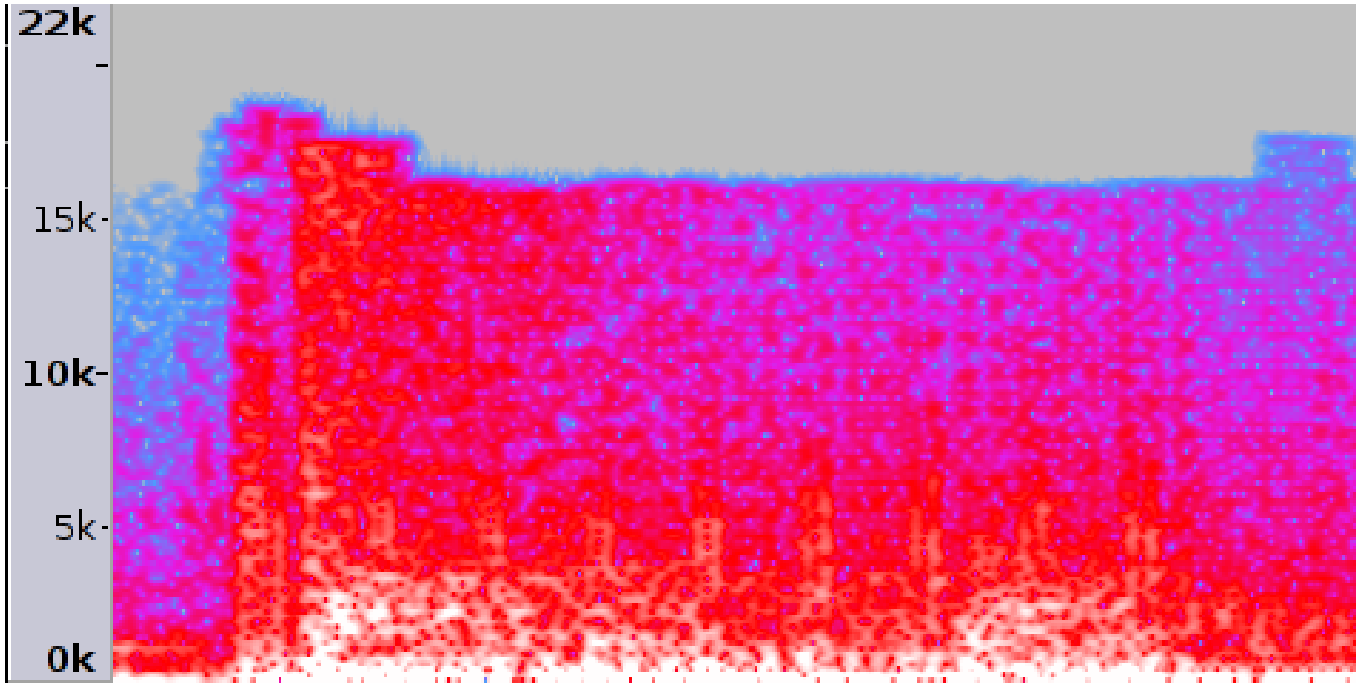
- Ääni = näytejono
- Spektri = taajuudet
- Synkkaus = synkronointi = efektit musan tahtiin
- Synkki implisiittisesti, soivan äänen perusteella
- Ei käsin säädettyjä ajanhetkiä

Ääni



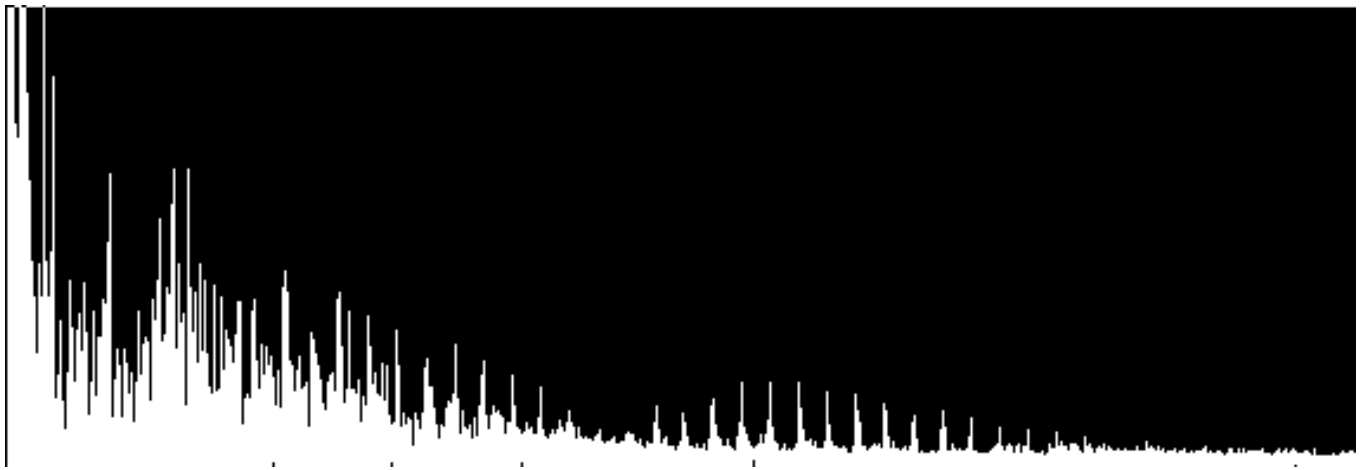
- Äänenpaineen voimakkuus ajan funktiona

Spektri



- Näytejonon sisältämät taajuudet
- Määritelty jollekin ajanpätkälle, ei hetkelle
- Vaihe ja amplitudi
- Lasketaan “ajan funktiona” liukuvasti

Hetkellisesti



- Äänipätkä voidaan hajottaa sineiksi ja kosineiksi
- Muunnos yhdestä signaalijonosta
- Amplitudi taajuuden funktiona

Synkkaus

- Efekti viululle/rummulle/huilulle/whatever
- Spektriä luetaan reaaliajassa musan kohdalta
- Huomataan spektristä haluttu taajuus
- Taajuusalueen voimakkuudesta efektin voimakkuus

Ajasta taajuusavaruuteen

- Syöte: pätkä ääntä (tai mitä tahansa signaalia)
- (???)
- Profit: jokaisen mahdollisen taajuuden amplitudi
- Reaaliselle inputille puolet samplenäytteen pituudesta

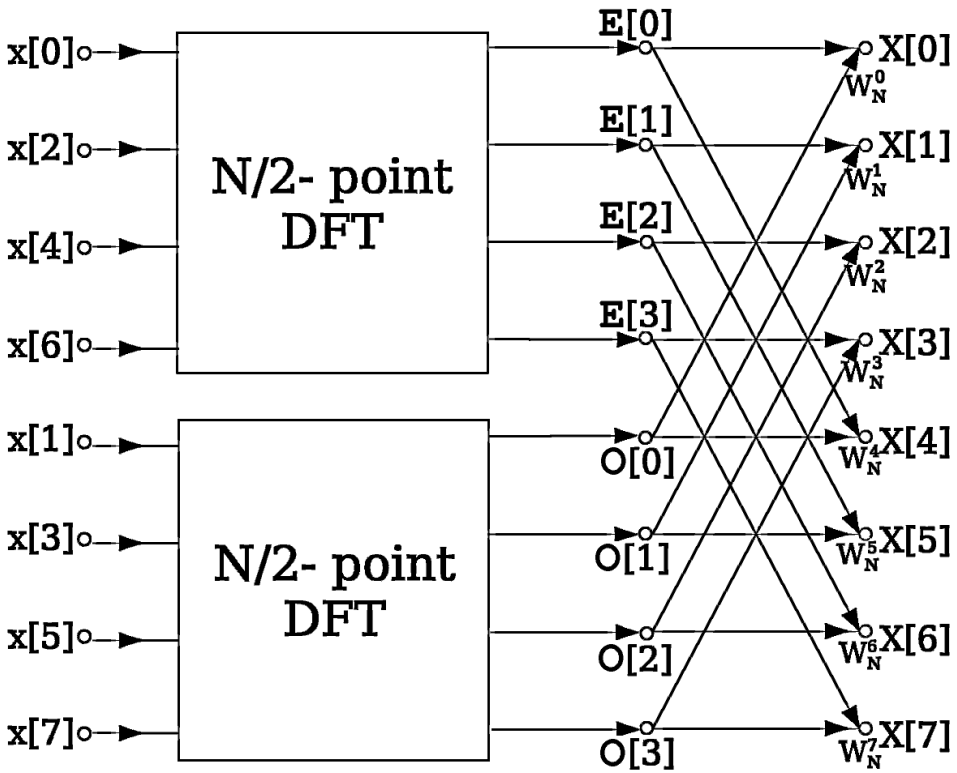
Fourier-muunnos

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\ e^{-2\pi i x \xi}\,dx \tag{1}$$

$$X_k = \sum_{n=0}^{N-1} x_n \cdot \omega^{kn} \tag{2}$$

- Monta tapaa ajatella teknisesti
- Eriparametristen sinien ja kosinien kombo
- (giffi wikipediassa) http://tinyurl.com/fourierlove
- Jatkuva / diskreetti

Fast fourier transform

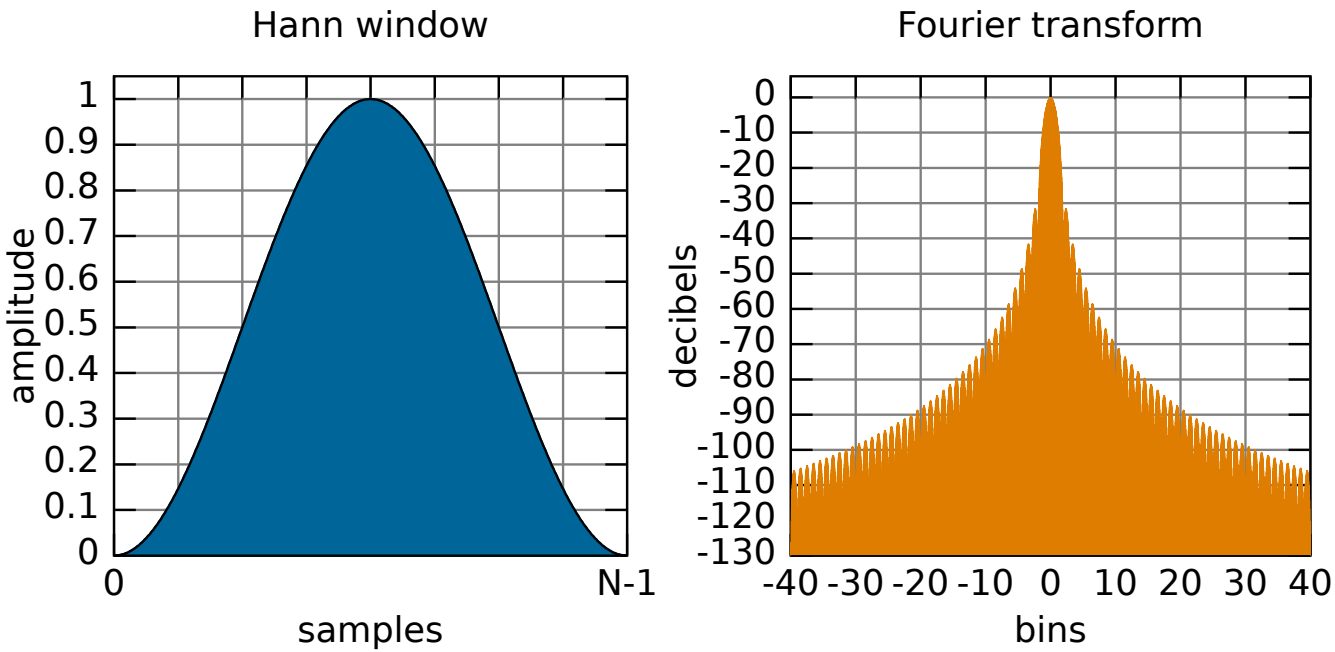


- mm. Cooley-Tukey
- Muitakin menetelmiä
- O(simosti) $\rightarrow$ O(n log n)

Ikkunointi

- Jatkuva muunnos on äärettömille signaaleille
- Diskreettejä näytteitä on äärellinen määrä
- Useimmiten aallonpituus ei mene tasan näytepituuden kanssa
- Reunojen epäjatkuvuus häiritsee
- Spectral leakage

Ikkunointi



- Kerrotaan näyteikkunan näytteet ikkunafunktion arvoilla
- Vaimennusta reunimmaisiin
- Vähän parempi vaste
- Ikkunafunktioita useita

Demossa

- Bassorumpu = matalia taajuuksia
- Huilut, viulut yms = korkeampia
- Lautasrummut pitkälti highpass-kohinaa
- Puhtaampi ääni helpompi havaita
- Dubsteppiin turhahkoa

Synkkaus

- Ajetaan FFT jollekin lyhyelle pätkälle
- Äänisamplet biisin soivasta kohdasta
- Spektristä jollekin osa-alueelle kynnysarvo
- Tai piikkien tunnistusta ajasta
- Jos aallon amplitudi riittävän suuri, tehdään jotain

Processing

- Processingin äänikirjasto Minim osaa FFT:n
- `FFT fft = new FFT(buffersize, samplerate);`
- `fft.forward(buf);`, `a = fft.getBand(i);`
- Taajuudet nollasta samplerate/2 asti
- Bufferikoko määrää taajuusresoluution
- Isompi bufferi, parempi taajuusresoluutio
- Isompi bufferi, huonompi aikaresoluutio

Demoilua

- “Pohjakoodi” testailuun
- Luodaan musaplayeri soittamaan musat
- Musat uudestaan muistiin fft:tä varten
- Myös BeatDetect-testailua
- Jaossa muiden graffathon-pakettien joukossa