

Äänen taajuuden seuraus

Toteutusdokumentti

Roope Salmi
Tiralabra, 4. periodi 2021

Tähän mennessä on toteutettu FFT, ristikorrelaatio, ja korrelaatiotäsmäysalgoritmi. Demo-ohjelmassa esitetään, kuinka tätä voidaan käyttää oskilloskoopin kuvaajan vakauttamiseen.

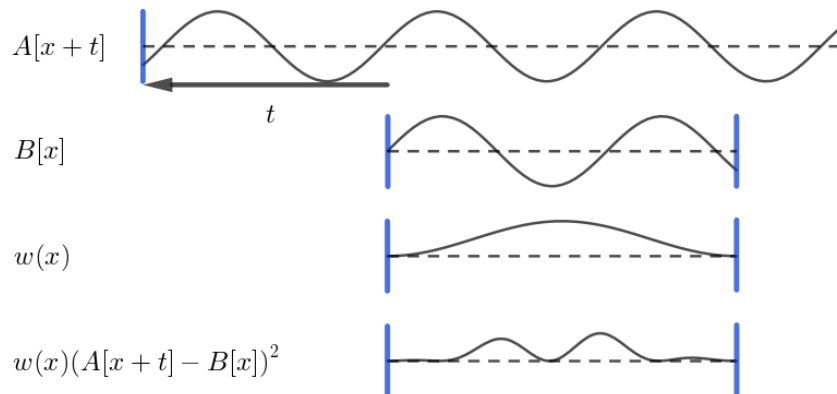
Korrelaatiotäsmäys

Algoritmi, joka etsii pidemmästä äänenpätkästä A sen kohdan, jossa lyhyempi äänenpätkä B esiintyy kaikista lähimpänä. Toteutus löytyy tiedostosta `src/correlation_match.rs`.

Olkoon signaalit $A[0..n]$ ja $B[0..m]$, $n \geq m$. Algoritmi etsii sellaisen aikasiirroksen t , jolla summa

$$d(t) = \sum_{x=0}^{m-1} w(x)(A[x+t] - B[x])^2$$

on minimaalinen. Etäisyys määräytyy siis erotuksien neliöiden kautta. Tässä w on paino-funktio, jonka avulla voidaan painottaa enemmän esimerkiksi keskikohtia kuin reunoja.



Kuva 1: Korrelaatiotäsmäys

Jos tämä summa esitetään muodossa

$$\sum_{x=0}^{m-1} w(x)A[x+t]^2 - 2w(x)B[x]A[x+t] + w(x)B[x]^2,$$

nähdään, että se voidaan laskea kahtena ristikorrelaationa ja yhtenä suorana tulona.

Ristikorrelaatiolla tarkoitetaan tässä siis operaatiota kahden funktion välillä, jonka tuloksesta on funktio $f * g$:

$$(f * g)[t] = \sum_{x=0}^N f[x]g[x+t]$$

Koska käsitellään diskreettejä näytteistettyjä signaaleja, ei funktion $d(t)$ minimikohtaa voida määrittää näytteenottoväliä tarkemmin. Minimikohdan voisi pyrkiä määrittämään tarkemmin parabolisella interpolaatiolla, mutta tätä ei ole vielä toteutettu.

Metodi on pitkälti sama, kuin YIN-algoritmi [1], mutta absoluuttisen taajuuden tunnistamisen sijaan täsmätään erillisiä signaaleja keskenään.

Oskilloskoopin vakautus

Korrelaatiotäsmäysalgoritmia hyödynnetään oskilloskoopin vakauttamiseen seuraavasti:

Signaaliksi A asetetaan uutta luettua signaalia. Signaaliksi B taas asetetaan vanhaa, aiemmin näytettyä signaalia. Ideana on, että algoritmi löytää signaalista A sopivan aikasiiroksen, jotta seuraavaksi näytettävä kuvaaja on mahdollisimman lähellä edellistä.

Jos oletetaan, että saapuva signaali on syklinen, niin taulukon A koko on valittava siten, että siihen mahtuu vähintään kaksi jaksoa. Tällöin uudesta signaalista löytyy aina sopiva kohta, joka vastaa aiemmin näytettyä signaalia täsmällisesti. Taulukon B koko taas on oltava korkeintaan puolet tästä.

Painofunktiona w käytetään jotakin ”kellokäyrän” tyyppistä funktiota, eli painotetaan näytön keskikohtia. Näin signaali pysyy keskitettynä, vaikka perustaaajuus muuttuu. Demo käyttää Hann-ikkunafunktiota [2].

Demo on suunniteltu toimimaan 44,1kHz näytteenottotaajuudella. Taulukon A koko on 1470 näytettä, ja taulukon B koko on 720 näytettä. Tällöin uusi kokonainen taulukko A luetaan reaaliajassa noin 30 kertaa sekunnissa. Matalin seurattava perustaaajuus on siis 60Hz.

Demossa on lisäksi säädettävä vaimenemiskerroin ja erillinen muisti. Uutta ääntä ei veratakaan juuri näytettyyn aaltomuotoon, vaan erilliseen muistiin, jota päivitetään eksponentiaalisella vaimenemisella. Tämän tarkoituksena on saada näkymä pysymään vakaana, vaikka ääneessä on väliaikaisia äkillisiä muutoksia.

$$muisti[x] := \alpha A[x+t] + (1 - \alpha)muisti[x]$$

FFT

FFT eli nopea Fourier-muunnos on toteutettu 2-kantaisena Cooley-Tukey -menetelmänä. Toteutus on tiedostossa `src/fft.rs`.

`fft::Prepared`-olioon esilasketaan tietylle muunnoksen koolle N twiddle-kertoimet, eli kaikki kompleksiluvut muotoa $e^{2\pi ix/N}$, joita algoritmin suorituksen aikana käytetään. Esilaskeminen on projektin käyttötarkoituksessa hyödyllistä, koska on tarpeen tehdä jatkuvasti samankokoisia muunnoksia. Testausdokumentista arvioidaan tämän optimoinnin hyötyä.

Kirjallisuutta

- [1] Alain de Cheveigné, Hideki Kawahara: "YIN, a fundamental frequency estimator for speech and music", The Journal of the Acoustical Society of America 111, 1917-1930 (2002) <https://doi.org/10.1121/1.1458024>
- [2] "Hann function", Wikipedia (2021), luettu 18.4. https://en.wikipedia.org/wiki/Hann_function