

SiTraNoSTAR: TIETOKONESOVELLUS ÄÄNEN TOISTONOPEUDEN MUOKKAAMISEEN KOLMITIEHAJOTELMAN AVULLA

Roope Salmi, Vesa Välimäki

Aalto-yliopiston sähkötekniikan korkeakoulu
Akustiikan laboratorio, Informaatio- ja tietoliikennetekniikan laitos
Otakaari 5, 02150 Espoo
rpsalmi@gmail.com, vesa.valimaki@aalto.fi

Tiivistelmä

Äänen toistonopeuden muokkaamista tarvitaan muun muassa musiikki- ja elokuvaluotannossa, videoneuvotteluissa sekä videon hidastuksissa. Se on kuitenkin haastavaa tehdä hyvälaatuisesti muuttamatta samalla äänen sävelkorkeutta ja sävyä. Esittelemme uuden version SiTraNo-sovelluksesta (SiTraNo*), joka kokoaa aihealueen viimeisintä tieteellistä tutkimusta käytännön työkaluun, jossa toistonopeutta voi säätää reaaliajassa. Menetelmä perustuu äänisignaalin kolmitiehajotelmaan sini-, transientti- ja kohinakomponentteihin, joka tehdään lyhytaikaisen Fouriermuunnoksen avulla. Äänen toistonopeutta voidaan vaihtaa hyvällä äänenlaadulla, kun nämä komponentit muokataan kullekin erikseen suunnitellulla algoritmilla. Aiempiin versioihin nähden SiTraNo* sisältää nopeamman algoritmin äänen eroteluun, uuteen tutkimukseen perustuvan kohinanmuotoilumenetelmän sekä muita parannuksia äänenlaatuun ja käyttöliittymään.

1 JOHDANTO

Tässä artikkelissa esitellään uusin versio tietokonesovelluksesta, jolla voidaan hidastaa tai nopeuttaa äänitettä ilman että äänenkorkeus muuttuu. Sovellus myös havainnollistaa, miten mikä tahansa ääni voidaan hajottaa kolmeen komponenttiin: sineihin, transientteihin ja kohinaan.

Ei ole ilmiselvää, miltä äänitteen tulisi kuulostaa, kun sen nopeutta muutetaan merkittävästi. Jos esimerkiksi transientteja, kuten sävelten alkuja tai lyömäsoittimen iskuääniä venytetään, ne eivät enää kuulosta luonnollisilta. Tästä syystä on kehitetty menetelmiä, jotka pyrkivät säilyttämään transientit ennallaan [1, 2]. Algoritmit, joita käytetään jäljelle jäävän osan nopeuden muuttamiseen tyypillisesti olettavat, että signaali koostuu rajallisesta määrästä sinikomponentteja. Tämä oletus ei kuitenkaan päde kohinalle, jota esiintyy osana puhetta, kaikkua ja useita luonnon ääniä. Parempi äänenlaatu saavutetaan kolmitiehajotelman avulla, jossa transienttien lisäksi erotellaan tonaalinen ääni ja kohina [3, 4, 2]. Moliner et al. [5] esittivät menetelmän varta vasten kohinaisen komponentin käsittelyyn. Sen avulla merkittävä, esimerkiksi nelinkertainen hidastus, onnistuu laadukkaasti.

Copyright ©2025 Roope Salmi ja Vesa Välimäki. Tämä on avoimesti julkaistu teos, joka noudattaa Creative Commons NIMEÄ 4.0 Kansainvälinen –lisenssiä (CC BY 4.0). Teosta saa kopioida, levittää, näyttää ja esittää julkisesti ja siitä saa luoda johdannaisteoksia, kunhan tekijän nimi ja lähde mainitaan asianmukaisesti.

SiTraNo* on uusi paranneltu versio sovelluksesta, jota edeltävät alkuperäinen SiTraNo [6] ja SiTraNo+ [7]. Sen nimi tulee sanoista *Sines*, *Transients* ja *Noise* eli suomeksi sinit, transientit ja kohina. Sovellus on suunniteltu opetuskäyttöön, sillä se sisältää erilaisia valintoja ja havainnollistuksia, jotka auttavat ymmärtämään menetelmien toimintaa. Komponenttien äänenvoimakkuutta voidaan säätää erikseen ja toistoalgoritmit transienteille ja kohinalle voidaan asettaa erikseen päälle tai pois.

Esittelemämme uusi versio on jatkokehitetty SiTraNo+-sovelluksen koodista ja se on toteutettu C++-kielellä JUCE-ohjelmistokehyksessä. SiTraNo*-sovelluksessa käyttäjä voi reaaliaikaisen kuuntelun lisäksi tallentaa äänitiedoston aikaskaalattuna valituilla asetuksilla. Havainnollistusta varten sovelluksessa on kolmivärinen spektrogrammi, jossa sini-, transientti ja kohinakomponentit erottuvat eri värisinä. Lähdekoodi ja valmiit ohjelmabinäärit ovat vapaasti saatavilla verkosta. [1]

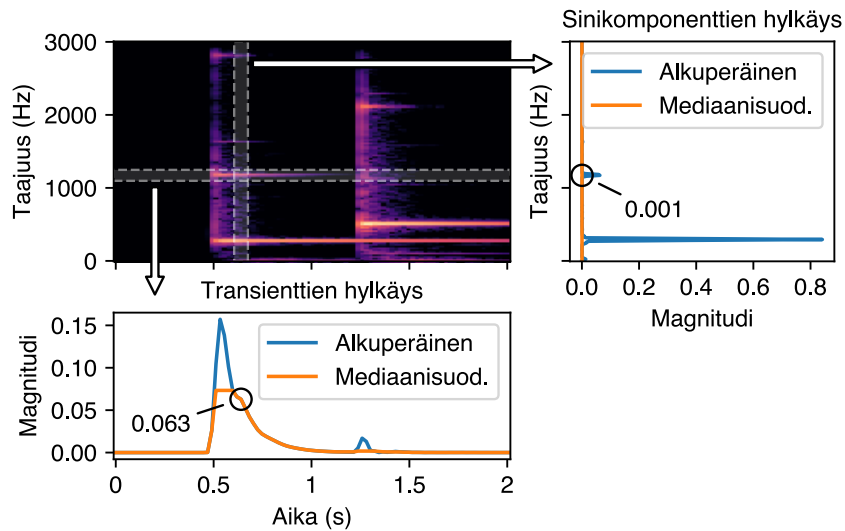
2 KOLMITIEHAJOTELMA

Hajotelma sineihin, transientteihin ja kohinaan (engl. STN decomposition) voidaan toteuttaa spektrogrammin eli lyhytaikaisen Fourier-muunnoksen (engl. *short-time Fourier transform*, *STFT*) avulla. Spektrogrammissa aika etenee X-akselilla ja taajuus Y-akselilla, ks. kuva [1]. Transientit näkyvät spektrogrammissa pystysuuntaisina viivoina, sillä ne ovat lyhytkestoisia ja laajakaistaisia [8]. Sinit puolestaan ilmenevät vaakasuuntaisina viivoina, koska ne ovat kapeakaistaisia ja soivat pitempään kuin transientit. Kohinalla ei ole selkeää viivamaista rakennetta [3].

FitzGerald [8] ehdotti mediaanisuuodattimien käyttöä määrittämään, kuinka vahvasti aika-taajuuspisteet ovat transienttimaisia tai tonaalisia. Mediaanisuuodatuksen ideana on muokata jokainen spektrogrammin rivi ja sarake erikseen liukuvalla ikkunalla, jonka keskikohdassa oleva arvo korvataan lukujen mediaanilla. Kuvassa [1] on esimerkki mediaanisuuodattimien toiminnasta. Rivisuuntainen mediaanisuuodatin säilyttää vaakasuuntaiset viivat (sinit) ja sarakesuuntainen säilyttää pystysuuntaiset viivat (transientit). Mediaanisuuodattimet kuitenkin hylkäävät äkilliset piikit, jotka ristikkäiseen suuntaan kulkeva viiva aiheuttaa, mikä havainnollistetaan kuvassa [1] esimerkkien avulla. Käyttämämme vaakasuuntaisen mediaanisuuodattimen pituus on 200 ms ja pystysuuntaisen 500 Hz [3].

Vaaka- ja pystysuuntaisten mediaanisuuodatettujen arvojen suhdetta kutsutaan spektripisteen tonaalisuudeksi (engl. *tonalness*) [3]. Suhteen perusteella lasketaan STFT:n peitteet sineille, transienteille ja kohinalle [4]. Kukin peite (engl. *mask*) kerrotaan STFT:n kompleksiarvolla jokaisessa pisteessä, minkä jälkeen kullekin komponentille tehdään käänteismuunnos. Peitteet ovat osin päällekkäisiä, kuitenkin niin, että niiden summa on 1 joka pisteessä. Hyvän äänenlaadun varmistamiseksi SiTraNo* toteuttaa hajotelman kahdessa vaiheessa. Ensin sinit erotellaan käyttäen STFT:tä pitkähköllä Hann-ikkunalla, jonka pituus on 8192 näytettä eli 185 ms, kun näytetaajuus on 44,1 kHz. Tällöin mediaanisuuodattimien pituudet ovat 9 (vaaka) ja 93 (pysty) pistettä. Jäännössiinaalista erotellaan transientit käyttäen lyhyempää ikkunaa (512 näytettä eli 11 ms), jolloin mediaanisuuodattimien pituudet ovat 138 (vaaka) ja 6 (pysty) pistettä. Toisen erottelukierroksen jälkeen jäljelle jää kohina. Spektrierottelun parametrit on valittu pseudo-optimaalisesti Fierron ja Välimäen [4] ehdottamalla tavalla.

[1] <https://github.com/ollpu/SiTraNoStar>



Kuva 1: Spektrogrammi kahdesta marimban äänestä hetkillä 0.5 s ja 1.25 s (ylävsemällä). Ajanhetken $P_t = 0.64$ s sarake näytetään erikseen oikealla, ja taajuuden $P_f = 1170$ Hz rivi näytetään alla. Viivagraafeissa esitetään alkuperäinen ja mediaanisuodatettu data. Pystysuuntainen mediaanisuodatoin (oikealla) hylkää sinikomponentit ja vaakasuuntainen (alla) transienttien huiput. Leikkauspisteessä (P_t , P_f) vaakasuuntaisen mediaanisuodattimen arvo on suurempi (ks. ympyrät), joten se luokitellaan tonaaliseksi.

SiTraNo*-sovelluksessa kolmitiehajotelma tehdään esikäsittelynä, kun käyttäjä avaa äänitiedoston. Mediaanisuodatus on verrattain raskasta ja muodostuu esikäsittelyn pulonkaulaksi, sillä käyttäjä joutuu odottamaan sen valmistumista ennen kuin sovellus voi toistaa ääntä. Tämän vuoksi SiTraNo*⁺:ssa tehostettiin mediaanisuodatuksen toteutusta. Ikkunan alkioita ylläpidetään kahdessa kekorakenteessa, joista toisessa on mediaania pienemmät alkiot ja toisessa sitä suuremmat alkiot. Tämä on todettu tehokkaaksi algoritmiksi, kun mediaani lasketaan alle 200 pisteestä kerrallaan [9], kuten SiTraNo*⁺:ssa. Esikäsittely valmistuu tällä algoritmilla noin 5 kertaa nopeammin kuin SiTraNo*⁺:ssa.

3 ÄÄNEN TOISTONOPEUDEN MUOKKAUS

Esikäsittelyn jälkeen SiTraNo* käyttää kullekin komponentille eri menetelmää toistonopeuden muokkaukseen. Sineille sovelletaan vaihelukittua vokooderia (engl. *identity phase-locking vocoder*) [10]. Yksittäiset transientit leikataan ja sijoitellaan uudelleen aikavenytyksen määräämään kohtaan. Kohinalle käytetään kohinanmuotoilua [5].

Vaihevokooderi on tekniikka, jossa ääni analysoidaan STFT:n avulla lyhyissä ikkunoissa (SiTraNo*⁺:ssa Hann-ikkuna, pituus 4096 näytettä eli 93 ms), aika-taajuuspisteiden vaihekehitys arvioidaan ja lopuksi ikkunat syntetisoidaan takaisin käänteismuunnoksella. Äänen toistonopeutta muokataan analysoimalla ikkunoita tiheämmin tai harvemmin, mutta syntetisoimalla ikkunoita vakiovälein (1/8 ikkunan pituudesta). Perinteinen menetelmä koherenssin säilyttämiseksi päättelee vaihekehityksen hetkellisen taajuuden kautta. Hetkellinen taajuus on mielekästä laskea taajuuskaistan peräkkäisistä pisteistä jos oletamme, että kaistalla esiintyy vain yksi sinikomponentti. Vaihevokooderin ongelmana on kuitenkin, että vierekkäiset taajuuskaistat eivät kertyvien virheiden takia pysy synkronoi-

tuina. Käyttämämme ratkaisu on vaihelukittu vokooderi [10], joka lukitsee heikompien taajuuskaistojen vaihekehityksen samaksi kuin vahvimman läheisen kaistan.

SiTraNo* tunnistaa yksittäiset transientit erotellun signaalin itseisarvon muutoksen huipusta. Kovin lähekkäiset (alle 90 ms) huiput katsotaan samaksi transienttitapahtumaksi. Tämän tiedon perusteella transientit “leikataan ja liimataan” aika-alueessa siten, että transienttitapahtumat alkavat aikaskaalauksen mukaan lasketuille hetkillä. Näin transientit eivät leviä aikaskaalauksen takia.

Kohinalle on ominaista jatkuva satunnainen vaihtelu sekä STFT:n itseisarvossa että vaihessa. Kun kohinasignaalia hidastetaan, satunnaisvaihtelun tulisi edelleen tapahtua nopeasti. Lähekkäiset aika-taajuuspisteet kuitenkin korreloivat keskenään, koska niiden aikaikkunat ja taajuuskaistat ovat päällekkäisiä. Siispä esimerkiksi vaiheiden arpominen tasajakaumasta ei ole luonnollisen kuuloista.

Kohinanmuotoilu (engl. *noise morphing*) [5] on uusi menetelmä, jossa tuotetaan reaaliajassa valkoista kohinaa, jota analysoidaan STFT:llä samoilla parametreilla kuin alkuperäistä signaalia (93 ms Hann-ikkuna). Synteesiä varten alkuperäisen signaalin STFT:n magnitudit kerrotaan valkoisesta kohinasta saaduilla kompleksisilla spektriarvoilla [5]. Näin aikaskaalauksessa kohinasignaali on samanlaista satunnaisvaihtelua, kuin valkoisessa kohinassa tavallisesti. Samalla vältetään vaihe-erot peräkkäisten ikkunoiden välillä, koska kaikki muokatut ikkunat saavat vaiheet samasta valkoisesta kohinasta.

4 KÄYTTÖLIITTYMÄ

SiTraNo*-sovelluksen käyttöliittymä on esitetty kuvassa [2]. Yläpuolisko on ohjausta varten: siinä valitaan äänitiedosto, miksataan eri komponentit, säädetään toistonopeus sekä valitaan transienttien ja kohinan muokkausalgoritmit. Alapuolisko visualisoi toistettavan signaalin aaltomuodon, hetkellisen magnitudispektrin ja spektrogrammin.

Käyttäjä voi avata käsiteltäväksi yhden äänitiedoston kerrallaan (kansiokuvake). Avaamisen jälkeen tiedosto esikäsitellään eli hajotellaan sineihin, transientteihin ja kohinaan luvun [2] mukaisesti. Uusi tallennuspainike (levykekuvake) antaa käyttäjän tallentaa muokatun äänitteen WAV-tiedostoksi nykyisillä nopeus- ja miksausasetuksilla. Mikserissä komponenttien äänenvoimakkuutta voi säätää erikseen tai komponentin voi mykistää kokonaan painamalla painiketta ylhäällä. Master-liuku säätää lopullisen äänenvoimakkuuden. Signaalien hetkelliset tasot näkyvät liukujen vieressä.

Aikaskaalauksen valinta (*Time Scale Modification*) kuvassa [2] yläoikealla ohjaa luvun [3] mukaista äänen venytystä eli sitä säätämällä äänen sävelkorkeus ei muutu. Suuremmat arvot hidastavat toistoa. Liukusäätimen alapuolella olevat painikkeet “preserve transients” ja “noise morphing” valitsevat ovatko transienttien uudelleensijoittelu ja kohinanmuotoilu aktiivisia. Jos menetelmä otetaan pois käytöstä, sen tilalla toimii vaihelukittu vokooderi. Näin voidaan arvioida miten menetelmät vaikuttavat äänenlaatuun.

Alempi toistonopeuden säädin (*Playback Speed*) kuvassa [2] ohjaa uudelleennäytteistykseen perustuvaa venytystä, joka on tarjolla vertailun vuoksi. Säätimen suuret arvot nopeuttavat toistoa ja pienet hidastavat. Tämä vastaa ääninauhan toistonopeuden muuttamista, joten äänenkorkeus muuttuu samalla. Asettamalla aikaskaalauksen ja toistonopeuden valinnat samaksi, säätöjen nopeusvaikutukset kumoavat toisensa ja ainoastaan sävelkor-



Kuva 2: SiTraNo*-sovelluksen käyttöliittymä, jossa on valittu tiedosto ‘Electropop.wav’. Liikusäädöt osoittavat, että transientteja vahvistetaan 2 dB ja kohinaa vaimennetaan 1.5 dB. Venytykseksi on valittu 3.53-kertainen hidastus (yläoik.). Näytön alaosassa piirretään käsitellyn äänen aaltomuoto (kesk. vas.), hetkellinen magnitudispektri (alavas.) ja spektrogrammi (alaoik.), missä äänen kolme komponenttia näytetään eri väreillä.

keus muuttuu. Tätä helpottaa “pitch shift” -painike, joka lukitsee liu’ut samaan arvoon. “Reset both” -painike palauttaa toistonopeuden ja sävelkorkeuden alkuperäiseksi (1.00).

Spektrogrammi käyttöliittymän alaoikealla kuvassa 2 havainnollistaa toistetun äänen magnitudia aika- (vaaka) ja taajuusakseleilla (pysty). Komponentit erotellaan erivärisillä desibeliasteikoilla: sinit piirtyvät punaisena, transientit vihreänä ja kohinaa sinisenä. Näiden yhdistelmät piirtyvät additiivisella värinmuodostuksella. Selkeyden parantamiseksi eri komponenttien spektrogrammit lasketaan kolmella eri STFT:llä, joilla on eripituiset Hann-ikkunat. Sinien komponentilla on pisin ikkuna (186 ms), transienteilla lyhyin (46 ms) ja kohinalla pituus on näiden välistä (93 ms). Pitkä ikkuna antaa hyvän taajuusresoluution, jotta sinikomponentit erottuvat myös matalilla taajuuksilla. Lyhyt ikkuna antaa puolestaan hyvän aikaresoluution, jotta lyhyet transientit erottuvat toisistaan kapeina vihreinä pystyviivoina, joita näkyy kuvassa 2 alaoikealla.

5 YHTEENVETO

Artikkelissa selitettiin äänen kolmitiehajotelma tonaalisiin, transienttimaisiin ja kohinaiisiin komponentteihin sekä siihen perustuva äänen toistonopeuden muokkaus. Esitelimme SiTraNo*-sovelluksen, joka muuttaa äänitteen aikaskaalaa näillä menetelmillä. Sovelluksen uusi versio soveltuu paitsi opetuskäyttöön, myös käytännön äänenkäsittelytyöhön tallennusominaisuuden ja toimintanopeuden ansiosta. Parametreiltaan optimoitu kolmitiehajotelma ja laadukkaat muokkausmenetelmät eri komponenttityypeille, kuten kohinanmuotoilu [5], tuottavat erinomaisia tuloksia toistonopeuden muokkaukseen monenlaisilla äänitteillä, kuten musiikilla ja laululla.

Kolmitiehajotelmaa voidaan hyödyntää laajasti äänenkäsittelyssä. Eroteltujen komponenttien erillinen käsittely ja äänenvoimakkuuden säätö voivat osoittautua hyödylliseksi esimerkiksi musiikin uudelleenmiksauksessa. Hidastamisen tai nopeutuksen lisäksi SiTraNo* soveltuu myös musiikkiäänitteen sävelkorkeuden tai sävellajin muuttamiseen.

VIITTEET

- [1] J. Driedger, M. Müller ja S. Ewert. Improving time-scale modification of music signals using harmonic-percussive separation. *IEEE Signal Processing Letters*, 21(1):105–109, 2014.
- [2] L. Fierro. *Audio Decomposition for Time Stretching*. Väitöskirja, Aalto-yliopiston sähkötekniikan korkeakoulu, 2024. <https://aaltodoc.aalto.fi/items/f63763f5-9503-451d-9533-c57930afa5ad>.
- [3] E.-P. Damskögg ja V. Välimäki. Audio time stretching using fuzzy classification of spectral bins. *Applied Sciences*, 7(12):1293, 2017.
- [4] L. Fierro ja V. Välimäki. Enhanced fuzzy decomposition of sound into sines, transients, and noise. *Journal of the Audio Engineering Society*, 71(7/8):468–480, 2023.
- [5] E. Moliner, L. Fierro, A. Wright, M. S. Hämäläinen ja V. Välimäki. Noise morphing for audio time stretching. *IEEE Signal Processing Letters*, 31:1144–1148, 2024.
- [6] L. Fierro ja V. Välimäki. SiTraNo: A MATLAB app for sines-transients-noise decomposition of audio signals. Kirjassa *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, s. 73–80, Wien, Itävalta (etänä), 2021.
- [7] T. Sinjanakhom, L. Fierro ja V. Välimäki. SiTraNo+: An audio application for sound decomposition and time-scale modification. Kirjassa *Akustiikkapäivät*, s. 161–166, Tampere, 2023.
- [8] D. FitzGerald. Harmonic/percussive separation using median filtering. Kirjassa *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, s. 217–220, Graz, Itävalta, 2010.
- [9] J. Suomela. Median filtering is equivalent to sorting. *arXiv preprint*, arXiv:1406.1717, 2014.
- [10] J. Laroche ja M. Dolson. Improved phase vocoder time-scale modification of audio. *IEEE Transactions on Speech and Audio processing*, 7(3):323–332, 1999.