ARCTIC POWER

Maatutkan luotausdatan kehittäminen wavelet-tekniikalla

Selvitysraportti BigData-analytiikan menetelmän kehittämisestä

Tekijä Kari Peisa

Päiväys: 15.6.2016

Tiivistelmä

Tässä selvityksessä esitetään menetelmä, jossa maatutkan luotausdatasta tuotetaan suodatettua ja konvergoitua dataa visualisointitarkoituksia varten. Menetelmä liittyy Big Datan paikalliseen analytiikkaan siten, että maatutkan massiivisesta luotausdatasta saadaan pilvipalveluihin tallennettua dataa, joka on tehokkaampi käsitellä esimerkiksi visualisoinneissa ja jossa tietyt halutut piirteet ovat säilyneet. Menetelmässä pyritään erottamaan tutkadatasta maakerrostumien antamat heijastevasteet. Menetelmä perustuu jatkuvan wavelet-muunnoksen hyödyntämiseen pulssimuotoisen datan erottamiseksi kohinaa ja muuta häiriöoskillaatiota sisältävästä signaalista. Selvityksessä on esitetty wavelet-tekniikasta lähinnä algoritmien implementoimisen kannalta oleelliset seikat, teoreettista ja matemaattista perustaa on käsitelty vähemmän. Lisäksi käsitellään hieman diskreetin wavelet-muunnoksen soveltuvuutta käytettyyn menetelmään.

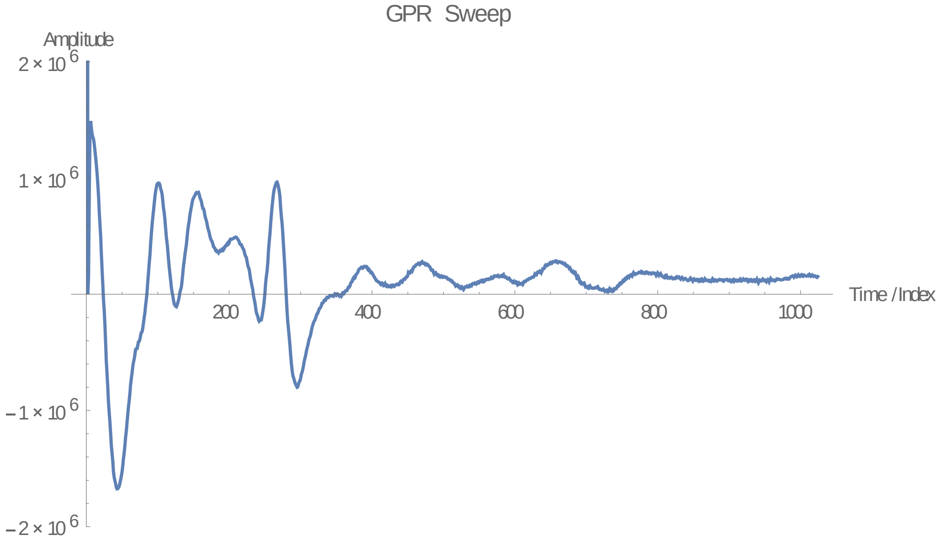
Avainsanat: Signaalin prosessointi, wavelet analyysi, jatkuva wavelet-muunos, ilmavastemaatutkan luotausdata

SISÄLLYSLUETTELO

1 Johdanto

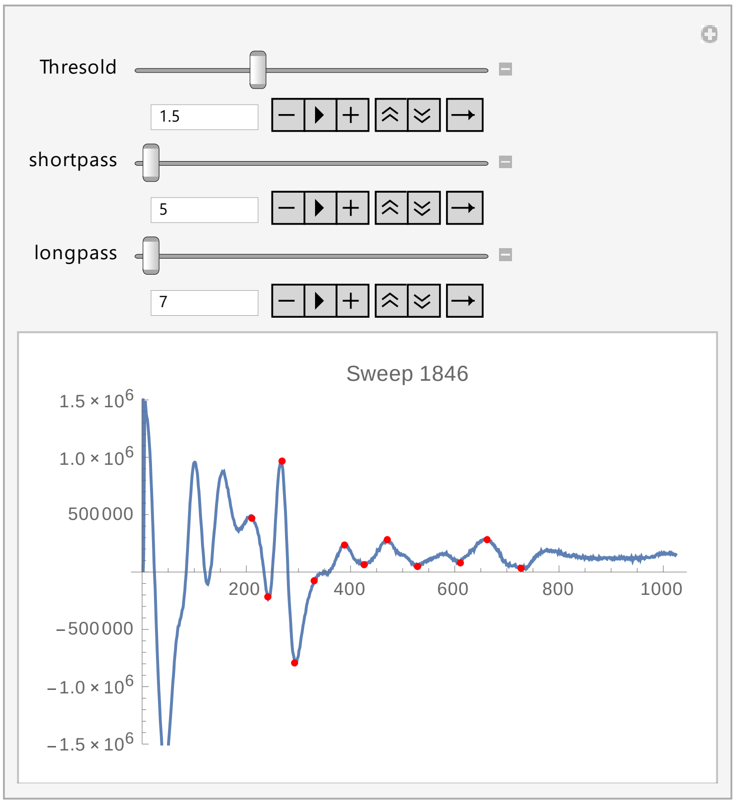
Tässä selvitysraportissa esitetään wavelet-analyysiin perustuva menetelmä, jonka avulla voidaan kehittää maatutkan luotausdataa Big Datan visualisointipalveluihin soveltuvaan muotoon. Wavelet-analyysia on yleisesti hyödynnetty myös muissa Big Datan analyysi- ja prosessointitekniikoissa.

Ilmavasteantenniin perustuvien maatutkien luotausdata koostuu peräkkäisistä pyyhkäisysignaaleista, jotka ovat tutkaheijasteita antennin alapuolisesta materiasta (**Kuva 1.**). Signaalin alkuosa sisältää antennin ja ilman kytkennästä johtuvat heijastevasteet. Alustassa olevat erilaiset rajapintakerrostumat, joissa kerrostumien sähköiset permittiivisyydet eroavat riittävästi, näkyvät signaalissa jäljempänä enemmän tai vähemmän selvästi pulssimaisina muotoina. Pulssi voi olla polaroitunut ylös- tai alaspäin riippuen siitä, mihin suuntaan materian permittivisyys muuttuu rajapinnassa. Hyvin yleistä on myös, että rajapinnat aiheuttavat kerrannaisheijastumia joissa tutkan lähettämä pulssi on heijastunut useammin samojen rajapintojen välillä. Tutka voi tuottaa myös itse kohinan lisäksi pienempitaajuisia häiriöpulsseja, mitä kutsutaan usein soimiseksi. Edellä mainitut syntyneet pulssit myös interferoivat osuessaan yhteen, minkä vuoksi heijastesignaalien tulkinta vaatiikin yleensä kokeneen ammattilaisen näkemyksen.



Kuva 1. Maatutkan pyyhkäisysignaali

Tässä selvityksessä ei paneuduta signaalin ja erilaisten rajapintakerrostumien välisen yhteyden analyysiin, vaan siihen miten luotausdataa voidaan kehittää niin, että siitä voitaisiin automaattisesti tuottaa alapuolisen rakenteen visuaalisia esityksiä. Heijastesignaaleja syntyy luotauksessa hyvin suuri määrä ja koko luotausdatan käsitteleminen vaatii myös massiivista tietokonelaskentaa. Wavelet-tekniikalla on mahdollista poimia signaaleista ne maksimi- ja minimikohdat ja niiden amplitudit, jotka ovat todellisen heijastepulssin kanssa riittävän yhdenmuotoisia taajuudeltaan ja myös muodoltaan. Menetelmän algoritmissa riittävyys voidaan parametrisoida siten, että ennen lopullista muunnosta tutkaaja voi dynaamisesti dataa tutkimalla määritellä riittävyyden tason (**Kuva 2.**).



Kuva 2. ***Mathematica*n funktion Manipulate tuottama interaktiivinen käyttöliittymä, jossa kolme wavelet-suodatuksen tasoa säätävää kontrollimuuttujaa. Punaiset pisteet signaalissa ovat Wavelet-suodatuksen tuloksena paikannetut maksimi- ja minimipisteet indeksi-ikkunassa [200,1000].**

2 Menetelmän kuvaus ja tekninen tausta

2.1 Menetelmän vaiheet

Selvitettävän menetelmän keskeiset vaiheet ovat:

- Käsitellään wavelet-tekniikalla jokainen tutkasignaali erikseen ja pyritään paikantamaan ja jättämään signaaliin vain todellisista rajapinnoista tulevat heijastuspulssien maksimi- ja minimipisteet.
- Prosessointi tehdään automaattisesti koko datalle, kunhan tutkaaja on valinnut soveltuvat parametrien arvot algoritmille.
- Prosessoitu tutkadata voidaan tallentaa Big Datan pilviympäristöön valmiiksi muotoon, josta erilaiset visualisoidut pistepilviprofiilikuvat saadaan nopeasti tuotettua palvelimella olevan käsittelyohjelmiston avulla.

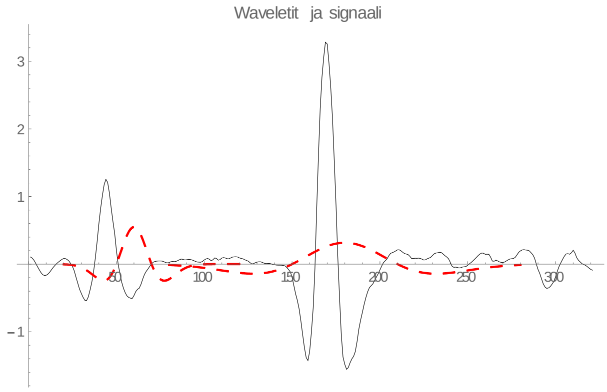
Menetelmän avulla tutkadata saadaan kehitettyä muotoon, joka alkuperäiseen dataan verrattuna vaatii huomattavasti vähemmän tallennuskapasiteettia ja jonka käsittely on nopeampaa ja soveltuvampaa visualisointialgoritmeille.

2.2 Wavelet-tekniikasta

Tässä esityksessä ei käsitellä wavelet-tekniikan matemaattisia periaatteita syvällisesti, vaan ainoastaan algoritmin implementoimisen kannalta oleelliset asiat. Algoritmin kehittäminen ja testaus on suoritettu *Mathematica*n versiolla 10.2.0.0, joka sisältää sisäänrakennetut perusfunktiot wavelet-muunnoksien tekemiseen ja niiden prosessointiin sekä lisäksi erityisen interaktiiviseen tutkimiseen soveltuvan funktion Manipulate. Funktio mahdollistaa interaktiivisen käyttöliittymän nopean rakentamisen. Käyttäjän määrittämiä kontrollimuuttujia säätämällä voidaan dynaamisesti tutkia parametrien vaikutusta haluttuun lopputulokseen

(**Kuva 2.**). Funktion toiminta oli jostain syystä hyvin epävakaa ja se kaatoi ohjelman usein. Funktion dynaaminen jatkuva evaluointi kaatoi käyttöliittymän toiminnan usein muutettaessa jonkin liukumuuttujan arvoa. Tällöin seurauksena oli, että myös yhteys muistissa oleviin jo evaluoituihin muuttujiin katosi eli *Mathematica*n Kernel kaatui.

Wavelet-funktiot ovat matemaattisesti määriteltyjä aaltomaisia funktiota, joita käytetään maskina analysoitaessa alkuperäisen signaalin ominaisuuksia. Matemaattisesti wavelet-muunnos toteutetaan konvoluutioilla, jotka tuottavat numeerisen arvon alkuperäisen signaalin ja maskina toimivan wavelet-funktion väliselle vastaavuudelle. Konvoluutiossa wavelet funktioita siirretään määrätyin askelin signaalin alusta loppuun. Lisäksi wavelet funktion muotoa muutetaan siten, että vastaavuudet eri taajuuksiin kullakin ajankohdalla saadaan esille ja vielä siten, että kaikki syntyneet wavelet-kertoimet ovat keskenään vertailtavissa (**Kuva 3**.).



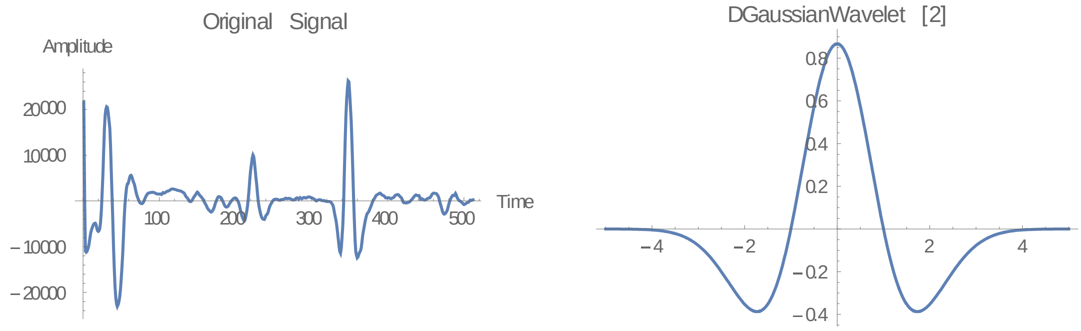
Kuva 3. Alkuperäinen signaali ja wavelet-funktioita (katkoviiva). Amplitudit on muutettu havainnollisuuden vuoksi.

Wavelet tekniikalla mikä tahansa aikaan (tai myös paikkaan) perustuva vaihtelu muunnetaan indeksoiduiksi lukuarvoiksi, jotka pitävät sisällään informaatiota kunkin ajankohdan vaihtelun voimakkuudesta ja jaksonpituudesta (taajuudesta). Tekniikkaan kuuluu mahdollisuus palauttaa muunnettu informaatio käänteismuunnoksella takaisin alkuperäiseksi dataksi. Ennen käänteistä muunnosta muunnettuun informaatioon voidaan tehdä haluttuja muutoksia, joilla esimerkiksi pyritään poistamaan häiriöksi luokiteltavaa vaihtelua tai parantamaan erottelukykyä havaintojen tekemiseksi.

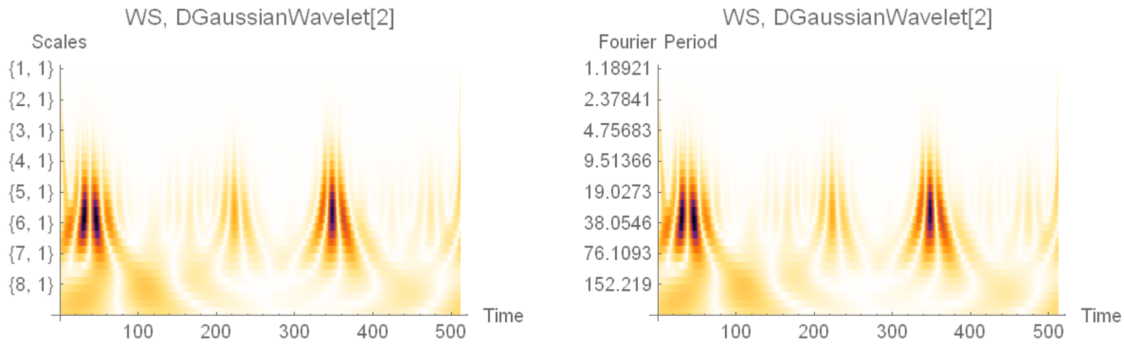
2.2.1 Signaalin jatkuva wavelet-muunnos

Jatkuvan wavelet-muunnoksen kertoimen arvo kertoo signaalin ja aaltofunktion vastaavuuden eli konvoluution arvon. Voimakas vastaavuus tuottaa itseisarvoltaan suuren arvon. Kertoimet tallennetaan tietorakenteeseen, joka voi olla useiden listojen muodostama 2-ulotteinen taulukko tai myös yksiulotteinen lista, jossa kukin lista voidaan lukea tietyllä indeksointiperiaatteella. Kerroinlistojen muodostama tietorakenne on samalla signaalin aika-jaksonpituustason kuvaus, joka esitetään **Kuvassa 4**. näkyvällä **skaalogrammilla** (scalogram). Listoihin liittyvät indeksit kuvaavat aaltofunktion mittaaman oskillaation jaksonpituutta ja kertoimen paikka listassa kertoo samalla mitatun oskillaation paikan alkuperäisessä aikasarjassa. On huomioitavaa, että jatkuvan wavelet-muunnoksen kertoimet sisältävät redundanttia tietoa eli saman oskillaation informaatio on tallentunut usealle kertoimelle.

Skaalogrammissa wavelet-kertoimien suuruus kuvataan yleensä väreillä, jolloin saadaan skaalogrammista itse asiassa 3-ulotteinen kuva. Kuvassa 3. on esitetty signaalin wavelet-muunnos, käytetty aaltofunktio ja skaalogrammit, joissa näkyy myös wavelet-skaalojen indeksoinnin ja alkuperäisen signaalin eri oskillaatioiden jaksonpituuksien (Fourier period) vastaavuudet.



a)



b)

Kuva 4. a) Alkuperäinen signaali ja wavelet funktio (Gaussin funktion 2. derivaatta)

b) Wavelet skaalogrammi. Vasemmalla pystyakselin indeksit kuvaavat Wavelet funktioiden jaksonpituuksia alaspäin kasvaen, oikealla vastaavat Fourier jaksonajat, jotka vastaavat todellisen signaalin jaksonpituuksia

Wavelet-suodatuksen ja Wavelet-kertoimien prosessoinnin kannalta eri ohjelmointiympäristöissä oleellista on tietää wavelet-muunnoksen tietorakenteessa käytetty indeksointiperiaate, joka saattaa vaihdella. Lisäksi on huomioitava, että alkuperäisen signaalin todelliset havaitut jaksonpituudet saadaan Fourierin muunnoksen avulla ja ne voidaan laskea tarkasti kullekin wavelet-funktiolle omilla laskentakaavoillaan. Seuraavassa wavelet-suodatuksen perustaa tarkastellaan lähemmin.

2.2.2 Wavelet-tekniikan suodattamisesta

Oktaavit (octaves), kaistat (voices) ja wavelet skaalat

Fourierin muunnoksen yhteydessä puhutaan aina alkuperäisen signaalin taajuuksista, koska siinä muunnoksessa aikasarja esitys muunnetaan taajuusspektriksi. Wavelet-muunnoksen yhteydessä puhutaan yleensä jaksonpituuksista eli taajuuksien käänteisluvuista. Muunnoksen koko jaksonpituusalue jaetaan **oktaaveihin**. Oletusarvoisesti oktaavien lukumäärä saadaan kaavasta $\log_2 \frac{N}{2}$, missä *N* on aikasarjan pituus. Kukin oktaavi on jaettu **kaistoihin**, joilla oktaavin jaksonpituusalue muuttuu tasavälein edellisestä oktaavista seuraavaan. Kaistojen lukumäärän oletusarvo riippuu ohjelmointiympäristöstä, *Mathematica*ssa se on 4.

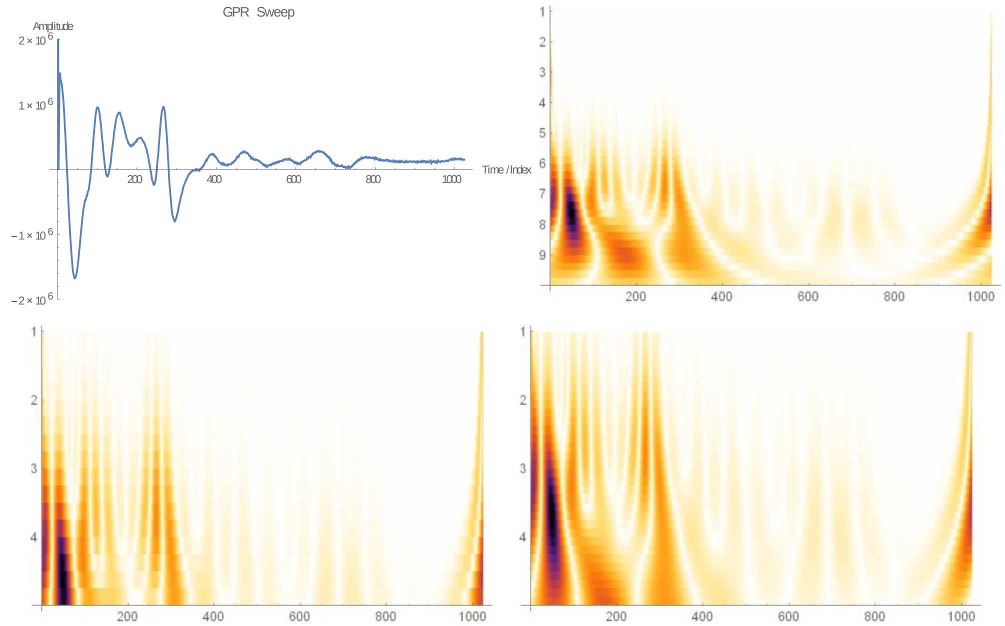
Maatutkan $512 = 2^9$ amplitudiarvon signaaliin tulee siis $\log_2 \frac{2^9}{2} = 8$ oktaavia kuten yllä kuvassa 2, jonka skaalogrammeissa tarkasti katsoen näkyy 4 kaistaa kussakin oktaavissa. Nimi skaalogrammi tulee sanasta **skaala**, joka liittyy wavelet-funktion skaalaamiseen koko jaksonpituusalueelle sopivaksi konvoluutioita varten. Pienimmän resoluution ns. **wavelet-skaala** on oletusarvoisesti 1. Tämä voidaan ymmärtää hyvin karkeasti ilmaisten, että 1. oktaavin 1. kaistalla {1,1} wavelet-kertoimet säilyttävät informaatiota noin 1 indeksin pituisen jakson oskillaatioista. Seuraavien oktaavien 1. kaistalla jaksonpituuden indeksit aina kaksinkertaistuvat. Kaistalla {2,1} $2 \cdot 1 = 2$, kaistalla {3,1} $2 \cdot 2 = 2^2$ jne., jolloin viimeisen oktaavin kaistan

{8,1} kertoimet liittyvät $2 \cdot 2^7 = 256$ indeksin pituisen jakson oskillaatioihin. Näin ollen viimeisen kaistan {8,4} jaksonpituuden indeksimäärä on sama kuin sarjan pituus eli esimerkissämme 512. On kuitenkin huomattava, että jatkuvan wavelet-muunnoksen kertoimet ovat redundanttisia eli sisältävät päällekkäistä informaatiota. Lisäksi, kuten edellä jo mainittiin, tarkat signaalin oskillaatioiden jaksonpituudet kullakin kaistalla voidaan laskea erityisillä kaavoilla.

Kaikkien kaistojen skaalat saadaan laskettua kaavasta $ws \cdot 2^{\frac{k}{nvoc}}$, missä ws = pienimmän resoluution wavelet-skaala ja k käy läpi arvot 1:stä arvoon $noct \cdot nvoc$, missä $nvoc$ = kaistojen ja $noct$ = oktaavien lukumäärä. Kaistan ja oktaavin indeksi saadaan luvusta k jakojäännöksinä $\mathrm{mod}(k, nvoc)$ ja $\mathrm{mod}(k, noc)$ vastaavasti.

Ohjelmointiympäristöissä wavelet-muunnoksen funktioille voidaan yleensä syöttää oktaavien ja kaistojen haluttu määrä sekä myös pienimmän resoluution wavelet-skaalan arvo. **Kuvassa 3.** on esitetty 1024 amplitudiarvon signaalin jatkuvan wavelet-muunnoksen kolme eri skaalogrammia, joissa oktaavien ja kaistojen sekä myös pienimmän resoluution wavelet skaalan arvoja on vaihdeltu. Kuvasta ilmenee kuinka näitä arvoja vaihtamalla wavelet-tekniikalla voidaan keskittyä juuri halutulle jaksonpituusalueelle.

Kuvan 5. skaalogrammeista nähdään miten pienimmän resoluution wavelet skaala vaikuttaa kaikkien oktaavien ja kaistojen mittaamiin jaksonpituuksiin edellä kuvatulla tavalla eli jaksonpituudet kaksinkertaistuvat aina yhden oktaavin alueella. Valitsemalla 1:stä suurempi pienimmän resoluution wavelet-skaala ja toisaalta pienempi oktaavien lukumäärä sekä suurempi kaistojen lukumäärä voidaan wavelet-muunnos kohdentaa tarkemmin juuri halutuille jaksonpituusalueille. Tällä on merkitystä toisaalta resoluution parantamiseen mutta myös laskennan vähentämiseen eli tehokkuuteen. (1, Addison, 2, Mathiak, 4, *Mathematica*)



Kuva 5. a) Alkuperäinen signaali, b) {oct, voc} = {9,4}, ws (wavelet skaala) = 1,
c) {oct, voc} = {4,4}, ws = 2, d) {oct, voc} = {4,8}, ws = 4

Wavelet-kertoimien prosessointi

Wavelet-muunoksen kertoimien prosessointi on yksinkertaisimmillaan sitä, että muunnetaan haluttujen kaistojen tai wavelet-kertoimien arvot nollaksi ennen takaisin muunnosta signaaliksi. Kun halutaan etsiä ja paikantaa pulsseja signaalista, on tarkoituksenmukaista parametrisoida edellinen siten, että kaistoja nollataan alku- ja loppupäästä lähtien eli tarkentaen kohti pulssille ominaisia jaksonpituuksia (scale-dependent thresholding). Lisäksi pulssin muodon sopivuutta voidaan mitata konvoluution arvolla eli wavelet-kertoimen itseisarvon suuruuden perusteella (thtresolding).

Kuvassa 2 on käyttöliittymä, jossa wavelet-pohjainen suodatus on parametrisoitu juuri edellä kuvatulla tavalla. Siinä nollataan kokonaisia oktaaveja, ei oktaavien yksittäisiä kaistoja. Kontrollimuuttuja *shortpass* lisää nollattavia oktaaveja alkupäästä eli suodattaa pois lyhytjaksoista vaihtelua. Kontrollimuuttuja *longpass* puolestaan lisää nollattavia oktaaveja loppupäästä eli suodattaa pois pitkäjaksoista vaihtelua. Kontrollimuuttuja *Thresold* nollaa wavelet-kertoimia, joiden itseisarvo on liian pieni.

Suodatettaessa wavelet-kertoimen itseisarvon perusteella on löydettävä jokin järkevä kynnysarvo, mitä itseisarvoltaan pienemmät tai suuremmat wavelet-kertoimet nollataan. Tässä kohdin on wavelet-muunnoksien yhteydessä usein käytetty hajonnan mittalukua **mediaanin absoluuttista mediaanipoikkeamaa** (Median absolute deviation, **MAD**), joka lasketaan kaavalla

$$MAD = Md(|x_1 - \bar{\boldsymbol{x}}_{\boldsymbol{Md}}|, |x_2 - \bar{\boldsymbol{x}}_{\boldsymbol{Md}}| \dots |x_n - \bar{\boldsymbol{x}}_{\boldsymbol{Md}}|), \qquad (1)$$

missä $\bar{\boldsymbol{x}}_{\boldsymbol{Md}} = Md(x_1, x_1, \dots, x_n)$. Tässä on syytä olla tarkkana, koska usein käytetään lyhenteelle MAD rinnakkaista ilmaisua **mediaanin absoluuttinen keskipoikkeama** (Mean Absolute Deviation), joka lasketaan siten, että kaavassa (1) ei lasketa poikkeamien itseisarvojen mediaania vaan niiden keskiarvo.

$$MAD = Mean(|x_1 - \bar{\boldsymbol{x}}_{\boldsymbol{Md}}|, |x_2 - \bar{\boldsymbol{x}}_{\boldsymbol{Md}}| \dots |x_n - \bar{\boldsymbol{x}}_{\boldsymbol{Md}}|) \qquad (2)$$

Myös keskiluvun $\bar{\boldsymbol{x}}_{\boldsymbol{Md}}$ tilalla voidaan käyttää muitakin keskilukuja useimmin keskiarvoa.

Keskiarvon ja keskihajonnan käyttäminen havaintojoukolle, joka ei ole normaalisti jakautunut, on hyvin riskialtista, koska ne ovat niin voimakkaasti riippuvia erittäin poikkeavista havaintoarvoista. Kun halutaan suuresta joukosta vinosti jakautuneita havaintoarvoja etsiä merkittävästi poikkeavat arvot, on mediaani ja MAD varmempi valinta kynnysarvojen määrittämiseksi. Lisäksi mediaanin laskeminen voidaan suorittaa erittäin tehokkaasti *QuickSelect*-algoritmilla.

Tyypillisesti maatutkan signaalista muodostettu wavelet-kertoimien lista sisältää paljon pieniä arvoja sekä muutamia itseisarvoltaan erittäin suuria konvoluution arvoja, Tällöin on hyvin todennäköistä, että lista on vino. Tällöin kaavan (1) mukainen MAD on luotettavampi kynnysarvon määrittäjä poikkeaville havaintoarvoille kuin keskihajonta. Lisäksi on huomattava, että wavelet-kertoimien mediaani on likipitäen nolla kaikilla listoilla johtuen wavelet-funktioiden määrittelystä. Tällöin MAD voidaan laskea riittävällä tarkkuudella erittäin tehokkaasti suoraan wavelet-kertoimien itseisarvojen mediaanina $MAD = Md(|w_1|, |w_1|, \dots, |w_n|)$.

Kun etsitään signaalista heijastepulssin muotoon parhaiten sopivia pulsseja, saadaan ne etsimällä itseisarvoltaan poikkeavan suuria wavelet-kertoimen arvoja. Tällöin tuo etsintä voidaan parametrisoida säädettävälle muuttujalle c siten, että wavelet-kerroin w nollataan, mikäli

$$|w| < c \cdot MAD, \qquad (3)$$

missä MAD kannattaa laskea yhteisenä kaikille niille kaistoille, joita ei ole vielä nollattu.

Etsittävät tutkan ilmassa vastaanottamat selkeät heijastepulssit ovat aina saman levyisiä ja muodoltaan samanlaisia. Tämä perustuu siihen, että rajapintojen heijastuksissa ja läpimenoissa taajuus ei muutu, joten viime kädessä ilmassa saapuvan heijastepulssin aallonpituuskin on aina sama riippumatta pulssin kulkureitistä alapuolisessa materiassa. Näin ollen, jos etsitään selkeitä pulssiheijastuksia, voidaan suodatuksessa rajoittua yhden tai korkeintaan kahden oktaavin alueelle. Tämä tapahtuu kuvan 5. mukaisesti suurentamalla wavelet-skaalan arvoa ja samalla vähentämällä oktaavien ja lisäämällä kaistojen lukumäärää ilman, että muunnoksen ja käsittelyn vaatima laskentamäärä paisuisi kohtuuttomasti. Eroteltavien jaksonpituuden analysointi kannattaa tehdä joko Fourierin muunnoksella tai wavelet-muunnoksella, jossa wavelet-funktiona käytetään esimerkiksi Morlet-funktiota, joka mittaa jaksonpituutta tarkemmin aikatason tarkkuuden kustannuksella. Wavelet muunnoksien aika-jaksonpituusresoluutiot noudattavat eräänlaista Heisenbergin epätarkkuusperiaatetta. (1, Addison, 3, Posio, 4, *Mathematica*)

2.3 Menetelmän testaus

Johtuen Manipulate funktion epävakaudesta on menetelmän testaus kuvattu jäljessä **Kuvan 6.** algoritmilla, jossa kontrollimuuttujille annetaan arvot ”käsin”. Testaus on suoritettu *Mathematica*n versiolla 10.2.0.0. ja testattavana maatutkatiedostona on käytetty Kuntotekniikan tätä tarkoitusta varten luovuttamaa datatiedostoa `"2_43_111A.DZT"`, jossa mittaus on tehty yhdellä tutka-antennilla. Algoritmi tulostaa havainnollisuuden vuoksi alkuperäisen signaalin (yksittäiset pisteet yhdistetty) ja wavelet-tekniikalla etsittyjen pulssien huiput siten, että minimit ja maksimit tulostuvat eri väreillä. **Kuvassa 7**. on esitetty algoritmin palauttamista kuvaajista joitakin esimerkkejä valituilla kontrollimuuttujien arvoilla, jotka on tulostettu kuvioon.

Algoritmissa tehdään myös havainnollisuuden lisäämiseksi alkuperäisen signaalin amplitudien keskiarvon siirto nollaan. Tällä menettelyllä saadaan lähes kaikkien paikallisten minimipisteiden arvot negatiivisiksi ja maksimipisteiden arvot positiivisiksi. Wavelet-muunnos on täysin invariantti keskiarvon siirron suhteen eli wavelet kertoimien arvot eivät operaatiossa muutu.

Algoritmissa ei ole visualisoitu heijastepulssin amplitudin suuruuden vaikutusta, mikä on alustan rakenteen lopullisen visualisoinnin kannalta tarpeellista.

Algoritmissa Kaikki isolla alkukirjaimella alkavat ruskealla fontilla merkityt tunnisteet ovat *Mathematica*n sisäänrakennettuja funktioita tai määreitä. Funktiot `ContinuousWaveletTransfor` ja `InverseContinuousWaveletTransform` liittyvät wavelet-muunnoksien tekemiseen, jotka löytyvät monista ohjelmointiympäristöistä. Funktio `WaveletMapIndexed` liittyy wavelet-kertoimien prosessointiin ja se on pelkästään *Mathematica*n ympäristölle kehitetty.

Funktio `reCwd` poistaa wavelet-muunnoksesta tarpeettomat imaginääriosat, joiden kerroin on käytännössä nolla, mutta äärellisen laskentatarkkuuden takia niille syntyy muunnoksessa hyvin pieni kerroin.

Muuttuja `octvoices` pitää sisällään suodatettavat oktaavit, esimerkiksi `{{1,1},{1,2},…,{5,3},{5,4},{7,1},{7,2},…,{9,4}}` mikä Mathematican mallin tunnistustekniikalla on koodissa konstruoitu muotoon `{{1|2|3|4|5,_}, {7|8|9,_}}`. Muuttuja `compl` pitää sisällään jäljelle jäävät nollasta eroavat oktaavit `{{6,1},{6,2},{6,3},{6,4}`. Jälkimmäisille wavelet-kertoimille funktio `madAll` laskee mediaanin absoluuttisen mediaanipoikkeaman (MAD), jota käytetään seuraavassa vaiheessa wavelet-kertoimien kynnystämisessä. Funktio madAll käyttää *Mathematica*n sisäänrakennettua funktiota MedianDeviation. Kuten edellä todettiin, tämän funktion voi wavelet-kertoimien tapauksessa implementoida QuickSelect-algoritmilla muotoon $MAD = Md(|w_1|, |w_1|, \dots, |w_n|)$, koska wavelet-kertoimien mediaani on määrittelyjen mukaan tasan nolla.

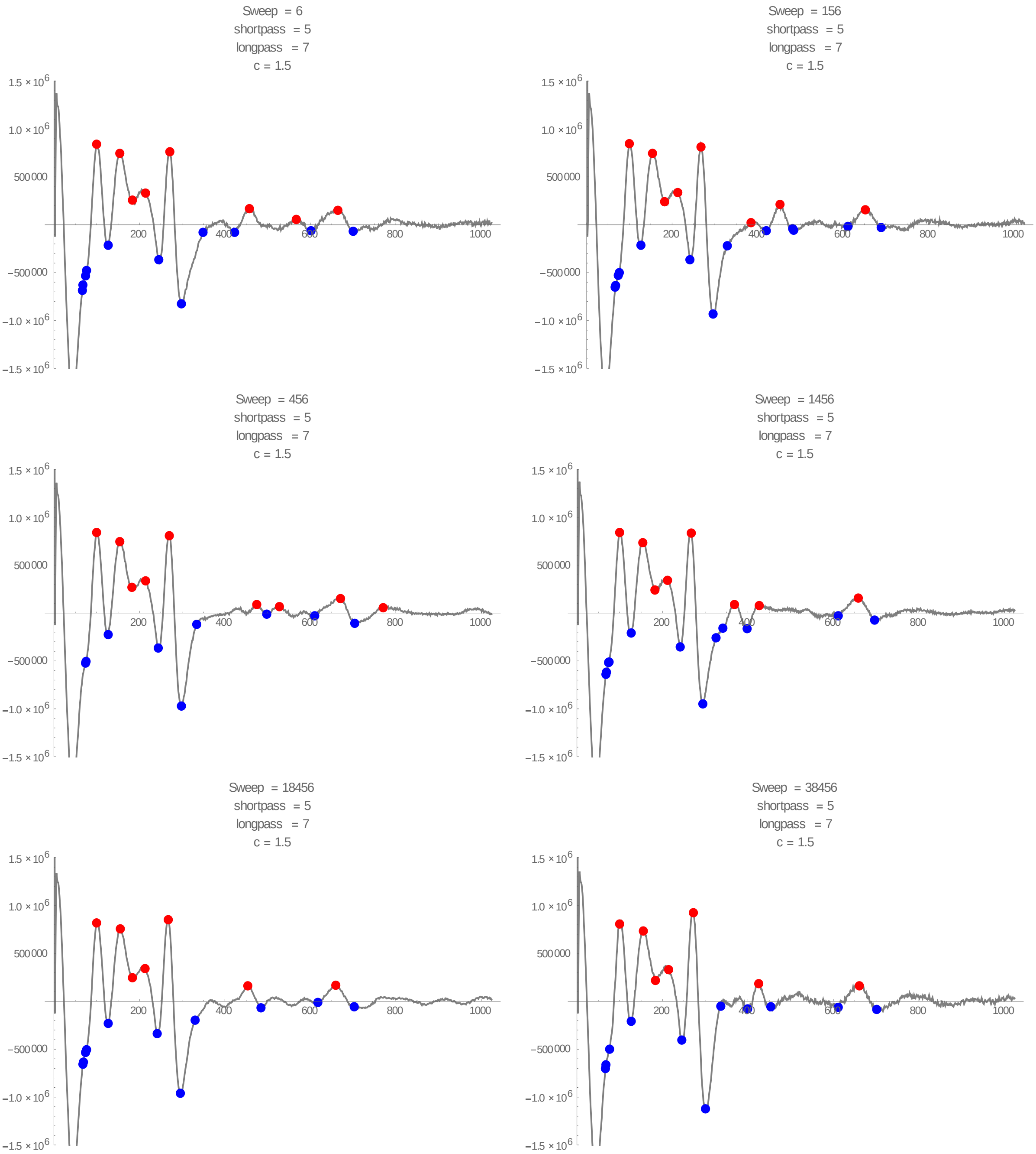
Muuttuja `newscan` on alkuperäisen signaalin wavelet-suodatettu signaali. Funktio `findLocals` etsii ja paikantaa suodatetusta signaalista maksimi- ja minimikohdat, ja palauttaa vastaavaat lokaalit pisteet alkuperäisestä signaalista. Funktiossa paikallinen ääriarvokohta etsitään kaavan (4) mukaisella periaatteella.

$$(list[i+1] - list[i]) \cdot (list[i+2] - list[i+1]) < 0$$
$$\rightarrow i+1 \; on \; lokaali \; \text{ääriarvokohta} \tag{4}$$

Aito "pienempi kuin <" merkitsee, että sellaiset ääriarvot, joissa on kaksi peräkkäistä samaa maksimi- tai minimiarvoa, eivät tallennu. Tällainen tilanne on signaalien amplitudeilla äärimmäisen harvinainen, jolloin pois jäänti ei ole ongelma. Suodatetulla signaalilla tällainen tilanne on taas hyvin yleinen eikä pitkiä tasaisia 0-amplitudisarjoja saakaan tallentaa ääriarvokohdiksi.

```
(* Annetaan kontrollimuuttujille arvot *)
shortpass = 5; (* oktaavit {1,2,3,4,5} nollataan*)
longpass = 7;  (* oktaavit {7,8,9} nollataan *)
c = 1.5; (* kynnysarvo=1.5*MAD, MAD lasketaan oktaaveille {6}*)
sweepNb=59; (* signaali nro 59 joukosta {1, 52119} *)

(* Luetaan muistissa olevasta datasta dataGPR käsiteltävä
   signaali ja tehdään keskiarvon siirto nollaan *)
scan = Flatten[Take[dataGPR, {sweepNb}]];
mean = Mean[scan];
scan = Map[# - mean &, scan];

(* Wavelet muunnos, funktio reCwd poistaa luvuista 0-arvoisia
   kompleksiosia, joita syntyy muunnoksessa *)
cwd = ContinuousWaveletTransform[scan];
cwd = reCwd[cwd];
octaves = cwd["Octaves"];

(* Asetetaan kaikki shortpassin ja longpassin määräämien
   oktaavien kertoimet nollaksi *)
octvoices = {{Apply[Alternatives, Range[shortpass]], _},
   {Apply[Alternatives, Range[longpass, octaves]], _}};
cwd = WaveletMapIndexed[0. # &, cwd, octvoices];

(* Lasketaan MAD nollasta eroaville kaistoille ja asetetaan
   kaikki kynnysarvoa pienemmät kertoimet nollaksi *)
compl = {Apply[Alternatives,
         Range[shortpass + 1, longpass - 1]], _};
MAD = madAll[cwd,compl];
cwd = WaveletMapIndexed[Function[coefList,
    Map[If[Abs[ #] < c*MAD, 0., #] &, coefList]],cwd, compl];

(* Tehdään käänteinen wavelet muunnos *)
newscan = InverseContinuousWaveletTransform[cwd];

(* Etsitään signaalin suodatetut maksimi- ja minimipisteet *)
maxes = findLocals[scan, newscan, {10, 1000}];

(* Tulostetaan signaali ja pistepilveen tulevat pisteet *)
fig1 = ListLinePlot[scan, PlotRange -> {-1500000, 1500000},
       PlotLabel ->
   "Sweep = " <> ToString[sweepNb] <>
   "\nShortpass = " <> ToString[shortpass] <>
   "\nLongpass = " <> ToString[longpass] <>
   "\nc = " <> ToString[c] , PlotStyle -> Gray];
fig2 = Graphics[Map[If[#[[2]] < 0, {Blue, PointSize[0.01],
         Point[#]}, {Red, PointSize[0.01], Point[#]}]&, maxes]];
Show[fig1, fig2],
```

Kuva 6. Wavelet-tekniikan testausalgoritmi *Mathematica*n versiona



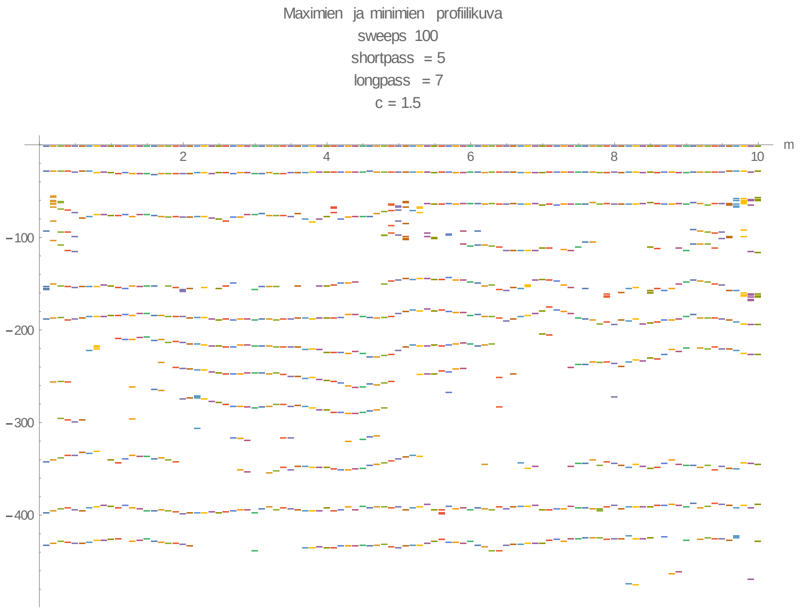
Kuva 7. **Esimerkkkitulostuksia maatutkan luotausdatan kehittämisestä. Wavelet-tekniikalla automaattisesti poimitut pulssien maksimit ja minimit näkyvät värillisinä pisteinä.**

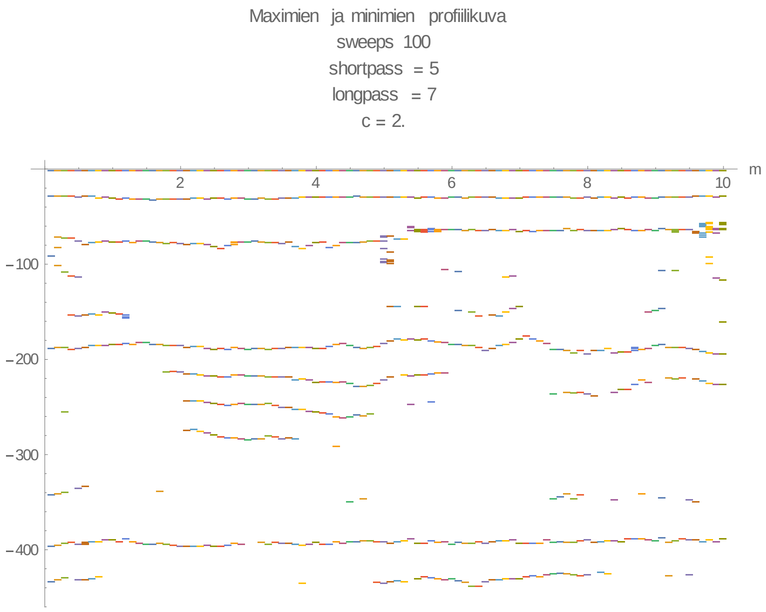
Profiilikuvat

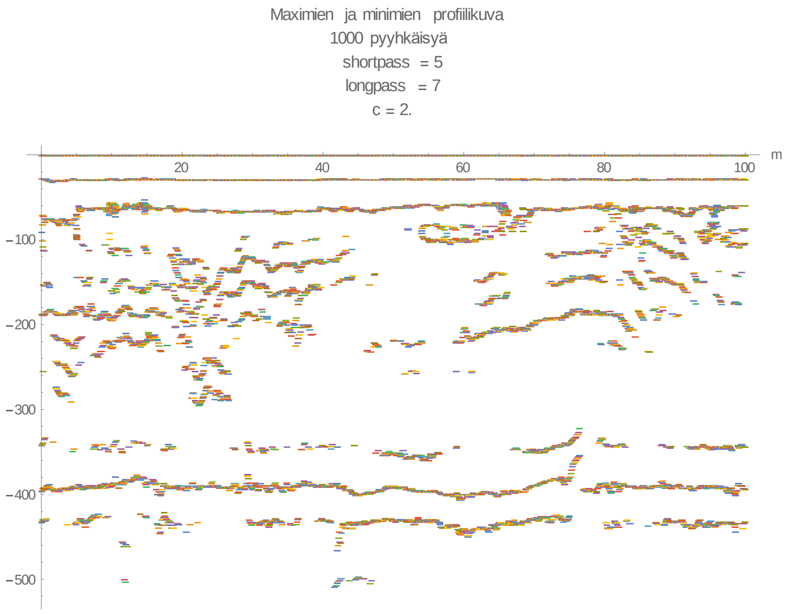
Testauksessa toteutettiin myös pienimuotoinen tutkaluotauksen profiilikuva, jossa wavelet-tekniikalla suodatetut lokaalit ääriarvopisteet kuvataan peräkkäisille signaaleille siten, että pisteiden aika-akselin mukaiset etäisyydet ensimmäisestä pintaheijasteen maksimikohdasta tulostuvat alakkain. Tämä voidaan toteuttaa pisteen $\{x_i, y_i\}$ muunnoskaavalla (5).

$$\left\{\{x_i, y_i\}_{i=(1,maxes)} \rightarrow \{k^*, zeroIndex_k - x_i\}\right\}_{k=(1,N)}, \qquad (5)$$

missä k on signaalin indeksi, $zeroIndex_k$ signaalin k pintaheijasteen maksimin indeksiarvo, joka tulee siis olemaan pintataso eli 0-arvo profiilikuvassa, ja N on kuvaukseen mukaan otettujen signaalien lukumäärä (**Kuva 8.**). Koordinaatin merkintä k^* kuvaa signaalin k mittaaman paikan etäisyyttä mittauksen alkukohdasta, joka tässä on laskettu periaatteella 10 pyyhkäisyä/m, mutta joka saadaan nykysistä datatiedostoista GPS paikantimen tallentamista koordinaateista. Kuvassa 8. ensimmäinen rajapinta on noin tasossa -50. Sitä ylempi suorahko linja on pintaheijastepulssin minimipiste. Kuvat osoittavat, että profiilikuvaan pitää selkeyttämisen vuoksi yhdistää rajapinnan pulssin maksimi ja minimipisteet yhdeksi arvoksi. Tosin tämä ei ole helppoa, koska aina pulssin kaikki kolme huippua (min, max, min) tai (max, min, max) eivät esiinny. Lisäksi kerrostumien syvyyden laskeminen onnistuu 1. kerrostumalle heijastepulssin amplitudiarvon ja metallimittauksen vertailupulssin avulla, mutta ei enää sen jälkeisille kerroksille ainakaan kovin suurella varmuudella. Kaavan (5) toteutuksessa pisteiden amplitudiarvot y_i katoavat.







Kuva 8. Wavelet-tekniikalla suodatettujen maximi- ja minimipisteiden profiilikuvia. Pystyakselilla on pisteen indeksietäisyys pintaheijasteen maksimista, joka sijaistee 0-tasossa.

2.4 Algoritmien tehokkuudesta

Jatkuvan wavelet-muunnoksen ja sen käänteisen muunnoksen laskeminen ovat molemmat kompleksisuudeltaan luokkaa O($N\ Log\ N$). Yhden 1024 amplitudiarvon signaalin wavelet-muunnoksen tai käänteisen muunnoksen tekeminen *Mathematica*lla kestää funktion Timing mukaan noin hieman vaihdellen 2 - 3 ms mitattuna peräkkäin suoritettujen muunnosten ajan keskiarvona. Funktio Timing mittaa ns. CPU-aikaa eli sitä aikaa, joka evaluointiin kuluu keskusyksikössä.

Tyypillisesti Mathematicassa Kuvan 6. algoritmin eli Kuvan 7. grafiikan tarvitseman datan tuottaminen yksittäisen signaalin käsittelyllä vei CPU- aikaa noin 0.030 s. Algoritmi pitää sisällään seuraavat toiminnot:

- Keskiarvon siirto nollaan
- Wavelet muunnoksen tekeminen
- Wavelet-kertoimien prosessointi
- Mediaanin absoluuttisen mediaanipoikkeaman laskeminen
- Käänteisen waveletmuunnoksen tekeminen
- Maksimipisteiden etsiminen

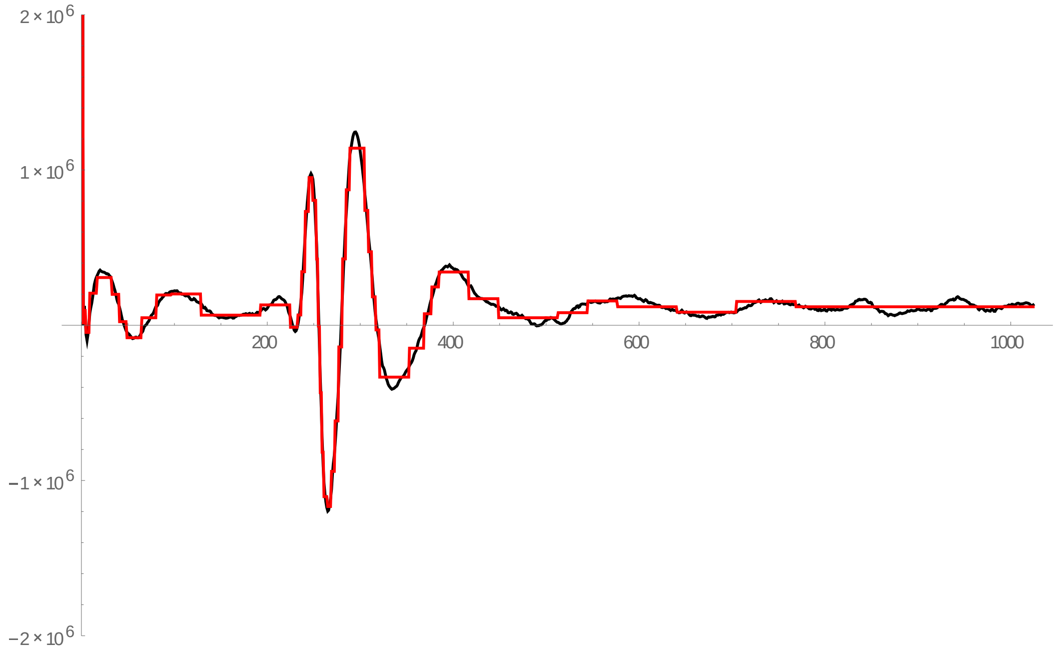
Profiilikuvien tulostamiseen tarvittavan datan prosessoimiseen kuluva aika ei juurikaan muuttunut edellisestä. Siinä algoritmiin tulee lisänä signaalin pintaheijasteen maksimiarvon paikantaminen maksimipisteistä ja maksimipisteiden muuttaminen profiilipisteiksi kaavan (5) mukaisesti, jotka eivät kumpikaan vaadi juuri laskenta-aikaa. 1000 signaalin muodostaman profiilikuvan datan prosessointi kesti *Mathematica*n funktion AbsoluteTiming mukaan vajaa 40 s, kun tutkadata on valmiina keskusmuistissa. Funktio AbsoluteTiming mittaa suoritukseen kuluvaa kokonaisaikaa alusta loppuun. Tällöin 50 000 signaalin tutkadatan profiilikuvien datan konstruointi veisi aikaa noin puolisen tuntia eli liian kauna ajatellen tutkadatan reaaliaikaista prosessointia pilvipalveluna.

2.5 Diskreetin waveletmuunnoksen soveltuvuudesta

Menetelmän testauksessa käytettiin myös diskreettiä wavelet-muunnosta, joka on huomattavasti tehokkaampi ja vähemmän muistikapasiteettia vaativa menetelmä kuin jatkuva wavelet-muunnos. Diskreettissä muunnoksessa wavelet-funktiot muodostavat ortogonaalisen kannan. Tämä tarkoittaa, että funktioiden konvoluutiot tuottavat wavelet-kertoimia, jotka eivät sisällä redundanttia tietoa. Toisin sanoen jokainen kerroin sisältää alkuperäisen signaalin oskillaatioista tiedon, joka ei sisälly muihin kertoimiin. Näin saadaan kompaktein mahdollinen esitysmuoto signaalin oskillaatioista. Diskreetin wavelet-muunnoksen ja sen käänteismuunnoksen laskeminen voidaan suorittaa erittäin tehokkaalla algoritmilla (Fast Wavelet Transform), jonka kompleksisuus on vain lineaarista O(N).

Diskreetin wavelet-muunnoksen kertoimet muodostavat puumaisen rakenteen, jossa kertoimien indeksointi menee eri tavalla kuin jatkuvalla wavele-muunnoksella. Toisin sanoen kertoimien prosessoinnin implementoiminen on toteutettava eri tavalla kuin jatkuvalla muunnoksella.

Diskreetillä wavelet-muunnoksella saadaan poistettua kohinaa erittäin tehokkaasti. Pulssimaisten muotojen paikannuksessa suodatetusta signaalista diskreetti muunnos tuottaa kuitenkin pettymyksen. Käännetyn muunnoksen signaalissa pulssit muuttuvat aina lopulta kantikkaiksi, jolloin pulssin paikantaminen ei luonnollisesti onnistu (**Kuva 9.**). Tähän ongelmaan ei tuntunut löytyvän ratkaisua, joten disktreetistä muunnoksesta luovuttiin.



Kuva 9. Diskreetin wavelet-muunnoksen kynnystäminen

3 LÄHTEET

(1) Addison, P.S. 2002. The Illustrated Wavelet Transform Handbook. Taylor & Francis Group, LLC

(2) Mathiak Tobias and others, 2011, Methodology for automatic boundary layer detection using ground penetrating radar, Advanced Ground Penetrating Radar (IWAGPR), 2011, 6th International Workshop on, DOI: 10.1109/IWAGPR.2011.5963891

(3) Posio Jani, 2005, Mittaussarjoista saatavan tiedon kehittäminen - SCOAP-lämpötilaprofiilin esikäsittely, ISBN 951-42-7899-2 Oulun yliopisto START-projekti

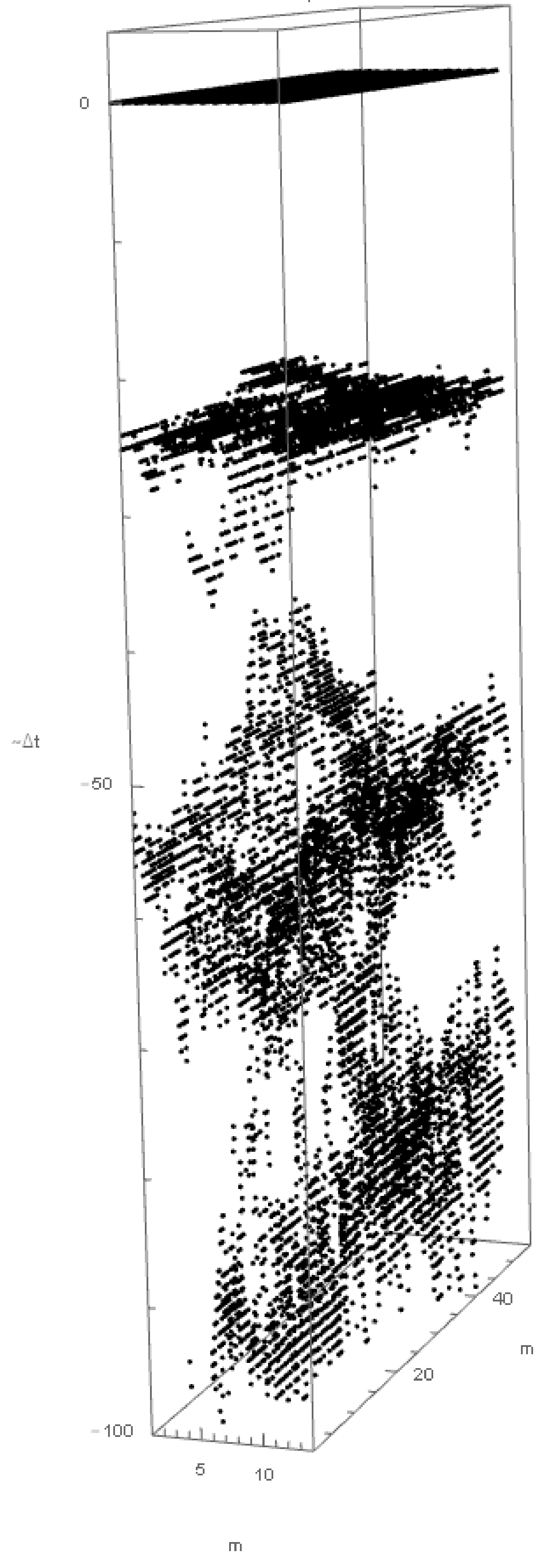
(4) Wolfram Mathematica 10.2.0.0, Built In Language Symbol Help

4 LIITTEET

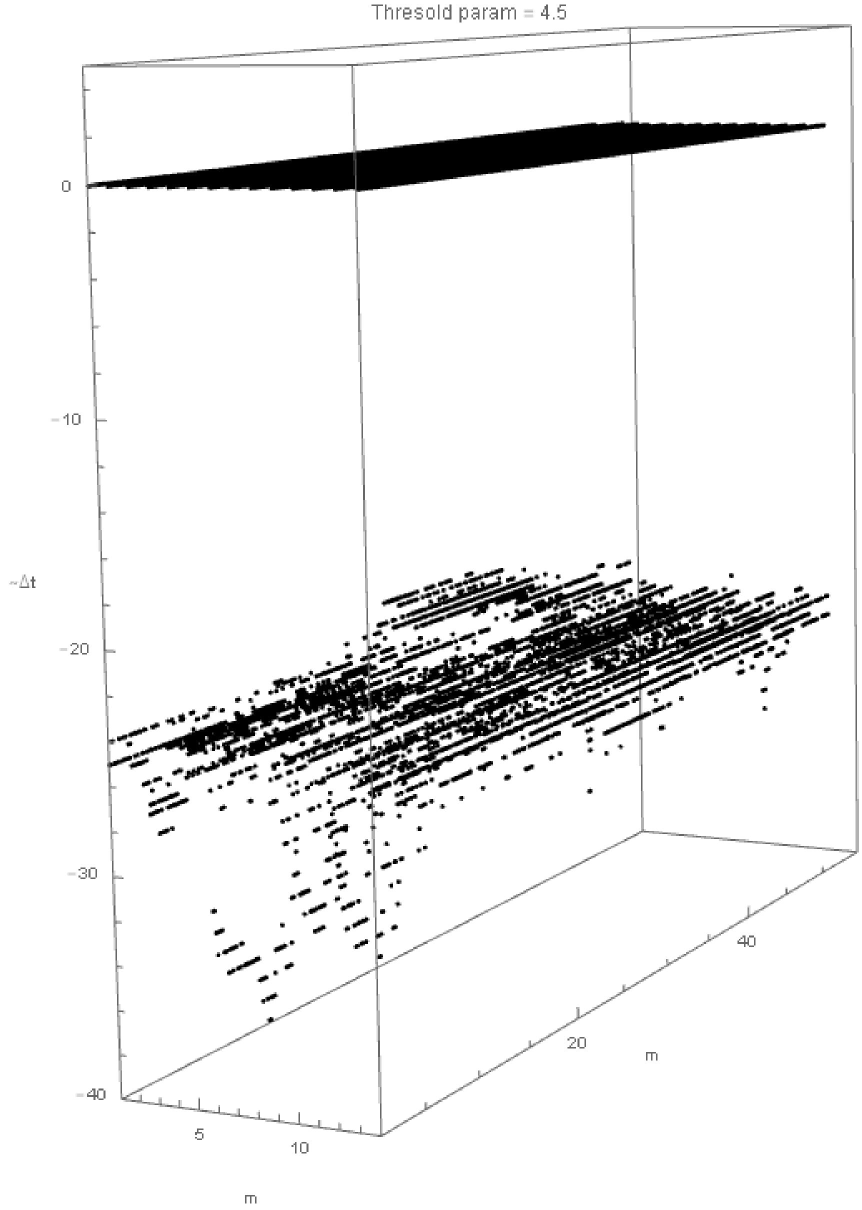
Vuonna 2016 menetelmää testattiin aidolla 3D datalla, joka oli skannattu Norvajärven varalaskupaikalta (asfalttipäällyste). Data muodostettiin useiden samalta kohdalta lähtevien rinnakkaisten ajolinjojen muodostamana yhdistettynä mittauksena. Menetelmää testattiin siten, että kaikki ajolinjat suodatettiin samalla periaatteella. Suodatetut skannaukset yhdistettiin 3D pistepilveksi siten, että leveyssuunnassa ajolinjojen väliksi asetettiin 1 m ja pituussuunnassa peräkkäisten pisteiden välimatkaksi 0.1 m. Korkeussuunnassa pulssien indeksietäisyys on jaettu 5:llä, jotta eri kerrokset mahtuisivat samaan kuvaan. Automaattiseen kuvausmenetelmään voidaan liittää myös kerroksien paksuuden arvioinnit, mikäli niiden dielektrisyysarvot voidaan määrittää.

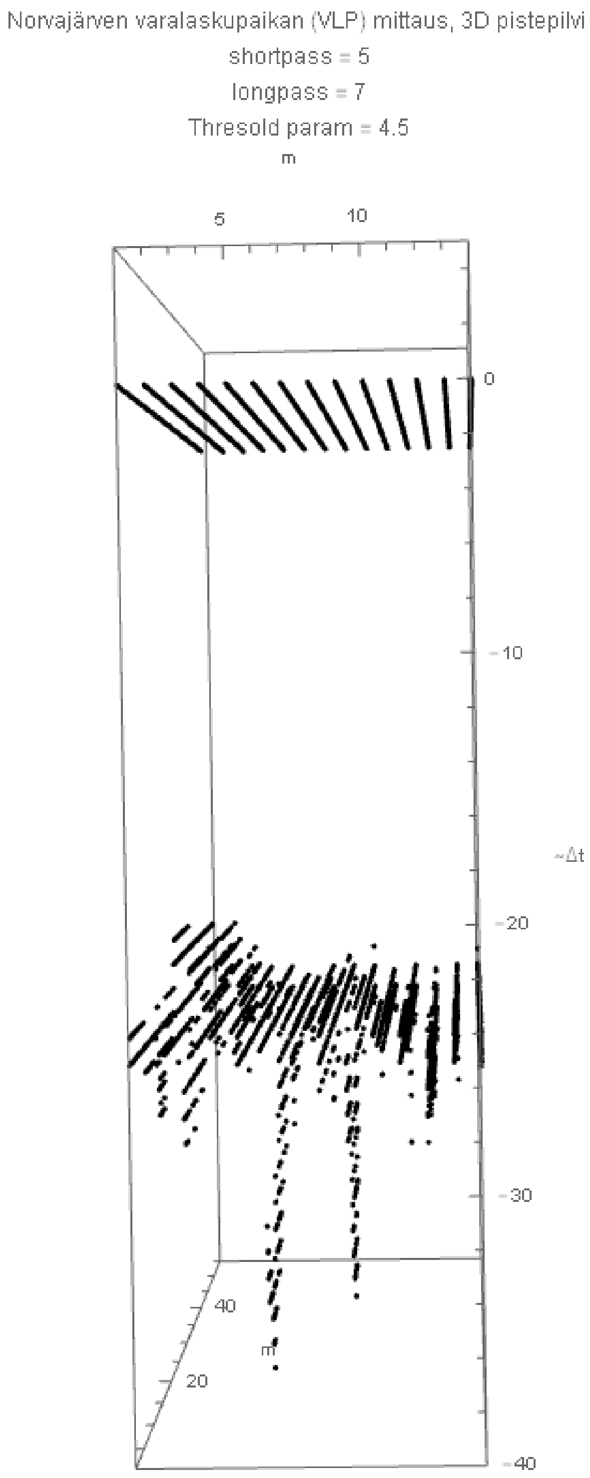
Kuvat (3 kpl) seuraavilla sivuilla

Norvajärven varalaskupaikan (VLP) mittaus, 3D pistepilvi
shortpass = 5
longpass = 7
Thresold param = 4.5



Norvajärven varalaskupaikan (VLP) mittaus, 3D pistepilvi
shortpass = 5
longpass = 7
Thresold param = 4.5





Norvajärven varalaskupaikan (VLP) mittaus, 3D pistepilvi
shortpass = 5
longpass = 7
Thresold param = 4.5
m
5
10
0
-10
~Δt
-20
-30
-40
40
m
20