

Pistepilvien visualisointi laitossuunnitteluohjelmistossa

Pro Gradu
Turun yliopisto
Tulevaisuuden teknologioiden laitos
Tietojenkäsittelytiede
2018
Timo Heinonen
Tarkastajat:
P.P.
H.H.

Turun yliopiston laatujärjestelmän mukaisesti tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck-järjestelmällä

TURUN YLIOPISTO

Tulevaisuden teknologioiden laitos

Timo Heinonen Pistepilvien visualisointi laitossuunnitteluoohjelmistossa

Pro Gradu, 18 s., 3 liites.

Tietojenkäsittelytiede

15. marraskuuta 2019

Turun yliopiston laatujärjestelmän mukaisesti tämän julkaisun alkuperäisyys on tarkastettu Turnitin Originality Check -järjestelmällä.

Juuh elikkäs gradutyötä...

Sisältö

1 Johdanto	1
1.1 Pistepilvet	1
1.2 Laserkeilaimet	4
1.3 Pistepilvidatan käsittey	5
1.4 Pistepilvien hyödyntäminen laitossuunnittelussa	8
2 Pistepilvien käsitteilyssä käytetyt tietorakenteet	9
2.1 Vaatimukset ja haasteet	9
2.2 Hierarkiset tietorakenteet	10
2.2.1 QSplat	10
2.2.2 Peräkkäispistepuut	12
2.2.3 Sisäkkäispistepuut	13
3 Laitossuunnittelusovellukseen optimoitu tietorakenne	17
4 Vertailu	17
5 Johtopäätökset	17

1 Johdanto

Tässä luvussa määritellään pistepilven käsite ja esitetään sille sovelluskohteita.

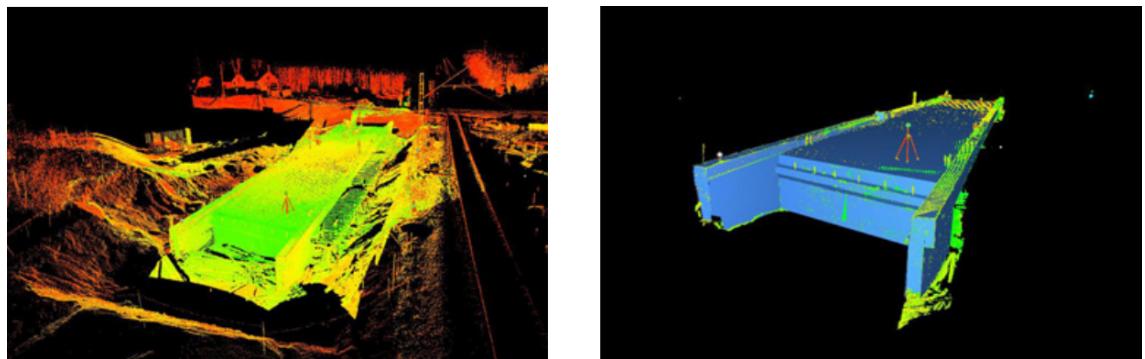
1.1 Pistepilvet

Pistepilveksi kutsutaan jotakin objektiota tai maisemaa kuvaavaa suurta joukkoa pisteitä kolmiulotteisessa avaruudessa. Pistepilvi tuotetaan yleensä laserkeilaimella (*engl. Laser Scanner*), joka ampoo ympärilleen laserpurskeita ja mittaa etäisyyksiä pisteisiin, joista purske heijastuu takaisin. Pistepilviä voidaan tuottaa myös synteettisesti ottamalla nätepisteitä mistä tahansa 3d-mallista, mutta tässä tutkielmassa keskitytään laserkeilaimilla tuottuihin pistepilviin.¹

Pistepilville on useita käyttökohteita, joista arkkitehtuuri ja rakentaminen on yksi tärkeimmistä. Pistepilvien hyödyntäminen voidaan aloittaa rakennusprojektiin hyvin varhaisessa vaiheessa. Rakennettavan tontin ympäristöstä voidaan ottaa laserkeilauksia, jotta suunniteltavan rakennuksen sopimista tontille voidaan helposti arvioida. Rakennusprojektiin aikana säädöllisillä laserkeilauksilla voidaan seurata tarkasti projektin etenemistä ja havaita mahdollisia ongelmia ajoissa. Pistepilvillä on myös tärkeä rooli rakennuksen valmistumisen jälkeen. Muutostöitä tehtäessä halutaan rakennuksesta saada ajantasalla oleva 3d-malli. Manuaalinen mallintaminen olisi hyvin suuri työ verrattuna muutaman kymmenen pistepilven luomiseen laserkeilaimella, mikä voidaan tehdä päivässä. [15]

Eräs mielenkiintoinen sovelluuskohde laserkeilaukselle on sillanrakentaminen. Sillanrakentamisessa haasteita tuottavat tien ja maaston geometrian yhteensovittamisen lisäksi projektin pitkä kesto. Silta on rakennettava osissa ja lopputuloksen on oltava suora, minkä takia siltatyömaalla suoritetaan usein tarkistusmittauksia. Älykäs silta -projektissa tutkitiin tapoja hyödyntää tietotekniikkaa sillanrakentamisessa ja -korjaamisessa, ja havaittiin

¹1980-luvulta lähtien pisteitä on ehdotettu yleisiksi renderöintiprimitiiveiksi kuvaamaan mitä tahansa geometriaa [12]. Aika kuitenkin näytti kolmion olevan suositumpi primitiivi ja nykyään grafiikkakirjastot ja -prosessorit onkin optimoitu kolmioiden käsittelyyn.



Kuva 1. Silta Joroisten ja Varkauden välissä Kuvasintiellä. Vasemmalla pistepilvi, oikealla pistepilvestä muodostettua geometriaa suunnitteluoohjelmassa. Kuva: [16]

laserkeilausen olevan hyvä tapa suorittaa tarkkuusmittauksia. Laserkeilaussella muodostettiin pistepilviä sillan kannesta ja rakenteista, jonka jälkeen niistä muodostettiin pintoja 3d-suunnitteluoohjelmaan. Pistepilvi ja siitä muodostettu geometria on esitetty kuvassa 1. [16]

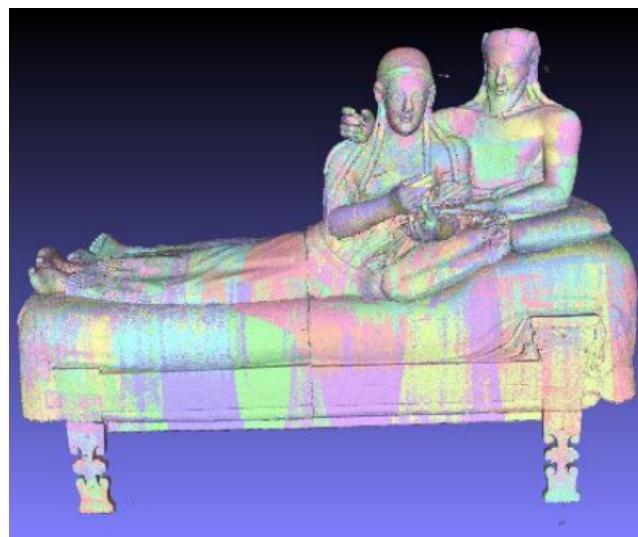
Pistepilviä käytetään hyväksi myös arkeologiassa. Roomalaiset perustivat vuoden 40 tienoilla Tonavan varrelle nykyisen Itävallan alueelle sotilasleirin, josta kasvoi myöhemmin Carnuntumin kaupunki. Kaupungin muurien ulkopuolelle rakennettiin amfiteatteri, johon mahtui 13000 katsojaa. Vuonna 2007 Ala-Itävallan osavaltion hallitus aloitti arkeologiset kaivaukset amfiteatterin alueella, joiden yhteydessä alueesta muodostettiin kattava pistepilvi noin kahdellasadalla laserkeilaussella. Ruudunkaappaus kysesestä pistepilvestä on esitetty kuvassa 2. [22]

Pistepilviä käytetään historiallisen kulttuuriperinnön säilyttämiseen myös pienemmäsä mittakaavassa. 1800-luvulla tehdyissä Cerveterin arkeologisissa kaivauksissa Italiassa löydettiin 500-luvulla terrakottasavesta tehty etruskilainen sarkofagi, joka kuvasi avio-paria rentoutumassa tuonpuoleisessa. Satoihin palasiin hajonnut sarkofagi restauroitiin vuonna 1893 ja se digitoitiin käyttämällä laserkeilausta ja fotogrammetriaa vuonna 2013 taidenäyttely varten. Sarkofagista keilattu pistepilvi on esitetty kuvassa 3. [13]

Eräs vaativa pistepilvien sovelluskohde on itseohjautuvat kulkuneuvot. Voidakseen navigoida liikenteessä itseohjautuva auto tarvitsee kameroiden ja ultraäänisensoreiden li-



Kuva 2. Carnuntumin kaupungin amfiteatteri taltioituna laserkeilauksella. Kuva: [20]



Kuva 3. Etruskilaisesta sarkofagista muodostettu pistepilvi. Kuva: [13]

säksi katollen laserkeilaimen, jolla voidaan tarkkailla auton etäisyyttä muihin tienkäytäjiin ja esteisiin. Itseohjautuvat autot ovat merkittävä tutkimuskohde myös pistepilven käsittelyn kannalta. Auton katolle asennettavan laserkeilaimen tulisi olla tarkka, nopea ja edullinen, ja sen tuottamaa pistedataa täytyy voida käsitellä reaalialjassa. [8]

Pistepilviä voidaan käyttää myös huomattavasti suuremmassa mittakaavassa. Maanmittauslaitos on kerännyt ilmasta käsin pistepilvidataa lähes koko Suomen maaperästä. Lentokoneesta keilattu pistepilvi on melko harva - puoli pistettä neliömetriä kohden -, mutta keilattavan kohteen laajuus tekee pilvistä valtavia. Pistepilviä on käytetty lähinnä metsävarojen kartoittamiseen, mutta Maanmittauslaitos aikoo ryhtyä keräämään pistedataa tarkemmillä laserkeilamilla myös rakennuksista. [11]

1.2 Laserkeilaimet

Perinteinen laserkeilain on jalustalla seisova laite, joka pyörii pystyakselinsa ympäri ampuen laserpurskeita ympärilleen tihein välein. Laserkeilain mittaa etäisyyksiä pisteisiin, joista laserpurske heijastuu takaisin keilaimeen ja muodostaa näistä pisteistä pistepilven. Heijastuksen voimakkuutta käytetään usein määräämään pisteelle väri.

Yleensä tarkasteltavaa kohdetta täytyy keilata useista eri suunnista, jotta saataisiin tarpeeksi kattava joukko pistepilviä. Kohteesta riippuen voidaan tarvita jopa satoja keilauksia. Pistepilvien sovittamista yhteen tarkastellaan luvussa 1.3.

Nykyaikaisella laserkeilaimella saadaan muodostettua hyvin tarkka ja tiheä pistepilvi nopeasti. Esimerkiksi kuvassa 4 näkyvä Leica Geosystems RTC360 -keilaimen luvataan mittaamaavan jopa kaksi miljoonaa pistettä sekunnissa ja kiertävän täyden ympyrän alle kahdessa minuutissa. Keilaimen lisäksi laitteessa on kamera, jolla saadaan määritettyä pisteille oikeat värit. [1]

Laserkeilaimien toimintaperiaatteissa on eroja. Kaksi yleisintä toimintaperiaatetta ovat kulkuaikateknikka (*engl. Time-Of-Flight*) ja vaihesiirtotekniikka (*engl. phase shift*). Kulkuaikateknikkassa pisteen etäisyys keilaimesta selviää ajasta, joka kuluu laserpurskeen



Kuva 4. Kulkuaikateknikaan perustuva Leica RTC360 -laserkeilain.

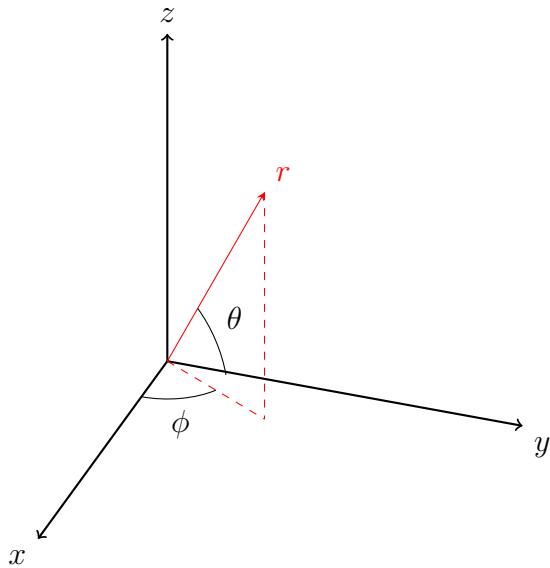
Kuva: https://leica-geosystems.com/-/media/images/leicageosystems/about-us/news%20room/reporter/reporter-83/09-discovering-the-power-of-scanning/leica-espresso_expert_insights_640x750_slider4.ashx?la=en&hash=9D35592CBB571C3777E1E8A9A0A056BD

lähetettämisestä sen heijastuksen vastaanottamiseen. Pisteen etäisyys keilaimesta on yksinkertaisesti $d = \frac{c \cdot t}{2}$, missä c on valonnopeus ja t on mitattu aika. [6] Vaihesiirtotekniikka perustuu keilaimesta lähtevän signaalin vaiheen vertaamista palaavan signaalin vaiheeseen. Pisteen etäisyys keilaimesta saadaan laskemalla $d = n \cdot \lambda + \frac{\Phi \cdot \lambda}{2 \cdot \pi}$, missä n on havainnon täysien aaltojen määrä, λ on signaalin aallonpituuus ja Φ on lähtevän ja palaavan signaalin vaihe-ero. [6]

Laserkeilain tallentaa mittaanansa pisteen pallokoordinaateissa. Pisteen siirtäminen pallokoordinaateista karteesiseen koordinaatistoon onnistuu laskemalla koordinaatit $x = r \cdot \sin \theta \cdot \cos \phi$, $y = r \cdot \sin \theta \cdot \sin \phi$, $z = r \cdot \cos \theta$, missä r on pallon säde, θ on korotuskulma ja ϕ atsimuuttikulma. Pallokoordinaatistoa on havainnollistettu kuvassa 5.

1.3 Pistepilvidatan käsittely

Laserkeilauksen tuottama pistepilvi sisältää joukon pisteytä koordinaatistossa, jonka origona on keilaimen sijainti. Usein keilauksen kohteesta otetaan kymmeniä tai jopa satoja



Kuva 5. Pallokoordinaatisto. Pisteen sijainti avaruudessa ilmaistaan korotuskulmalla θ , atsimuuttikulmalla ϕ ja säteellä r .

keilauksia, jotka täytyy saada samaan koordinaatistoon. Tätä kutsutaan pistepilvien rekisteröinniksi.

Pistepilvet voidaan rekisteroidä usealla eri tavalla. Joskus keilattavaan kohteeseen asetetaan erityisiä merkkikuvioita, jotka näkyvät useasta keilaimesta. Kun tiedetään merkkien etäisyyys ja suunta kustakin keilaimesta, voidaan pistepilvet sovittaa yhteen koordinaatistoon. Joissakin sovelluksissa käyttäjä merkitsee pilvistä pisteitä, jotka kuvaavat samaa aluetta.

Pistepilviä voidaan rekisteroidä myös ilman merkkikuvioita. Iteratiivinen lähimmän pisteen algoritmi (*engl. iterative closest point, ICP*) sovitaa pistepilven toiseen etsimällä rotaation ja translaation, jolla pilvien välinen virhe saadaan minimoitua. ICP-algoritmi määrittää ensin pilvistä toisiaan vastaavat pisteet ja virhe lasketaan kaikkien vastaavuksien välisistä etäisyyksistä. Yksinkertaisimillaan pistettä vastaavaksi pisteeksi merkitään sovitettavan pilven lähinnä sijaitseva piste. ICP-algoritmi tarvitsee käyttäjältä usein hyvän alkuarvauksen, jotta pilvien sovittaminen onnistuisi.²

Pistepilvissä on usein muukalaispisteitä (*engl. outlier*), johtuen esimerkiksi laserkei-

²Täysin automaattista pistepilvien rekisterointia on tutkittu paljon, ks. esim. [21].

laimen epätarkkuudesta tai vaikkapa tuulen heiluttamista puiden lehdistä. Yksinkertainen tekniikka poistaa muukalaispisteitä pilvestä on verrata pisteen normalivektoria sen naapuripisteiden normaaleihin. Pisteen p_i normalivektori voidaan selvittää pääkomponenttianalyysillä (*engl. principal component analysis, PCA*). Ensin on etsittävä pilvestä pisteen p_i k lähintä naapuria, jonka jälkeen naapuriston pistestä lasketaan ominaisarvot. Kahta suurinta ominaisarvoa vastaavaa ominaisvektoria voidaan käyttää kuvaamaan tasoa, joka sovitetaan naapuruston päälle. Jäljelle jäävä ominaisvektori kuvailee pisteen p_i normaalialia. [10]

Joissakin sovelluksissa halutaan luoda pistedataasta kohteen pintoja kuvaava polygoniverkko, jotta visualisointi olisi nopeampaa olemassaoleville grafiikkakirjastoille ja visuaalinen lopputulos parempi. Yksinkertainen tekniikka luoda tiivis kolmiointi pistepilvestä on Delaunayn kolmiointi³ (*engl. Delaunay triangulation*). Delaunayn kolmiointi perustuu Voronoin diagrammiin (*engl. Voronoi diagram*), joka jakaa pisteytä sisältävän tason tai avaruuden konvekseihin Voronoin soluihin. Voronoin solu kattaa sen alueen, jossa etäisyys solua vastaavaan pisteeseen on pienempi kuin muihin pistesiin. Kahden solun välillä on Voronoin jana, josta etäisyys kahteen pisteeseen on sama, ja kolmen janan leikkauspisteessä on Voronoin kärki, josta etäisyys kolmeen pisteeseen on yhtä suuri. Delaunayn kolmiointi on Voronoin diagrammin duaaligraafi, josta luodaan graafi siten, että pisteen välillä on kaari, mikäli niitä vastaavat Voronoin solut jakavat Voronoin janan. [7]

Joskus pistepilvet halutaan kompaktimpaan esitysmuotoon, jossa niiden käsittely on helpompaa. Pistepilvestä voidaan luoda panoramakuva projisoimalla laserkeilaimen ympäröimät pistet kaksiulotteiselle kuvatasolle.⁴ Panoramakuvan resoluutiosta riippuen pistepilvestä voidaan karsia huomattava määrä pisteytä, joiden koko olisi liian pieni projisointuna kuvatasolle. Panoramakuvaan on helppo soveltaa erilaisia kuvankäsittelyalgoritmeja, kuten piirteentunnistusta (*engl. feature detection*).

³Delaunayn kolmiointi sopii harvoin oikeilla laserkeilaimilla taltioituihin, häiriötä sisältäviin pistepilviin. Hyvän yleiskatsauksen realissemistä pinnanmuodostustekniikoista ovat julkaisseet esim. [3]

⁴Projektio vaikuttaa suuresti panoramakuvan laatuun. ks. esim [9]

1.4 Pistepilvien hyödyntäminen laitossuunnittelussa

Aiemmin mainittiin erilaisia sovelluksia laserkeilainten tuottamille pistepilville. Tässä tutkielmassa keskitytään pistepilvien hyödyntämiseen tietokoneavusteisessa suunnittelussa (*engl. computer aided design, CAD*) ja erityisesti laitossuunnitteluohjelmistoissa (*engl. plant design software*).

Tietokoneavusteisessa suunnittelussa pistepilviä käytetään olemassaolevien rakenteiden taltiointiin. Usein käytetty esimerkki on autoteollisuuden alalta: ryhmä suunnittelijoita kokeilee uutta korimallia rakentamalla prototyppiauton helposti muovattavasta materiaalista. Kun prototyppi on todettu aerodynaamiseksi ja miellyttävän näköiseksi, täytyy se saada digitoitua, jotta se voidaan siirtää massatuotantoon. Usein yksinkertaisin ja kustannustehokkain tapa on laserkeilata prototyppi ja jatkokäsitellä pistepilveä niin, että saadaan luotua haluttu 3d-malli.

Laitossuunnittelussa yleisempi ongelma on 3d-mallin vanhentuminen tai sen puuttuminen kokonaan. Vaikka laitosta alunperin suunniteltaessa siitä olisi tehty 3d-malli, laitteistojen sommittelua saatetaan muuttaa ilman, että samoja muutoksia tehdään 3d-malliin. Kun 3d-mallia halutaan taas hyödyntää, voi olla kustannustehokkaampaa luoda laitoksesta laserkeilaimella pistepilvi kuin mallintaa tehdyt muutokset suunnitteluohjelmalla. [14]

Pistepilviä voi hyödyntää monin tavoin laitossuunnittelussa. Vähiten töitä vaatii pilven käyttö sellaisenaan. Jos laitokseen halutaan vaikkapa uusi vesiputki, voidaan sen mahtuminen varmistaa reitittämällä putki suunnitteluohjelmistolla ja tarkastamalla, osuko se pistepilveen. Jos vanhasta laitoksesta halutaan luoda ajantasalla oleva 3d-malli, voi suunnittelija mallintaa laitosta pistepilvien avulla. Pistepilven päälle on helppo sijoittaa suunnitteluohjelman putkistoja ja laitteita oikeille paikoilleen. Markkinoilla on myös ohjelmistoja, joiden luvataan tuottavan pistepilvestä automaattisesti älykäs 3d-malli komponenttitietoineen ilman aikaavievää päälemallinnusta [2].

2 Pistepilvien käsitellyssä käytetyt tietorakenteet

Tässä luvussa selvitetään, mitä vaatimuksia pistepilvien käsitteily asettaa tietorakenteille ja visualisointialgoritmeille, tutustutaan aiheesta tehtyihin julkaisuihin ja esitetään laitos-suunnittelohjelmistolle optimoitu tietorakenne.

2.1 Vaatimukset ja haasteet

Suurin haaste pistepilvien käsitellyssä on niiden koko. Nykyaintelias laserkeilain, kuten luvussa 1.2 esitetty Leica Geosystemsin RTC360, tuottaa pistepilven, jossa on satoja miljoonia pisteitä. Kun tällaisella keilaimella tehdään useita keilausia, on pisteiden määrä valtava. Oletetaan esimerkiksi, että suressa projektissa käytetään pistepilviä, joissa on yhteensä miljardi pistettä. Kun koordinaatit tallennetaan kolmella nelitavuisella liukuluvuulla ja värit RGB-muodossa kolmella tavulla ja lisätään perään vielä yksi täytetavu, voidaan yksi pilven piste esittää 16:lla tavulla. Miljardin pisteen pilvi olisi siis kooltaan 16 gigatavua, joka saattaisi vielä mahtua tehokkaan työaseman keskusmuistiin, mutta ei grafiikkaprosessorin muistiin. Yleensä koko pilveä ei haluta pitää kerralla keskusmuistissa, vaan pisteitä haetaan levyltä muistiin ulkoisen muistin algoritmeilla (*engl. out-of-core algorithm*).

Pisteiden määrän vuoksi ei ole realistista olettaa, että kaikki pisteet voitaisiin visualisoida reaalialajassa. Pisteiden määrää voidaan karsia harventamalla pilveä esimerkiksi jaka-malla pilvi säännöllisiin kuutioihin, niin sanottuihin vokseleihin, (*engl. volume element, voxel*), ja näyttämällä vain yksi piste kustakin vokselista. Pilven harventaminen kuitenkin aiheuttaa yksityiskohtien katoamista pilvestä, joten sitä täytyy käyttää sovelluskohteesta riippuen maltillisesti.

Usein on hyväksyttävä, ettei pistepilveä saada visualisoitua reaalialajassa. Interaktiivisessa ohjelmistossa, kuten laitossuunnittelohjelmistossa, on tärkeää kuitenkin pitää ruudunpäivitystaajuus tarpeeksi korkeana. Tällöin voidaan pistepilvestä piirtää ruudulle ensin karkea yleiskuva, jota tarkennetaan vähitellen, ellei käyttäjä keskeytä visualisointia

esimerkiksi vaihtamalla kuvakulmaa. Tällainen astettainen visualisointi on mahdollista käyttämällä hierarkisia tietorakenteita pistepilven käsitellyssä. Tämän tekniikan etuna on se, että käyttäjä näkee välittömästi pistepilven yleisen muodon ilman, että hänen täytyy odottaa, että koko pilvi on visualisoitu. Jos käyttäjällä riittää kärsivällisyys, näkee hän kaikki pilven yksityiskohdat, kun visualisointialgoritmi on käynyt koko tietorakenteen läpi.

Tämän tutkielman osana kehitetään laitossuunnitteluoohjelmistolle optimoitu hierarkinen tietorakenne pistepilvien käsitellyyn. Esitetään tietorakenteelle seuraavat vaatimukset:

- On voitava visualisoida karkea yleiskuva pistepilvestä vain pienellä osalla datasta.
- Pistepilveä on voitava harventaa siten, että tarkkuuden menetystä voidaan kontrollida.
- On käytettävä ulkoisen muistin algoritmeja, eli koko pilveä ei pidetä kerralla keskusmuistissa.
- Hakuoperaatioiden tulee onnistua logaritmisessa ajassa pisteiden määrään suhteutettuna.
- Käyttäjän on voitava määrittää pilvestä alueita, joiden sisältävien pisteiden ominaisuuksia, kuten näkyvyyttä tai väriä, voidaan muuttaa.

2.2 Hierarkiset tietorakenteet

2.2.1 QSplat

Yksi ensimmäisistä pistedatan visualisointiin käytetyistä hierarkisista tietorakenteista on Rusinkiewiczin ja Levoyn esittämä QSplat, joka on kehitetty polygoniverkon visualisointiin pisteiden avulla. Tietorakenne muodostetaan kolmiodusta mallista, jossa kolmioi-

den normaalit tunnetaan, joten se ei suoraan sovella raa'an pistepilvidatan käsitellyyn.⁵ QSplatissa on käytetty kuitenkin monia kiinnostavia tekniikoita, joita voi hyödyntää pistepilvien käsitelyssä. [17]

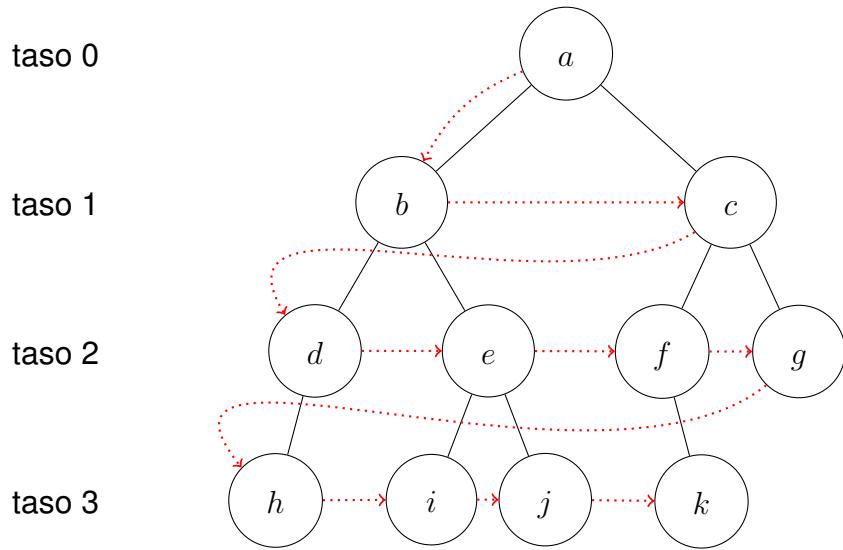
QSplat perustuu puurakenteeseen, jonka solmuissa on avaruutta rajaavia palloja (*engl. bounding sphere*). Pallot jakavat avaruutta rekursiivisesti pienempiin osiin siten, että juuren pallo sisältää kaikki kolmiot ja jokainen sisäinen solmu jakaa avaruuden keskimäärin neljään osaan. Puun latva saavutetaan, kun avaruuden jakamisen seurausena jäljelle jää yksi kolmio. Siitä muodostetaan lehtisolmu, jonka rajaava pallo sisältää koko kolmion. Puun visualointi onnistuu piirtämällä jokaisen pallon kohdalle sopivan kokoinen täplä (*engl. splat*). Puurakenne mahdollistaa myös tehokkaan pisteiden karsimisen. Jos solmun pallo ei ole näkökentässä, eivät sen lapsetkaan ole ja haaraa ei tarvitse käydä läpi. [17]

Puurakenne tallennetaan levylle leveysjärjestyksessä (*engl. breadth-first*). Tämän ansiosta puun tasot muodostavat luonnolliset tarkkuustasot (*engl. level-of-detail, LOD*): juurisolmun pallo esittää koko mallia, ensimmäinen taso sisältää muutaman pienemmän pallon, ja niin edelleen. Kun tällainen tiedoston sisäinen rakenne yhdistetään ulkoisen muisin teknikoihin, voidaan täplien piirtäminen aloittaa heti, kun tarpeeksi puun solmuja on ladattu levyltä muistiin.⁶ Puun rakennetta on havainnollistettu kuvassa 6. [17]

Toinen hyödyllinen QSplatissa käytetty tekniikka on koordinaattien kvantisointi (*engl. quantization*). Kun tarkkuudesta voidaan tinkiä, solmujen pallojen absoluuttisia koordinaatteja ei tallenneta, vaan niiden sijainti ilmaistaan suhteessa vanhempiinsa. Pallon säteen ja keskipisteen suhteellisen poikkeaman ilmaisemiseen käytetään vain 13:a arvoa. Pallon säde r voi olla välillä $[\frac{1}{13}, \frac{13}{13}]$ ja samaten keskipisteen suhteellisen poikkeaman x, y ja z -koordinaatit ovat vanhemman pallon läpimitan kolmastoistaosan monikertoja. Kun vielä hylätään vanhemman pallon ulkopuolella olevat keskipisteet ja käytetään hakutau-

⁵Itse asiassa Rusinkiewicz ja Levoy käyttivät laserkeilatusta pistepilvestä muodostettua kolmioverkkoa, jonka tietorakenne esitti yksinkertaisempina pistedatanana.

⁶Rusinkiewicz ja Levoy päättävät ruudulle projisoitujen täplien koon perusteella kuinka syvälle puussa tulee edetä.



Kuva 6. Puun läpikäyntijärjestys (punainen katkoviiva) muodostaa luonnolliset tarkkuus-tasot

lua, voidaan pallon sijainti esittää vain 13:lla bitillä, kun normaali liukulukuesitys vaatii vähintään 16 tavua. [17]

QSplat onnistui visualisoimaan 1,5-2,5 miljoonaa pistettä sekunnissa, mikä on sen aikaisella laitteistolla erinomainen tulos [17]. Kuten sanottu, se ei sellaisenaan kuitenkaan sovelli laserkeilattujen pistepilvien käsitteilyyn. Pistepilvien pisteiden normaaleja ei yleensä tiedetä, joten ne pitäisi esiprosessointivaiheessa selvittää esimerkiksi luvussa 1.3 esitettyllä tekniikalla. QSplat tarjoaa kuitenkin monia tekniikoita, joita pistepilviä käsittelevässä tietorakenteessa voidaan hyödyntää, kuten hierarkinen rakenne ja koordinaattien suhteellinen esitystapa.

2.2.2 Peräkkäispistepuut

Dachsbacher et al. esittelevät niin kututut peräkkäispistepuun (*engl. sequential point tree*), jossa pistet on aluksi järjestetty samaan tapaan pallopuuhun kuin QSplatissa. Pistet sijaitsevat puun lehtisolmuissa ja sisäsolmuissa säilytetään täpliä, jotka juuri ja juuri peittävät solmun lasten rajaavat pallot. Täplien väri määräytyy lapsisolmujen värien keskiarvolla. [4]

Jokaiseen puun solmuun on virhelaskelmien perusteella lisätty rajat katseluetäisydelle, jolla solmu valitaan visualisoitavaksi. Visualisointivaiheessa hierarkia litistetään tau-lukoksi, joka voidaan syöttää suoraan grafiikkaprosessorille. Grafiikkaprosessori käy tau-lukkoa läpi ja valikoi sopivat solmut, joiden perusteella täpliä piirretään ruudulle. [4]

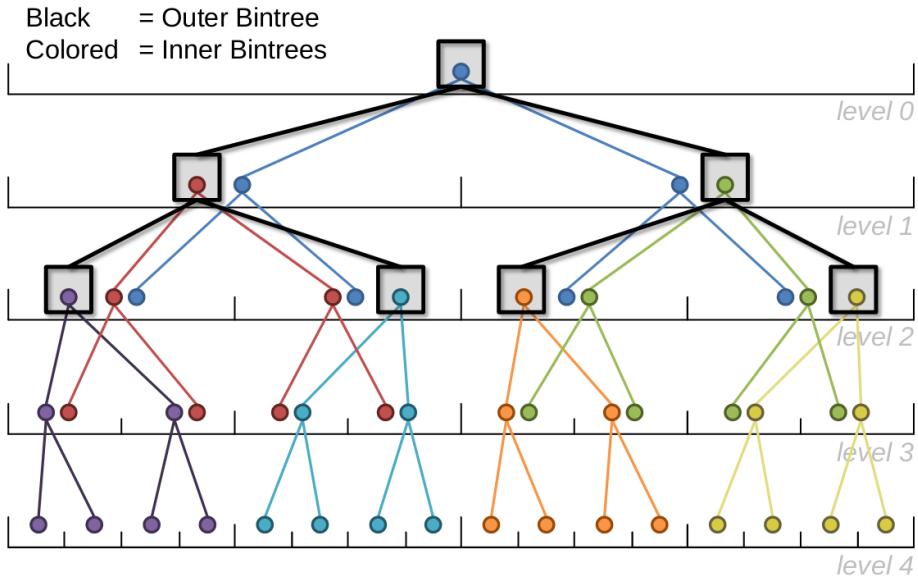
Peräkkäispistepuut voidaan visualisoida nopeasti grafiikkaprosessorin käytön ansios-ta, mutta niissä on myös heikkouksia. Tietorakenteen vaatimuksena on, että kaikki data mahtuu grafiikkaprosessorin muistiin. Näin on vain pienillä malleilla ja tilannetta pahen-taa se, että peräkkäispistepuut eivät ole kovin säästäväisiä muistin suhteen. Hierarkian jokaisessa sisäsolmussa luodaan lisää dataa, kun lapsisolmujen unionia kuvataan täplän sijainnilla ja koolla, sekä keskimääräisellä värellä.

Wimmer ja Scheiblauer esittävät parannuksia peräkkäispistepuihin. Uusien täplien luomisen sijaan puun sisäsolmuissa valitaan lapsisolmuista edustaja, joka parhaiten ku-vaa sisäolmun esittämää avaruuden osaa. Tämä on tärkeä huomio, sillä käsiteltäessä massiivisia pistepilviä tulisi välttää ylimääräisen datan luomista. Hierarkiasta muodoste-taan tarkkuustasot siten, että alempat tarkkuustasot sisältyvät ylempiin tasoihin ja visua-lisoitaessa oikea tarkkuustaso valitaan täplien koon ja katseluetäisyden perusteella. Sol-mua vastaavan täplän koko määrätyy siitä, kuinka syvällä hierarkiassa se on. Wimmer ja Scheiblauer kutsuvat tästä rakennetta muistioptimoiduksi peräkkäispistepuksi. (*engl. memory optimized sequential point tree, MOSPT*) [23]

2.2.3 Sisäkkäispistepuut

Wimmer ja Scheiblauer kritisoivat muistioptimoituja peräkkäispistepuita siitä, että ne ei-vät tue näkökentän ulkopuolisten pisteiden tehokasta karsimista ja siitä, ettei muistiopti-mointi yksinään riittänyt poistamaan tarvetta ulkoisen muistin algoritmeille. Ratkaisuksi he esittivät sisäkkäisiä oktettipuita (*engl. nested octree*). Oktettipuu⁷ on yksinkertainen avaruutta rekursiivisesti jakava tietorakenne, jonka jokainen sisäsolmu jakaa kuvaamansa

⁷Tätä suomennosta on käytetty esimerkiksi [5]



Kuva 7. Sisäkkäinen binääripuu, jossa sekä ulomman, että sisempien puiden syvyys on kolme. Ulompaa puuta kuvaavat mustat neliöt ja sisempää värikkät ympyrät. Puista muodostuu viisi tarkkuustasoa. Kuva: [19]

avaruuden osan kahdeksaan osaan. Wimmerin ja Scheiblauerin tietorakenteessa oktetti-puita on kahdessa tasossa. Ulompaa oktettipuuta käytetään avaruuden jakamiseen, sen tehokkaseen läpikäymiseen ja näkökentän ulkopuolisten alueiden karsimiseen. Ulomman puun jokainen solmu sisältää yhden sisemmän oktettipuun, joka vastaa samaa avaruuden osaa, kuin ulkoisen puun solmu. Pisteet sijoitetaan sisempiin puihin, yksi jokaiseen solmuun. [23]

Sisäkkäisistä oktettipuista luodaan tarkkuustasot siten, että sisemmistä puista kerätään pisteitä ulomman puun tasojen mukaan. Tarkkuustasoon kuuluvat pisteet sijaitsevat siis ulomman puun samalla tasolla, mutta useiden sisempien puiden eri tasolla. Tarkkuustasojen muodostumista on havainnollistettu kuvassa 2.2.3. Puut tallennetaan levylle tarkkuustaso kerrallaan, mikä mahdollistaa ulkoisen muistin algoritmien käytön. Visualisoitaessa tarvitsee levyltä lukea pisteitä vain haluttuun tarkkuustasoon asti, eikä loppuja pisteitä tarvitse ladata muistiin. [23]

Scheiblauer jalostaa sisäkkäisten oktettipuiden ideaa väitöskirjassaan esittelemällä muokkavat sisäkkäiset oktettipuut (*engl. modifiable nested octree, MNO*). Jos edellä esitel-

tyjä sisäkkäisiä oktettipuita halutaan muokata rakentamisen jälkeen, on sisemmät puut rakennettava ja muokattu hierarkia tallennettava levylle uudestaan. Nimensä mukaisesti MNO mahdollistaa tehokkaan pisteiden lisäämisen ja poistamisen. [19]

MNO:n rakenne eroaa sisäkkäisistä oktettipuista siten, että sisemmät puut korvataan säädöllisillä, kolmiulotteisilla ruudukoilla, joihin pisteet tallennetaan. MNO:n rakentaminen alkaa juurisolmusta, joka vastaa kaikki pisteet peittävää avaruutta. Solmun sisältämä ruudukko jakaa solmua kuvaavan avaruuden osan $128^3 = 2097152$ soluun. Pisteitä lisätään puuhun yksi kerrallaan niin, että jokaiseen ruudukon soluun mahtuu vain yksi piste. Jos solu on varattu, sijoitetaan piste ylimääräiseen taulukkoon odottamaan, että vastaavia pisteitä kertyy tarpeeksi, jotta olisi järkevä luoda uusia solmuja puuhun. Kun ennaltaehkästävästi vähimmäismäärä pisteitä on kertynyt ylimääräisten pisteiden taulukkoon, luodaan ruudukon sisältävälle solulle lapsisolmuja ja sijoitetaan ylimääräiset pisteet niihin. Ruudukkoon sijoitettavien pisteiden määrälle on hyvä asettaa myös yläraja. [19]

Jokainen tietorakenteen solmu tallennetaan omaan tiedostoonsa levylle, josta niitä ladataan muistiin visualisointivaiheessa tarvittaessa. Visualisointialgoritmien kuuluu käyttäjän asettama pistebudjetti, joka asettaa ylärajan yhdessä ruudunpäivityksessä piirrettävien pisteiden määrälle.⁸ Tätä rajaa säätmällä käyttäjä saa jonkinlaisen kontrollin ruudunpäivitystaajuuden suhteeseen. [19]

Tiedostorakenne mahdollistaa hierarkian yksinkertaisen muokkaamisen. Lisättäessä uusia pisteitä MNO:hon tarkastetaan ensin, sijoittuuko se juurisolmun kuvaamaan avaruuden osaan. Jos näin on, onnistuu lisääminen kuten rakennusvaiheessa. Muussa tapauksessa juurisolmulle luodaan vanhemmia kunnes jokin niistä muodostaa tarvittavan kokoinen avaruuden, ja piste lisätään sen ruudukkoon. Kun puun vanhan juuren yläpuolelle luodaan uusia solmuja, jää niiden ruudukot vajaaksi. Tällöin alemmista solmuista nostetaan pisteitä ylöspäin niin kauan, kunnes vajaita ruudukoita on vain lehtisolmuissa. Pisteiden poistaminen puusta on triviaalia, kun sisäsolmuihin mahdollisesti jäävät tyhjät ruudukot

⁸Scheiblauer testasi pistepilvivisualisoijaansa asettamalla rajan vain sataantuhanteen pisteeseen.

täytetään kuten pisteitä lisättääessa. [19]

Markus Schütz jatkoi Wimmerin ja Scheiblauerin työtä esittelemällä opinnäytetyös-sään verkkoselaimessa ajettavan Potree-nimisen pistepilvivisualisoijan. Potreen käyttämä tietorakenne perustuu Scheiblauerin muokkattaviin sisäkkäisiin oktettipuihin, mutta hierarkian rakennusvaiheessa kiinnitetään huomiota pisteiden tasaiseen jakautumiseen solmu-jen välille. Oktettipuun sisäsolmujen ruudukoihin hyväksytään uusia pisteitä vain, jos ne ovat tarpeeksi kaukana muista ruudukon pisteistä. Lehtisolmut hyväksyvät ennaltamää-rätyyn rajaan saakka kaikki pisteet, kunnes ne muutetaan sisäsolmuiksi ja liian lähekkäin olevat pisteet jaetaan uusien lapsisolmujen kesken. [18]

Potree käyttää ulkoista muistia tehokkaasti ja pystyy käsittelemään jopa 640 miljardia pistettä sisältäviä pistepilviä.⁹ Rakennusvaiheessa oktettipuun solmuja tallennetaan tasai-sin väliajoin levylle, jottei muisti täytyisi. Kun jokainen solmu tallennetaan omaan tie-dostoonsa, on yksittäisten solmujen tallentaminen ja lukeminen levyltä helppoa. Massii-visia pistepilviä kuvaavat hierarkiatkin voivat olla satojen megatavujen kokoisia. Schütz ratkaisee suurten hierarkioiden nopean lataamisen verkon yli jakamalla senkin puuraken-teeseen. Nämä voidaan välttää sekä turhien pisteiden, että näkökentän ulkopuolella olevien hierarkian haarojen lataaminen muistiin. [18]

Potreen visualisointialgoritmi priorisoi niitä hierarkian solmuja, jotka ovat lähellä kat-selupistettä ja joiden kuvaruudulle projisoitu koko on suuri. Visualisoinnin suoritusky-yä voidaan säädellä Scheiblauerin toteutuksen mukaisesti käyttäjän asettamalla piste-budgetilla. Schütz on kehittänyt Potreehen myös hienostuneen, grafiikkaprosessorilla ajet-tavan algoritmin mukautuvaan pisteiden koon määrittämiseen; pistepilven harvemmissa osissa piirretään pisteet suurempina, jottei reikiä esiintyisi. [18]

⁹Kyseinen pistepilvi (Actueel Hoogtebestand Nederland, ANH2, <http://ahn2.pointclouds.nl/>) kuvailee koko Alankomaiden valtiota ja se vaatii 7,68 teratavua tallennustilaan. Potreen tietorakenteessa pistepilvi jakautui 13:lle tasolle ja 38:aan miljoonaan solmuun.

3 Laitossuunnittelusovellukseen optimoitu tietorakenne

4 Vertailu

5 Johtopäätökset

Viitteet

- [1] Leica Geosystems AG, 2018.
- [2] Aveva. Aveva laser modeller.
- [3] Matthew Berger, Andrea Tagliasacchi, Lee Seversky, Pierre Alliez, Gaël Guennebaud, Joshua Levine, Andrei Sharf, and Cláudio Silva. A survey of surface reconstruction from point clouds. *Computer Graphics Forum*, 03 2016.
- [4] Carsten Dachsbacher, Christian Vogelsgang, and Marc Stamminger. Sequential point trees. *ACM Transactions on Graphics*, 22:657, 07 2003.
- [5] Jonne Davidsson. Technical report, 3point Oy, Lapinlahdenkatu 16 00180 Helsinki, 1 2019.
- [6] Jere Fabritius. Terrestrial three-dimensional laser scanning in aveva pdms. Opinäytettyö, Tampereen ammattikorkeakoulu, 2009.
- [7] Joachim Giesen and Frederic Cazals. Delaunay triangulation based surface reconstruction: Ideas and algorithms. 2006.
- [8] Jeff Hecht. Lidar for self-driving cars. *Optics and Photonics News*, 29(1):26–33, Jan 2018.
- [9] Hamidreza Houshiar, Jan Elseberg, Dorit Borrmann, and Andreas Nuchter. A study of projections for key point based registration of panoramic terrestrial 3d laser scan. *Geo-spatial Information Science*, 18, 03 2015.
- [10] Chien Ming Huang and Yi-Hsing Tseng. Plane fitting methods of lidar point cloud. In *29th Asian Conference on Remote Sensing 2008, ACRS 2008*, 29th Asian Conference on Remote Sensing 2008, ACRS 2008, pages 1925–1930, 12 2008.
- [11] Jarmo Huhtanen. *Helsingin sanomat*.
- [12] Turner Whitted Marc Levoy. The use of points as a display primitive. Technical report, Computer Science Department, University of North Carolina at Chapel Hill, 1985.

- [13] Fabio Menna, Erica Nocerino, Fabio Remondino, Matteo Dellepiane, Marco Callieri, and Roberto Scopigno. 3d digitization of an heritage masterpiece - a critical analysis on quality assessment. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B5:675–683, 06 2016.
- [14] Markus Piipponen. 3d-suunnittelun hyödyntäminen tehdassuunnittelussa. Opinnäytetyö, Satakunnan ammattikorkeakoulu, 2012.
- [15] Tan Qu and Wei Sun. *Journal of Civil Engineering and Architecture*, 9:1269–1278, 2015.
- [16] Heikkilä Rauno, Karjalainen Antti, Pulkkinen Pekka, Haapa aho Esa, Jokinen Mauko, Oinonen Aarno, and Jaakkola Mika. Siltojen 3d-suunnittelua- ja mittausprosessin kehittäminen ja käyttöönottaminen (Älykäs silta). Technical report, Tiehallinto, 2005.
- [17] Szymon Rusinkiewicz and Marc Levoy. Qsplat: A multiresolution point rendering system for large meshes. *Proceedings of SIGGRAPH*, 2000, 10 2001.
- [18] Markus Sch Potree: Rendering large point clouds in web browsers. Master's thesis.
- [19] Claus Scheiblauer. *Interactions with Gigantic Point Clouds*. PhD thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, 2014.
- [20] Claus Scheiblauer and Michael Pre gesbauer. Consolidated visualization of enormous 3d scan point clouds with scanopy. In *Proceedings of the 16th International Conference on Cultural Heritage and New Technologies*, pages 242–247, November 2011.
- [21] Pascal Willy Theiler. *Automated Registration of Terrestrial Laser Scanner Point Clouds*. PhD thesis, ETH Zürich, 2015.
- [22] Verkkolähde.
- [23] Michael Wimmer and Claus Scheiblauer. Instant points: Fast rendering of unprocessed point clouds. In *Proceedings Symposium on Point-Based Graphics 2006*, pages 129–136. Eurographics, Eurographics Association, July 2006.