

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, 25 s., 3 liites.

Tietojenkäsittelytiede

24. 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 ja laserkeilaimet	1
1.2 Pistepilvidatan käsittey	6
1.3 Pistepilvien hyödyntäminen laitossuunnittelussa	8
1.4 Tutkielman tavoitteet ja rakenne	9
2 Pistepilvien käsitellyssä käytetyt tietorakenteet	10
2.1 Pistepilvien käsitellyn haasteet	10
2.2 Hierarkiset tietorakenteet	11
3 Laitossuunnitteluohjelmistoon optimoitu tietorakenne	17
3.1 Käyttötapaukset ja vaatimukset tietorakenteelle	18
3.2 Tietorakenteen valinta	20
3.3 Ruudukko	22
3.4 Pisteiden valinta	23
3.5 Visualisointi	23
4 Arviointi	23
4.1 Tietorakenteen rakentaminen ja lataaminen	23
4.2 Visualisointi	23
4.3 Pisteiden valitseminen	23
4.4 Jotain muuta?	23
4.5 Jotain muuta?	23
5 Johtopäätökset	23
Viitteen	23

1 Johdanto

Tässä luvussa määritellään pistepilven käsite, tutustutaan laserkeilaimiin, joilla pistepilviä taltioidaan ja mainitaan muutamia yleisiä pistepilvien käsittelyn työvaiheita. Lisäksi esitellään muutama esimerkki pistepilvien käyttökohteista. Lopuksi määritellään tämän tutkielman tavoitteet, toteutustapa ja rakenne.

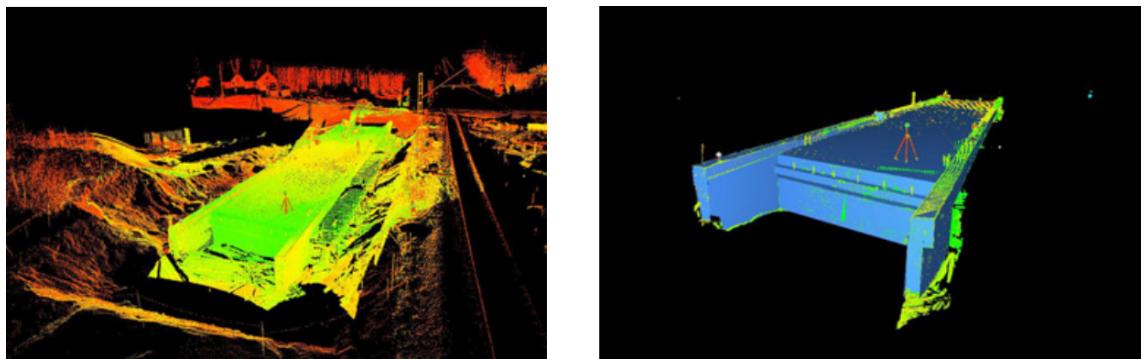
1.1 Pistepilvet ja laserkeilaimet

Pistepilveksi kutsutaan jotakin objektiota tai maisemaa kuvaavaa suurta joukkoa pisteitä kolmiulotteisessa avaruudessa. Pistepilvi tuotetaan yleensä laserkeilaimella (*engl. Laser Scanner*), joka ampuu ympärilleen laserpurskeita ja mittaa etäisyyksiä pisteisiin, joista purske heijastuu takaisin. Pistepilviä voidaan tuottaa myös synteettisesti ottamalla näytepisteitä 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 laserkeilauska, jotta suunniteltavan rakennuksen sopimista tontille voidaan helposti arvioida. Rakennusprojektiin aikana säädöllisillä laserkeilauskilla voidaan seurata tarkasti projektin etenemistä ja havaita mahdollisia ongelmia ajoissa. Pistepilvillä on myös tärkeä rooli rakennuksen valmistumisen jälkeen. Muutostötä tehtäessä halutaan rakennuksesta saada ajantasalla oleva 3d-malli. Manuaalinen mallintaminen olisi hyvin suuri työ verrattuna muutaman kymmenen pistepilven luomiseen laserkeilaimella, mikä voidaan tehdä päivässä. [2]

Eräs mielenkiintoinen sovelluskohde laserkeilauskelle on sillanrakentaminen. Sillanrakentamisessa haasteita tuottavat tien ja maaston geometrian yhteensovittamisen lisäksi

¹1980-luvulta lähtien pisteitä on ehdotettu yleisiksi renderöintiprimitiiveiksi kuvaamaan mitä tahansa geometriaa [1]. Aika kuitenkin näytti kolmion olevan suosituimpia primitiivi ja nykyään grafiikkakirjastot ja -prosessorit onkin optimoitu kolmioiden käsittelyyn.



Kuva 1. Silta Joroisten ja Varkauden välissä Kuvasintiellä. Vasemmalla pistepilvi, oikealla pistepilvestä muodostetua geometriaa suunnitteluoohjelmassa. Kuva: [3]

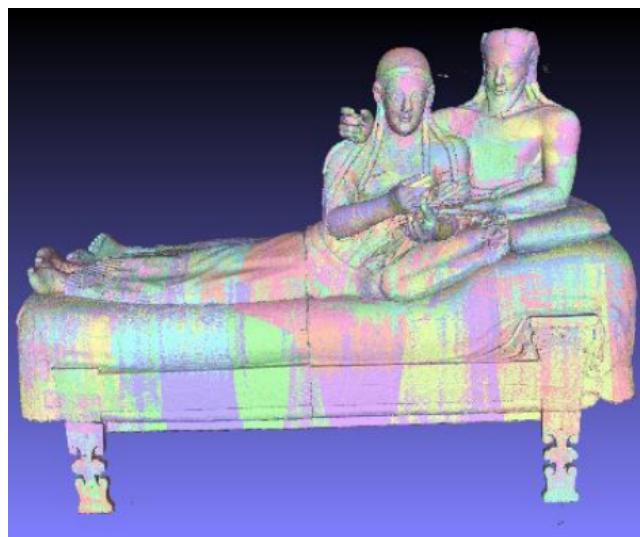
projektiin pitkä kesto. Silta on rakennettava osissa ja lopputuloksen on oltava suora, minkä takia siltatyömaalla suoritetaan usein tarkistusmittauksia. Älykäs silta -projektissa tutkittiin tapoja hyödyntää tietotekniikkaa sillanrakentamisessa ja -korjaamisessa, ja havaittiin laserkeilauksen olevan hyvä tapa suorittaa tarkkuusmittauksia. Laserkeilauksella muodostettiin pistepilviä sillan kannesta ja rakenteista, jonka jälkeen niistä muodostettiin pintoja 3d-suunnitteluoohjelmaan. Pistepilvi ja siitä muodostettu geometria on esitetty kuvassa 1. [3]

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 laserkeilauksella. Ruudunkaappaus kysestä pistepilvestä on esitetty kuvassa 2. [5]

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



Kuva 2. Carnuntumin kaupungin amfiteatteri taltioituna laserkeilauksella. Kuva: [4]



Kuva 3. Etruskilaisesta sarkofagista muodostettu pistepilvi. Kuva: [6]

taidenäytelyä varten. Sarkofagista keilattu pistepilvi on esitetty kuvassa 3. [6]

Eräs vaativa pistepilvien sovelluskohde on itseohjautuvat kulkuneuvot. Voidakseen navigoida liikenteessä itseohjautuva auto tarvitsee kameroiden ja ultraäänisensoreiden lisäksi katollen laserkeilaimen, jolla voidaan tarkkailla auton etäisyyttä muihin tienkäyttäjiin ja esteisiin. Itseohjautuvat autot ovat merkittävä tutkimuskohde myös pistepilvien käsitelyn kannalta. Auton katolle asennettavan laserkeilaimen tulisi olla tarkka, nopea ja edullinen, ja sen tuottamaa pistedataa täytyy voida käsitellä reaalialjassa. [7]

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 tarkemmilla laserkeilamilla myös rakennuksista. [8]

Perinteinen pistepilvien mittaamiseen käytetty 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 tarpeksi kattava joukko pistepilviä. Kohteesta riippuen voidaan tarvita jopa satoja keilauksia. Pistepilvien sovittamista yhteen tarkastellaan luvussa 1.2.

Nykyenkäisellä laserkeilaimella saadaan muodostettua hyvin tarkka ja tiheä pistepilvi nopeasti. Esimerkiksi kuvassa 4 näkyvä Leica Geosystems RTC360 -keilaimen luvataan mittamaavan jopa kaksi miljoonaa pistettä sekunnissa ja kiertävän täyden ympyrän alle kahdessa minuutissa. Keilaimen lisäksi laitteessa on kamera, jolla saadaan määritettyä pisteille oikeat värit. [9]

Laserkeilaimien toimintaperiaatteissa on eroja. Kaksi yleisintä toimintaperiaatetta ovat

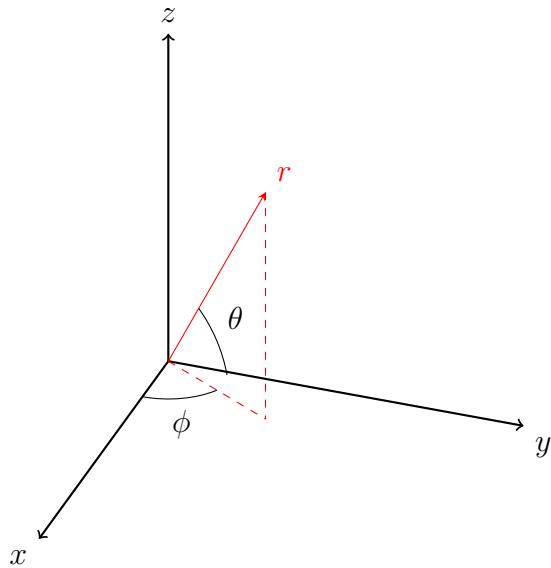


Kuva 4. Kulkuaikateknikaan perustuva Leica RTC360 -laserkeilain.

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

kulkuaikateknikka (*engl. Time-Of-Flight*) ja vaihesiirtotekniikka (*engl. phase shift*). Kulkuaikateknikkassa pisteen etäisyys keilaimesta selviää ajasta, joka kuluu laserpurskeen lähetettämisestä sen heijastuksen vastaanottamiseen. Pisteen etäisyys keilaimesta on yksinkertaisesti $d = \frac{c \cdot t}{2}$, missä c on valonnopeus ja t on mitattu aika. [10] 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. [10]

Laserkeilain tallentaa mittaan 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. Pallokoordinaatista on havainnollistettu kuvassa 5.



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

1.2 Pistepilvidatan käsittely

Laserkeilauksen tuottama pistepilvi sisältää joukon pisteitä koordinaatistossa, jonka origona on keilaimen sijainti. Usein keilauksen kohteesta otetaan kymmeniä tai jopa satoja keilauksia, jotka täytyy saada samaan koordinaatistoon. Tätä kutsutaan pistepilvien rekisteröinniksi.

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

Pistepilviä voidaan rekisteröidä 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 vastavuksien välisistä etäisyyksistä. Yksinkertaisimmillaan 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 laserkeilaimen 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 kuvaa pisteen p_i normaalialia. [12]

Joissakin sovelluksissa halutaan luoda pistedataasta kohteen pintoja kuvaava polygoniverkko, jotta visualisointi olisi nopeampaa olemassaolevilla grafiikkakirjastoilla 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. [14]

Joskus pistepilvet halutaan kompaktimpaan esitysmuotoon, jossa niiden käsittely on helpompaa. Pistepilvestä voidaan luoda panoramakuva projisoimalla laserkeilaimen ympäröimät pistet kaksiulotteiselle kuvatasolle.⁴ Panoramakuvan resoluutiosta riippuen pis-

²Täysin automaattista pistepilvien rekisteröintiä on tutkittu paljon, ks. esim. [11].

³Delaunayn kolmiointi sopii harvoin oikeilla laserkeilaimilla taltioituihin, häiriötä sisältäviin pistepilviin. Hyvän yleiskatsauksen realistisemmista pinnanmuodostustekniikoista ovat julkaisseet esim. [13]

⁴Projektio vaikuttaa suuresti panoramakuvan laatuun. ks. esim [15]

tepilvestä voidaan karsia huomattava määrä pisteitä, joiden koko olisi liian pieni projisointuna kuvatasolle. Panoramakuvaan on helppo soveltaa erilaisia kuvankäsittelyalgoritmeja, kuten piirteentunnistusta (*engl. feature detection*).

1.3 Pistepilvien hyödyntäminen laitossuunnittelussa

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

Tietokoneavusteisessa suunnittelussa pistepilviä käytetään olemassaolevien rakenteiden taltiointiin. Usein käytetty esimerkki on autoteollisuuden alalta: ryhmä suunnittelijoita kokeilee uutta korimallia rakentamalla prototyypiauton 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 suunnitteluoohjelmalla. [16]

Kuten sillanrakennuksessa, myös laitoksia rakentaessa on joskus syytä suorittaa tarkeusmittauksia ja verrata niitä alkuperäiseen 3d-malliin. Pistepilvet ovat tähänkin tarkeukseen sopivia. Nykyaikaisten laserkeilainten tuottamat pistepilvet ovat niin tarkkoja, että niistä voi havaita esimerkiksi putkien roikkumisen ja lämpölaajenemisen [16].

Pistepilviä voi hyödyntää monin tavoin laitossuunnittelussa. Vähiten töitä vaatii pilven käyttö sellaisenaan. Jos laitokseen halutaan vaikkapa uusi vesiputki, voidaan sen

mahtuminen varmistaa reitittämällä putki suunnitteluoohjelmistolla ja tarkastamalla, osuu-ko se pistepilveen. Jos vanhasta laitoksesta halutaan luoda ajantasalla oleva 3d-malli, voi suunnittelija mallintaa laitosta pistepilvien avulla. Pistepilven päälle on helppo sijoittaa suunnitteluoohjelman putkistoja ja laitteita oikeille paikoilleen. Markkinoilla on myös oh-jelmistoja, joiden luvataan tuottavan pistepilvestä automaattisesti älykäs 3d-malli kompo-nenttietoineen ilman aikaavievää päälemallinnusta [17].

Jotakin yleisiä käyttötapauksia pistepilvien kanssa työskentelystä on esitetty luvussa 3.1.

1.4 Tutkielman tavoitteet ja rakenne

Tämän tutkielman tarkoituksesta on tutustua pistepilvien käsittelyn ja visualisoinnin tuot-tamiin ongelmiin ja selvittää niihin ratkaisuvaihtoehtoja alan julkaisujen pohjalta. Sel-vitystyön pohjalta kehitetään tietorakenne erityisesti laitossuunnitteluoohjelmiston tarpei-siin. Lopuksi arvioidaan tietorakenteen suorituskykyä ja soveltuvuutta tehtävään käyttäen verrokkina yksinkertaista ei-hierarkista tietorakennetta.

Tässä luvussa on perehdytty pistepilviin ja laserkeilaimiin yleisellä tasolla. Luvussa 2 tutustutaan alan kirjallisuuteen ja esitellään tietorakenteita, joiden ominaisuuksia voisi hyödyntää myös laitossuunnitteluoohjelmistossa. Luvussa 3 määritellään vaatimuksia, joi-ta laitossuunnitteluoohjelmisto asettaa pistepilvien käsittelyyn ja visualointiin käytetylle tietorakenteelle. Tämän jälkeen ehdotetaan luvussa 2 esiteltyihin julkaisuihin perustuva, laitossuunnitteluoohjelmistoon soveltuva tietorakenne. Tätä tietorakennetta verrataan yk-sinkertaiseen toteutukseen luvussa 4.

Tämä tutkielma on tehty CADMATIC Oy:n toimeksiantona. CADMATIC on suoma-lainen ohjelmistoyritys, joka kehittää tuotteita laivojen ja laitosten tietokoneavusteiseen suunnittelun. CADMATIC on toiminut alalla 1980-luvulta lähtien ja sillä on asiakkai-naan yli tuhat organisaatiota yli viidestäkymmenestä maasta [18]. Yrityksen pääkonttori on Turussa. Tutkielmassa kehitettävä tietorakennetta testataan CADMATIC Plant Mo-

deller -laitossuunnitteluoohjelmistossa, sekä eBrowser-mallinkatseluohjelmistossa.

2 Pistepilvien käsitellyssä käytetyt tietorakenteet

Tässä luvussa selvitetään, mitä haasteita pistepilvien käsitteily asettaa tietorakenteille ja visualisointialgoritmeille. Lisäksi tutustutaan haasteisiin vastaaviin hierarkisista tietorakenteista kirjoitettuihin julkaisuihin.

2.1 Pistepilvien käsitelyn haasteet

Suurin haaste pistepilvien käsitellyssä on niiden koko. Nykyainkainen laserkeilain, kuten luvussa 1.1 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 muiston 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 matallisesti.

Usein on hyväksyttävä, ettei pistepilveä saada visualisoitua reaalialajassa. Interaktiivi-

sessa ohjelmistossa 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 käytämällä 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.

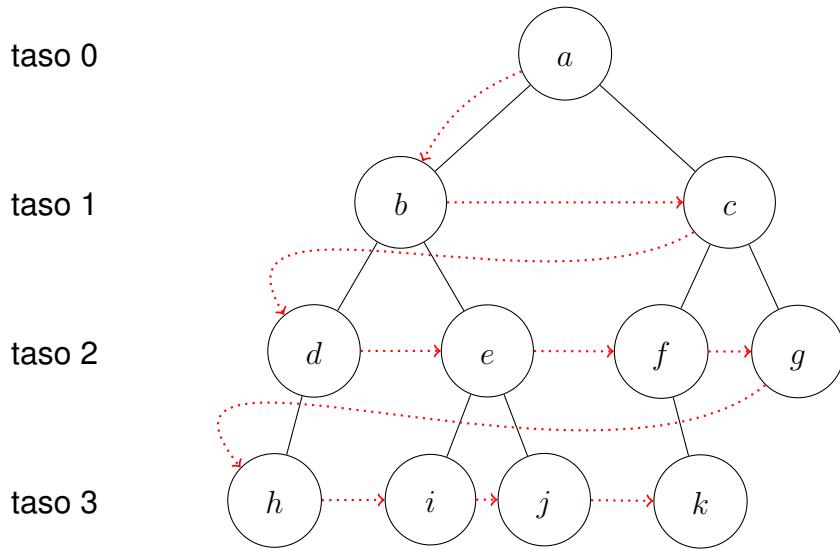
2.2 Hierarkiset tietorakenteet

Pistepilven käsitteily on paljon tutkittu aihe ja siihen käytetyistä tietorakenteista löytyy paljon julkaisuja. Esitellään seuraavaksi muutama kiinnostava hierarkinen tietorakenne. Aihealueen pioneerityönä pidetään Rusinkiewiczin ja Levoyn Qsplatia, joka onkin antanut vaikuttavia uudemmillen tietorakenteille. Dachsbacher et al. esittelivät peräkkäispisteitä, jotka voidaan visualisoida hyvin nopeasti grafiikkaprosessorilla. Viime vuosina pistepilven visualisoinnin tutkimuksen kirkkainta kärkeä on edustanut Wienin teknillisen yliopiston tietokonegrafiikan tutkimusyksikkö. Tämän tutkielman päälähteinä käytetään Claus Scheiblauerin ja Markus Schützin julkaisuja sisäkkäispistepuista.

Qsplat

Yksi ensimmäisistä pistedatan visualisointiin käytetyistä hierarkisista tietorakenteista on Rusinkiewiczin ja Levoyn esittämä QSplat, joka on kehitetty polygoniverkon visualisointiin pisteen avulla. Tietorakenne muodostetaan kolmiodusta mallista, jossa kolmioiden normaalit tunnetaan, joten se ei suoraan sovella raa'an pistepilvidatan käsitellyyn.⁵ QSplatissa on käytetty kuitenkin monia kiinnostavia tekniikoita, joita voi hyödyntää pistepilven käsitellyssä. [19]

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



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

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 seurauksena jäljelle jää yksi kolmio. Siitä muodostetaan lehtisolmu, jonka rajaava pallo sisältää koko kolmion. Puun visualisointi 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. [19]

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. [19]

Toinen hyödyllinen QSplatissa käytetty tekniikka on koordinaattien kvantisointi (*engl.*

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

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 hakutaulua, voidaan pallon sijainti esittää vain 13:lla bitillä, kun normaali liukulukuesitys vaatii vähintään 16 tavua. [19]

QSplat onnistui visualisoimaan 1,5-2,5 miljoonaa pistettä sekunnissa, mikä on sen aikaisella laitteistolla erinomainen tulos [19]. Kuten sanottu, se ei sellaisenaan kuitenkaan sovelli laserkeilattujen pistepilvien käsittelyyn. Pistepilvien pisteiden normaaleja ei yleensä tiedetä, joten ne pitäisi esiprosessointivaiheessa selvittää esimerkiksi luvussa 1.2 esitettyä tekniikalla. QSplat tarjoaa kuitenkin monia tekniikoita, joita pistepilviä käsittelvässä tietorakenteessa voidaan hyödyntää, kuten hierarknen rakenne ja koordinaattien suhteellinen esitystapa.

Peräkkäispistepuut

Dachsbacher et al. esittelevät niin kutstutun peräkkäispistepuun (*engl. sequential point tree*), jossa pisteet on aluksi järjestetty samaan tapaan pallopuuhun kuin QSplatissa. Pisteet sijaitsevat puun lehtisolmuissa ja sisäsolmuissa säilytetään täpliä, jotka juuri ja juuri peittävät solmun lasten rajaavat pallot. Tälien väri määräytyy lapsisolmujen värien keskiarvolla. [20]

Jokaiseen puun solmuun on virhelaskelmien perusteella lisätty rajat katseluetäisyydelle, jolla solmu valitaan visualisoitavaksi. Visualisointivaiheessa hierarkia litistetään tauukoksi, joka voidaan syöttää suoraan grafiikkaprosessorille. Grafiikkaprosessori käy tauukkoa läpi ja valikoi sopivat solmut, joiden perusteella täpliä piirretään ruudulle. [20]

Peräkkäispistepuut voidaan visualisoida nopeasti grafiikkaprosessorin käytön ansiossa, mutta niissä on myös heikkouksia. Tietorakenteen vaatimuksena on, että kaikki data

mahtuu grafiikkaprosessorin muistiin. Näin on vain pienillä malleilla ja tilannetta pahentaa se, että peräkkäispistepuut eivät ole kovin säästäväisiä muistin suhteen. Hierarkian jokaisessa sisäsolmussa luodaan lisää dataa, kun lapsisolmujen unionia kuvataan täplän sijainnilta ja koolla, sekä keskimääräisellä värellä.

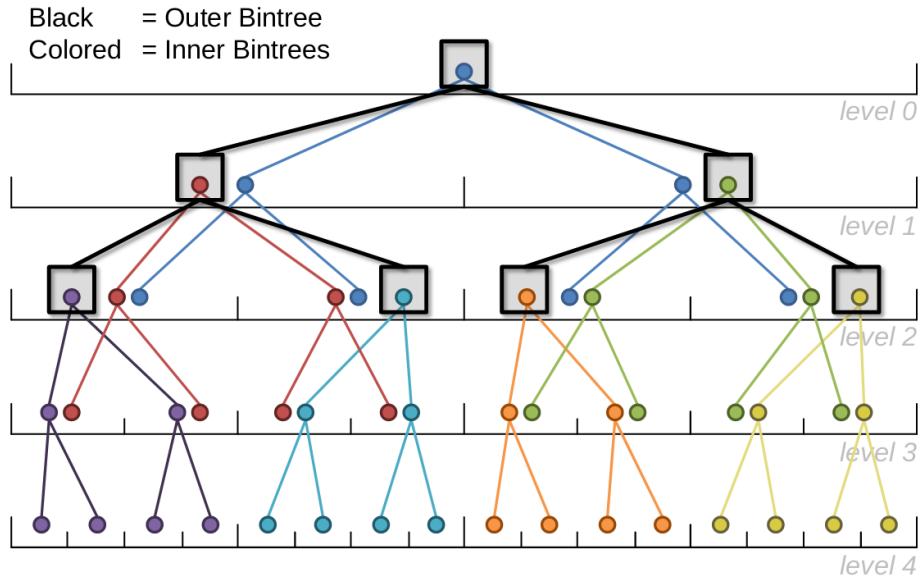
Wimmer ja Scheiblauer esittävät parannuksia peräkkäispistepuuihin. Uusien täplien luomisen sijaan puun sisäsolmuissa valitaan lapsisolmuista edustaja, joka parhaiten kuvaa sisäolmun esittämää avaruuden osaa. Tämä on tärkeä huomio, sillä käsiteltäessä massiivisia pistepilviä tulisi välittää ylimääräisen datan luomista. Hierarkiasta muodostetaan tarkkuustasot siten, että alempat tarkkuustasot sisältyvät ylempiin tasoihin ja visualisoitaessa oikea tarkkuustaso valitaan täplien koon ja katseluetäisyyden perusteella. Solmua vastaavan täplän koko määräytyy siitä, kuinka syvällä hierarkiassa se on. Wimmer ja Scheiblauer kutsuvat tästä rakennetta muistioptimoiduksi peräkkäispistepuksi. (*engl. memory optimized sequential point tree, MOSPT*) [21]

Sisäkkäispistepuut

Wimmer ja Scheiblauer kritisoivat muistioptimoituja peräkkäispistepuita siitä, että ne eivät tue näkökentän ulkopuolisten pisteiden tehokasta karsimista ja siitä, ettei muistioptimointi 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 avaruuden osan kahdeksaan osaan. Wimmerin ja Scheiblauerin tietorakenteessa oktettipuita 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. [21]

Sisäkkäisistä oktettipuista luodaan tarkkuustasot siten, että sisemmistä puista kerä-

⁷Tätä suomennosta on käytetty esimerkiksi [22]



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: [23]

tään 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. Puut tallennetaan levylle tarkkuustaso kerrallaan, mikä mahdollistaa ulkoisen muistin algoritmien käytön. Visualisoitaaessa tarvitsee levyltä lukea pisteitä vain haluttuun tarkkuustasoon asti, eikä loppuja pisteitä tarvitse ladata muistiin. [21]

Scheiblauer jalostaa sisäkkäisten oktettipuiden ideaa väitöskirjassaan esittelemällä muokkavat sisäkkäiset oktettipuut (*engl. modifiable nested octree, MNO*). Jos edellä esiteltyjä 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. [23]

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 peittäväää 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 ennaltamääritty 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. [23]

Jokainen tietorakenteen solmu tallennetaan omaan tiedostoonsa levylle, josta niitä ladataan muistiin visualisointivaiheessa tarvittaessa. Visualisointialgoritmiin kuuluu käyttäjän asettama pistebudjetti, joka asettaa ylärajan yhdessä ruudunpäivityksessä piirrettävien pisteiden määrälle.⁸ Tätä rajaa säätämällä käyttäjä saa jonkinlaisen kontrollin ruudunpäivitystaajuuden suhteeseen. [23]

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 täytetään kuten pisteitä lisättäessä. [23]

Markus Schütz jatkoi Wimmerin ja Scheiblauerin työtä esittelemällä opinnäytetyösseen verkkoselaimessa ajettavan Potree-nimisen pistepilvvisualisoijan. Potreen käyttämä tietorakenne perustuu Scheiblauerin muokkattaviin sisäkkäisiin oktettipuihin, mutta hierarkian rakennusvaiheessa kiinnitetään huomiota pisteiden tasaiseen jakautumiseen solmujen välille. Oktettipuun sisäsolmujen ruudukoihin hyväksytään uusia pisteitä vain, jos ne ovat tarpeeksi kaukana muista ruudukon pisteistä. Lehtisolmut hyväksyvät ennaltamää-

⁸Scheiblauer testasi pistepilvvisualisoijaansa asettamalla rajan vain sataantuhanteen pisteeseen.

rättyyn rajaan saakka kaikki pisteet, kunnes ne muutetaan sisäsolmuiksi ja liian lähekkäin olevat pisteet jaetaan uusien lapsisolmujen kesken. [24]

Potree käyttää ulkoista muistia tehokkaasti ja pystyy käsittämään jopa 640 miljardia pistettä sisältäviä pistepilviä.⁹ Rakennusvaiheessa oktettiipuun solmuja tallennetaan tasaisin väliajoin levylle, jottei muisti täytyisi. Kun jokainen solmu tallennetaan omaan tiedostoonsa, on yksittäisten solmujen tallentaminen ja lukeminen levyltä helppoa. Massiivisia pistepilviä kuvaavat hierarkiatkin voivat olla satojen megatavujen kokoisia. Schütz ratkaisee suurten hierarkoiden nopean lataamisen verkon yli jakamalla senkin puurakenteeseen. Nän voidaan välttää sekä turhien pisteiden, että näkökentän ulkopuolella olevien hierarkian haarojen lataaminen muistiin. [24]

Potreen visualisointialgoritmi priorisoii niitä hierarkian solmuja, jotka ovat lähellä kat-selupistettä ja joiden kuvaruudulle projisoitu koko on suuri. Visualisoinnin suorituskykyä voidaan säädellä Scheiblauerin toteutuksen mukaisesti käyttäjän asettamalla piste-budgetilla. Schütz on kehittänyt Potreehen myös hienostuneen, grafiikkaprosessorilla ajettavan algoritmin mukautuvaan pisteiden koon määrittämiseen; pistepilven harvemmissa osissa piirretään pisteet suurempina, jottei reikiä esiintyisi. [24]

3 Laitossuunnitteluhjelmistoon optimoitu tietorakenne

Tässä luvussa esitellään pistepilvien käsittelyyn ja visualisointiin soveltuva tietorakenne ja algoritmeja erityisesti laitossuunnitteluhjelmiston tarpeisiin. Ensin määritellään vaatimukset tietorakenteelle tyypillisten pistepilvien käyttötapausten mukaan, jonka jälkeen valitaan alan kirjallisuudesta sovelluskohteelle hyödyllisimmät tekniikat.

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

3.1 Käyttötapaukset ja vaatimukset tietorakenteelle

Kolme yleistä käyttötapausta pistepilvien kanssa työskenneltäessä ovat mallintaminen, mittaaminen ja katselu, jotka asettavat erilaisia vaatimuksia pistepilviä käsitlevälle ja visualisoivalle Laitossuunnitteluoohjelmistolle. Esitellään seuraavaksi käyttötapaukset ja niiden asettamat vaatimukset.

Mallintaminen

Kun laitoksesta halutaan luoda ajantasalla oleva 3d-malli pistepilven avulla, täytyy se mallintaa suunnitteluoohjelmiston käyttämäksi geometriaksi pistepilveä mukailen. Lattiat ja seinät on tasoina helppo asettaa paikalleen, kuten myös suunnitteluoohjelmiston komponenttikirjastosta löytyvä laitteet. Suurin työ on yleensä putkistoissa, ilmakanavissa ja kaapeliradoissa. Useat suunnitteluoohjelmistot tarjoavat jonkinasteista automatisointia etenkin putkien reititykseen pistepilven päälle. Ohjelmisto voi automaattisesti tunnistaa pilvestä sylinterit ja asettaa niiden päälle sopivia putkisto-osia. Vaihtoehtoisesti käyttäjä voi valita pilvestä muutamia pisteitä ja ohjelmisto laskee niiden perusteella putken pituuden ja halkaisijan ja asettaa oikean osan paikalleen. Mallinnustyö ja etenkin automaattiset mudontunnistusalgoritmit asettavat ohjelmistolle vaatimuksen tarkkuudesta. Laitossuunnitteluoohjelmistossa käytetään yleensä millimetrejä perusyksikköinä, joten pistepilvessä ei saisi esiintyä senttimetriin virheitä.

Mallintamisessa tärkeässä roolissa on suunnittelijan käyttämät näkymät ja pistepiven rajaaminen. Yleensä suunnittelija käyttää muutamaa koordinaattiakselien suuntaista näkymää samanaikaisesti, jotta kurSORIN saa helposti oikeaan paikkaan. Näkymän syvyys asetetaan usein hyvin pieneksi, jotta mallista näkyisi vain kulloisenkin mallinnustyön vaatima pieni siivu. Myös pistepilveä voidaan rajata niin, että siitä näkyy vain tarpeellinen osa. Pistepilviä visualisoivan ohjelmiston tulisi siis kyettä rajamaan pilveä toistuvasti ja nopeasti. Käyttökokemus olisi paras, jos käyttäjä pystyisi hiirellä interaktiivisesti määrittämään tilan, jonka sisäpuolella olevat pisteet visualisoitaisiin. Lisäksi pistepilvi tulee

voida visualisoida useaan eri näkymään samanaikaisesti.

Mittaaminen

Toinen tärkeä ominaisuus pistepilvien kanssa työskennellessä on mittaaminen. Pistepilviä käytetään usein tarkastamaan, mahtuuko laitokseen jokin uusi laite tai putkisto. Tällöin on hyödyllistä suorittaa mittauksia joko kahden pistepilven pisteen, tai pisteen ja 3d-mallin geometrian välillä. Mittausoperaatiossa käyttäjä valitsee pistepilvestä cursorilla haluamansa pisteen ja ohjelmisto palauttaa lähimmäksi kursoria projisoidun pisteen. Käyttäjän kannalta olisi miellyttävää, jos mittausoperaatioita tehtäessä ei tarvitsisi odotaa, kun pistepilven miljoonien pisteiden joukosta etsitään juuri kursorin alla oleva piste. Yksittäisten pisteiden hakeminen pilvestä täytyy siis olla nopeaa.

Katselu

Kolmas yleinen pistepilvien käyttökohde on 3d-mallin katselu joko laitossuunnitteluoohjelmistossa tai erityisessä mallinkatseluohjelmistossa. Etenkin suunnitteluprojektienv esimiehet haluavat usein tarkastella suunnittelijoiden luomaa 3d-mallia helposti ja nopeasti. Luonnollisesti malliin kuuluvat pistepilvet tulevat myös näkyä katselijalle. Tämä saattaa tuottaa haasteita ohjelmiston kannalta, sillä katseluohjelmistojen käyttäjillä on käytettävissä harvoin yhtä järeää laitteistoa, kuin suunnittelijoiden työasemat. Mallinkatseluohjelmistossa pistepilveä harvemmin rajataan pienemmäksi, joten visualisoitavia pisteitä on niin paljon, etteivät ne mahdu kerralla keskusmuistiin tai grafiikkaprosessorin muistiin. Yleensä käyttäjä myös liikuttaa näkymää mallin ympäri enemmän kuin mallinnustyössä, joten pistepilvivisualisoinnin suorituskyky ja tarkkuustasot ovat entistäkin tärkeämpiä.

Tässä tutkielman kehitetään laitossuunnitteluoohjelmistolle optimoitu hierarkinen tie-torakenne pistepilvien käsittelyyn. Esitetään tietorakenteelle seuraavat vaatimukset edellä mainittujen käyttötapausten perusteella:

1. On voitava visualisoida karkea yleiskuva pistepilvestä vain pienellä osalla datasta.
2. On käytettävä ulkoisen muistin algoritmeja, eli koko pilveä ei pidetä kerralla kes-

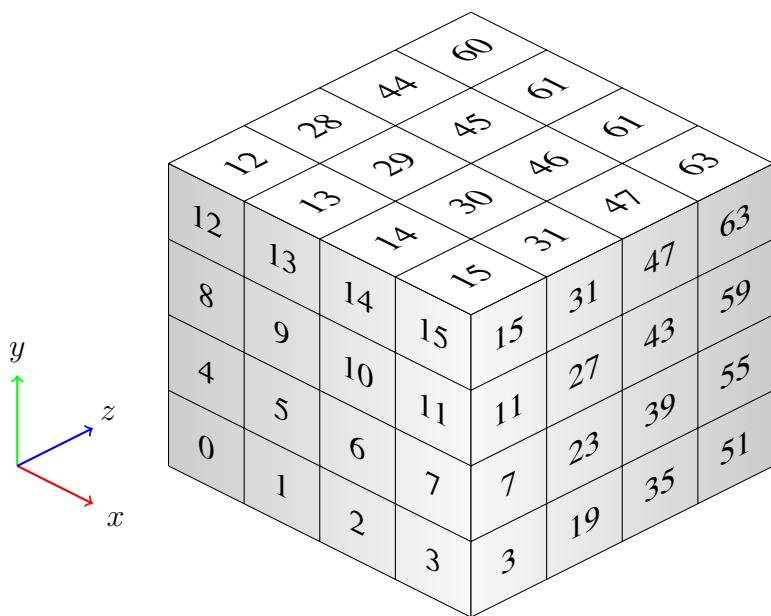
kusmuistissa.

3. Pistepilven vaatimaa tallennustilan määrää voidaan laskea harventamalla sen tiheasti näytteistettyjä osia.
4. Käyttäjän on voitava määrittää pilvestä alueita, joiden sisältävien tai ulkopuolelle jäävien pisteiden ominaisuuksia, kuten näkyvyttä tai väriä, voidaan muuttaa.
5. Pilvestä on voitava nopeasti ja tarkasti valita yksittäisiä pisteitä.
6. Pistepilvessä ei saa esiintyä yli millimetrin suuruisia virheitä.

3.2 Tietorakenteen valinta

Luvussa 2.2 esitellyt sisäkkäispistepuut näyttävät soveltuwan hyvin laitossuunnitteluohjelmiston vaatimuksiin. Oktettipuun läpikäyminen taso kerrallaan muodostaa tehokkaasti tarkkuustasot, joten vaatimus 1 on helppo tyydyttää. Pisteiden asettelu sisäkkäisten oktettipuiden solmuihin mahdollistaa myös vaatimuksen 2 mukaisesti ulkoisen muistin käytämisen. Scheiblauerin muokattavien sisäkkäisten oktettipuiden jokainen solmu sisältää ruudukon, johon pisteet sijoitetaan. Mitä syvemmällä tasolla solmu on, sitä pienempiä ruudukon solut ovat. Vaatimuksen 3 esittämä pilven harvennus onnistuu asettamalla puulle enimmäissyyvys ruudukon koon mukaan ja hylkäämällä lehtisolmuissa kaikki pisteet, jotka tulisi lisätyksi jo varattuun soluun.

Valintaoperaatiot onnistuvat nopeasti oktettipuussa. Puun jokainen solmu sisältää tiedon sen sisältämien pisteiden rajaavasta laatikosta, joten jos valinnan sijainti ei osu rajaavan laatikon sisälle, ei kyseisen solmun lapsisolmujakaan tarvitse tarkastaa. Yksittäisiä pisteitä tarvitsee tarkastella vasta kun valittavanalueen raja kulkee puun solmun rajaavan laatikon läpi, tai kun käyttäjä haluaa valita vain yhden pisteen. Tietyllä aluella sijaitsevat pisteet jakautuvat useaan oktettipuun solmuun, minkä johdosta valintaoperaatiot eivät ole triviaaleja sisäkkäisissä oktettipuissa. Vaatimuksiin 4 ja 5 voidaan kuitenkin vastata



Kuva 8. 64:n solun ruudukko

sisäkkäisillä oktettipuilla. Scheiblauer ja Schütz eivät tiivistäneet pistepilviä, joten niiden tarkkuus ei kärsinyt. Näin myöskään vaatimus 6 ei tuota ongelmia.

3.3 Ruudukko

Algoritmi 1: RakennaOktettipuu

Syöte : Joukko pistepilviä P

Tuloste: Pisteet yhdessä oktettipuussa O

```

1 // Selvitetään ensin pilvien rajaavien laatikkojen unioni
2  $L \Leftarrow$  Tyhjä rajaava laatikko;
3 for Pistepilvi  $pc \in P$  do
4   if Jokin pilven  $pc$  pisteistä on laatikon  $L$  ulkopuolella then
5     | Kasvata laatikkoa  $L$  siten, että se peittää kaikki pilven  $pc$  pisteet
6   end
7 end
8 // Lisätään pisteet oktettipuuhun
9 for Pistepilvi  $pc \in P$  do
10  Sekoita  $pc$ :n pisteet satunnaiseen järjestykseen;
11  for Piste  $p \in pc$  do
12    | LisääPuuhun( $O, p$ )
13  end
14 end

```

3.4 Pisteiden valinta

3.5 Visualisointi

4 Arviointi

4.1 Tietorakenteen rakentaminen ja lataaminen

4.2 Visualisointi

4.3 Pisteiden valitseminen

4.4 Jotain muuta?

4.5 Jotain muuta?

5 Johtopäätökset

Viitteet

- [1] T. W. Marc Levoy, Technical report, Computer Science Department, University of North Carolina at Chapel Hill (unpublished).
- [2] T. Qu ja W. Sun, Journal of Civil Engineering and Architecture **9**, 1269 (2015).
- [3] H. Rauno *et al.*, Technical report, Tiehallinto (unpublished).
- [4] C. Scheiblauer ja M. Pregebaumer, in *Proceedings of the 16th International Conference on Cultural Heritage and New Technologies* (PUBLISHER, 2011), pp. 242–247.
- [5] v. . Verkkolähde, <https://www.carnuntum.at/en/science-history/carnuntum-in-roman-times>.
- [6] F. Menna *et al.*, ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **XLI-B5**, 675 (2016).
- [7] J. Hecht, Optics and Photonics News **29**, 26 (2018).
- [8] J. Huhtanen, Helsingin sanomat .
- [9] L. G. AG, 2018.
- [10] J. Fabritius, Opinnäytettyö, Tampereen ammattikorkeakoulu, 2009.
- [11] P. W. Theiler, Ph.D. thesis, ETH Zürich, 2015.
- [12] C. M. Huang ja Y.-H. Tseng, in *29th Asian Conference on Remote Sensing 2008, ACRS 2008, 29th Asian Conference on Remote Sensing 2008, ACRS 2008* (PUBLISHER, 2008), pp. 1925–1930.
- [13] M. Berger *et al.*, Computer Graphics Forum (2016).
- [14] J. Giesen ja F. Cazals, (2006).
- [15] H. Houshiar, J. Elseberg, D. Borrmann ja A. Nuchter, Geo-spatial Information Science **18**, (2015).
- [16] M. Piipponen, Opinnäytettyö, Satakunnan ammattikorkeakoulu, 2012.
- [17] Aveva, AVEVA Laser Modeller.
- [18] CADMATIC, Keskustelut CADMATIC:n henkilöstön kanssa, 2019.
- [19] S. Rusinkiewicz ja M. Levoy, Proceedings of SIGGRAPH **2000**, (2001).
- [20] C. Dachsbaecher, C. Vogelgsang ja M. Stamminger, ACM Transactions on Graphics **22**, 657 (2003).

- [21] M. Wimmer ja C. Scheiblauer, in *Proceedings Symposium on Point-Based Graphics 2006*, Eurographics (Eurographics Association, 2006), pp. 129–136.
- [22] J. Davidsson, Technical report, 3point Oy, Lapinlahdenkatu 16 00180 Helsinki (unpublished).
- [23] C. Scheiblauer, Ph.D. thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, 2014.
- [24] M. Sch Master's thesis, .