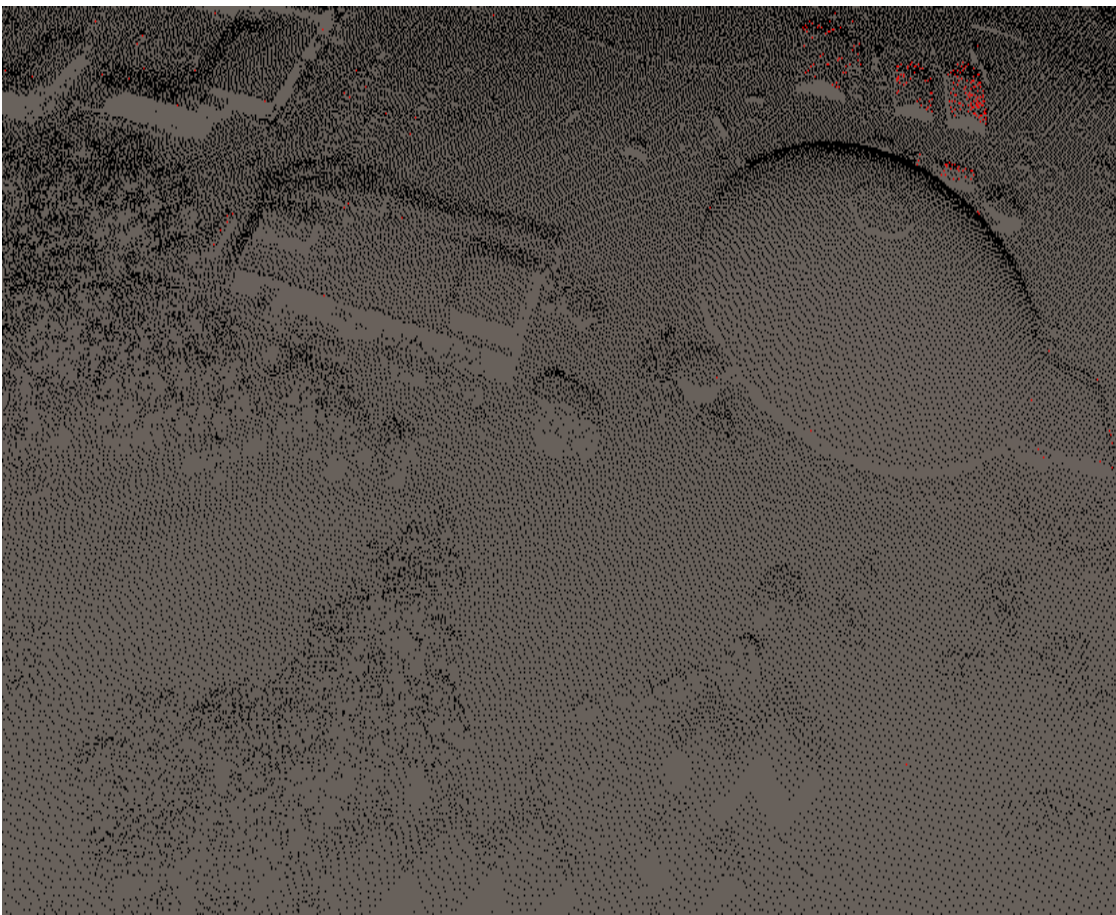


Mauno Kiiskinen

LiDAR-pistepilvivisualisointi



Tradenomi
Tietojenkäsittely
Kevät 2021



KAMK • University
of Applied Sciences

Tiivistelmä

Tekijä(t): Mauno Kiiskinen

Työn nimi: LiDAR-pistepilvivisualisointi

Tutkintonimike: Tradenomi, tietojenkäsittely

Asiasanat: Pistepilvi, visualisointi, LiDAR, Unity3D, VR, 5G, ohjelmointi

Opinnäytetyössä kommunikointiin 2D LiDAR-laitteelle ja kerättiin sen avulla dataa ympäristöstä. Työssä esitellään tekniikat saadun LiDAR-datan purkamiseen ja sen käsittelyyn, jotta saadut etäisyydet voidaan muuntaa yhtenäiseen koordinaatistoon.

LiDAR on valopulssien avulla toimiva laite, joka kerää ympäristöstä dataa mittaamalla ajan valopulssien lähettämisen ja palaamisen välillä. LiDAR-laitteita käytetään monien eri alojen kehittämiseen esim. robotiikassa ja maanmittauksessa, jossa halutaan mitata etäisyyksiä tarkoilla tuloksilla.

Pistepilvi on kokoelma pisteitä, jotka ovat kerätty yhtenäiseen koordinaatistoon. Pistepilvien avulla kuvataan esineitä ja muotoja 3D-tilassa pisteiden avulla. Pisteet muodostavat yhdessä esineiden tai muotojen pinnat, jotka voivat olla hyödyllisiä tietoja monien eri alojen käyttötarkoituksissa, esim. rakennusallalla, missä pistepilvistä voidaan muodostaa 3D-malleja tuotteista.

LiDAR-laite yhdistettiin VR-laitteistoon, jonka avulla mahdollistettiin 3D-pistepilvien muodostaminen ilman ulkopuolisia moottoreita. Dataa visualisoitiin Unity3D-pelimoottorissa muodostamalla siitä mesh-objekti, jonka verteksit vastaavat LiDAR-laitteistolta muodostettua pistepilvidataa. Visualisointiin käytettiin partikkelimateriaalia, jonka avulla voitiin antaa eri väriarvo jokaiselle mesh-objektin verteksille.

Työtä on mahdollista jatkokehittää luomalla geneerinen LiDAR-kommunikointiohjelmisto, jolloin LiDAR-laitteistoa on mahdollista vaihtaa helpommin ilman muutoksia ohjelmistokoodiin. Datastruktuurien avulla opinnäytetyön lopputulosta voidaan optimoida toimimaan tehokkaammin reaaliajassa päivitettävien mesh-objektien kanssa.

Abstract

Author(s): Mauno Kiiskinen

Title of the Publication: LiDAR Point Cloud Visualization

Degree Title: Bachelor of Business Administration

Keywords: Point cloud, visualization, LiDAR, Unity3D, VR, 5G, programming

The objective of this Bachelor's thesis was to communicate to a 2D LiDAR device and use it to collect data from the environment. This thesis presents techniques for encoding and decoding the obtained LiDAR data in order to read the data and transform it in a unified coordinate system.

LiDAR is a device that uses light pulses to collect data from the environment by measuring the time between the transmission and return of light pulses. LiDAR devices are used for the development of many different fields, such as robotics and surveying, where it is desired to measure distances with accurate results.

A point cloud is a collection of points collected in a unified coordinate system. Point clouds are used to visualize objects and shapes in 3D using points. The points form the surfaces of objects and shapes, which can be useful information in for many different applications, such as construction, where point clouds can be used to form 3D models of the products.

LiDAR was connected to VR equipment, which allowed the creation of 3D point clouds without external motors. The data can be visualized in the Unity3D engine by forming a mesh with vertices correspond to the point cloud data generated from the LiDAR device. Particle material was used for visualization, which allowed a different color value to be assigned to each vertex of the mesh object.

It is possible to further develop the thesis by creating a generic LiDAR communication software, which makes it possible to change the LiDAR device more easily without changes to the software code. With the help of data structures, the outcome of the thesis can be optimized to work more efficiently with mesh objects that can be updated in real time.

Sisällys

1	Johdanto	1
2	Laitteisto	3
2.1	LiDAR-käyttötarkoitukset.....	4
2.2	Pistepilvi.....	6
2.3	Verkottaminen	8
3	Tekninen toteutus	9
3.1	Suunnittelu.....	9
3.2	Datan prosessointi.....	10
3.3	LiDAR-yhteyden luominen	10
3.4	Visualisointi.....	14
3.4.1	Raakadata koordinaatistoon	14
3.4.2	Virheenkorjaus	15
3.4.3	Pistepilven visualisointi	16
4	Lopputulos	20
4.1	Kehitysideat ja haasteet	22
4.2	Asiakkaan palaute	25
5	Yhteenveto.....	27
	Lähteet	28
	Liitteet	

Symboliluettelo

CPU	(engl. Central Processing Unit) Prosessori
Ethernet	Lähiverkkotekniikka
float3	Muuttuja sisältää kolme liukulukuarvoa
GPU	(engl. Graphics Prosessing Unit) Grafiikan prosessointiyksikkö
Kasipuu	(engl. Octree) Puudatastrukturi
MD-komento	LiDAR-laitteelle lähetettävä mittauksen aloittamisen komento
Mesh	Dataobjekti geometrialle
MeshFilter	Unity3D-geometrian tallentamiseen käytettävä komponentti
MeshRenderer	Unity3D-geometrian piirtoon käytettävä komponentti
Mesh server	Mesh objekteille tarkoitettu palvelin
MQTT broker	(engl. Message Queuing Telemetry Transport) IoT. Laitteiden standardikommunikointiprotokolla
RTP	(engl. Real-time Transport Protocol) Tietoliikenneprotokolla tosiaikaisen datan siirtoon
SCIP2	Kommunikointiprotokolla
SerialPort	Sarjaportti kahden tietokoneen väliseen kommunikointiin
Säie	Tietokoneohjelman itsenäisesti suoritettava osa tai tehtävä
Transform	Unity3D-peliobjektin sijainnin, orientaation ja skaalan tallentamiseen käytettävä komponentti
UDP	(engl. User Datagram Protocol) Yhteydetön tietoliikenneprotokolla
Unity3D-pelimoottori	Pelien kehittämiseen tarkoitettu moottori

Vector3	3D-vektorien ja pisteiden esitysmuoto
Verteksi	3D-mallin polygonin kulmapiste
View-Frustum Culling	Piirtotekniikka, jossa objektit näköalueen ulkopuolella jätetään piirtämättä
VR	(engl. Virtual Reality) Virtuaalinen todellisuus

1 Johdanto

Nokia Networks on tietoliikenneverkkoihin käytettävien laitteiden sekä ohjelmistojen kehittämiseen suuntautuva yritys. Nokia Networks palkkasi minut tutkimaan pistepilvien reaaliaikaista visualisointia käyttäen LiDAR-laitteistoa (engl. sanoista light detection and ranging) datan lähteenä. Opinnäytetyön tavoitteena on toteuttaa yhteys LiDAR-laitteiston sekä Unity3D-pelimoottorin välille, jotka voivat olla yhdistettyinä eri päätepisteisiin ja visualisoida LiDARilta saatua dataa reaaliajassa. Opinnäytetyöhön suunniteltiin myös datan prosessointivaiheita, jotka tulevat toimimaan päätepisteiden välillä dataa välittäen ja prosessoiden. Näihin välivaiheisiin kuuluvat UDP (engl. User Datagram Protocol) yhteyden muodostaminen, RTP over UDP (engl. Real-time Transport Protocol), MQTT (engl. Message Queuing Telemetry Transport) broker sekä mesh server. Nämä välivaiheet oli tarkoitus toteuttaa osana opinnäytetyötä, mutta niistä luovuttiin. Kun data on saatu välitettyä verkon yli kaikille käyttäjille, datasta muodostetusta pistepilvestä luodaan yksinkertainen visualisointi.

Opinnäytetyössä tullaan keskittymään datan keräämiseen LiDAR-laitteistolla yhdistettynä VR-laitteistoon, jossa kerätty data tullaan muuttamaan pistepilviksi. Alkuperäinen tarkoitus oli optimoida pistepilvien visualisointi laskentavarjostimilla, mutta tästä suunnitelmasta luovuttiin opinnäytetyön edetessä. LiDAR-laitteilta kerätty data täytyy prosessoida, jotta datasta saadaan muodostettua pistepilvi. Prosessointiin sisältyy laitteiston edessä olevien esteiden etäisyyksien hankkiminen sekä etäisyyksien muuttaminen Unity3D-pelimoottorin käyttämiin koordinaatteihin. Unity3D-pelimoottorin tapa visualisoida objekteja on käyttää MeshRenderer-komponenttia sekä mesh-dataa. Näin voidaan luoda vain yksi objekti, joka piirtää näkyville määrittelemämme pistepilven.

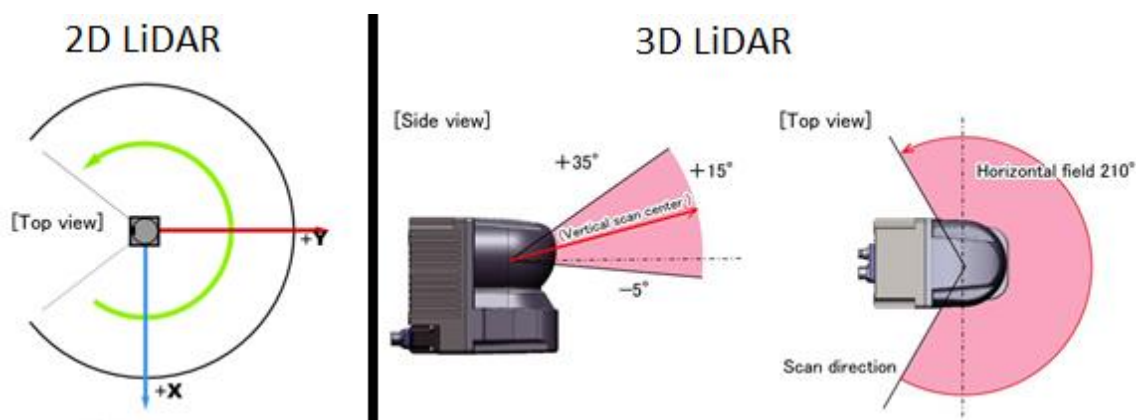
Mesh-lähestymistavassa on kuitenkin paljon ongelmia pistepilviä visualisoidessa. Mesh-objekteilla on Unity3D-pelimoottorissa rajoitettu verteksimäärä, joka yksittäiselle meshille voidaan antaa. Tämä voidaan ratkaista jakamalla mesh pienemmiksi kokonaisuuksiksi, joita Unity3D-pelimoottorissa kutsutaan submesheiksi. Jos pisteitä halutaan päivittää tai lisätä, meshien tapauksessa koko prosessi on erittäin hidasta, koska kaikki laskenta ja päivittäminen tapahtuu CPU:lla. Tämän takia pistepilviä on käytännöllisempää prosessoida GPU:lla, jota opinnäytetyössä oli tarkoitus käyttää. Näihin asioihin syvennyttään visualisointia käsittelevässä luvussa, jossa esitellään myös eri tapoja visualisoida dataa. Pyrin myös tuomaan esille, mitkä rajoitteet visualisointitekniikoilla ovat sekä mitä ne pystyvät tarjoamaan muihin tekniikkoihin verrattuna.

Tämän opinnäytetyön tavoitteena on esitellä, miten 2D LiDAR-laitteiston kanssa voidaan kommunikoida datan hankkimiseksi ja miten saatu data voidaan prosessoida ja käyttää pistepilvivisualisointiin. Opinnäytetyössä luotiin demo Unity3D-pelimoottorissa, jonka avulla tutkitaan pistepilvien visualisoinnin teknisiä rajoitteita sekä optimointimahdollisuuksia. Työssä käytettiin 2D LiDAR- ja VR-laitteistoa yhdistettynä toisiinsa, jossa VR-laitteisto toimii sijainti- ja orientaatioidatan tarjoajana. Lisäksi opinnäytetyössä pyritään tuomaan esille mahdollisia jatkokehitysideoita, jolla opinnäytetyön lopputulosta voidaan parantaa yksinkertaisilla lisäyksillä.

2 Laitteisto

LiDAR on valon mittaamiseen ja etäisyyden tunnistamiseen käytetty valotutka, joka käyttää valoa pulssilaserin muodossa etäisyyden mittaamiseen. LiDAR-laitteisto toimii lähettämällä valopulsseja esineiden pinnoille, jonka jälkeen laitteisto mittaa, kuinka kauan valolla kestää osua esineisiin ja palata takaisin skannerille. Nämä valon palauttavat pulssiarvot yhdistettynä muuhun dataan generoivat tarkkoja kolmiulotteisia kuvaelmia maan muodoista sekä pintojen yksityiskohdista, joista voidaan yhdessä muodostaa pistepilviä. LiDAR-laitteisto muodostuu kolmesta pääkomponentista, jotka ovat skanneri, laseri sekä GPS-vastaanotin. [1.]

Eroavaisuutena 2D ja 3D LiDAR-laitteiden välillä on ulottuvuus, missä skannatut etäisyydet kerätään. 2D LiDAR-laitteisto käyttää yksittäistä tasoa kaapatakseen X- ja Y- ulottuvuudet. 2D-skannerit ovat käytännöllisempiä sekä hinnaltaan edullisempia, kun pyritään suorittamaan havaitsemis- tai etäisyystehtäviä. 3D LiDAR-laitteet ovat usein kalliimpia 2D-laitteisiin verrattuna. Mallista riippuen kalliimmilla laitteilla voidaan myös saada valon intensiteetti- sekä väriarvoja. Ero voidaan nähdä käytännössä kuvassa 1. [2.]



Kuva 1. 2D ja 3D LiDAR-laitteistojen ulottuvuudet

Alkuperäinen suunnitelma opinnäytetyössä oli käyttää 3D LiDAR-laitteistoa, josta saadaan hankittua staattista dataa LiDAR-laitteistoa ympäröivästä ympäristöstä. Opinnäytetyössä käytetty LiDAR-laitteisto on Hokuyo URG-04LX-UG01-malli, joka ei ole perinteinen 3D LiDAR-laitteisto, vaan yksinkertainen versio, josta saadaan 2D-etäisyysdataa. Laitteelta ei siis tulla saamaan muuta oleellista dataa kuin etäisyys esteisiin. Tämä laite nähdään kuvassa 2. Laite on pääsääntöisesti tarkoitettu tutkimustyöhön sekä autonomisten robottien kehittämiseen, jossa LiDAR-laitetta

käytetään esteiden tunnistamiseen skannerin edessä. URG-04LX-UG01 -malli tarjoaa käyttäjille vähäisellä virrankulutuksella korkearesoluutioisia mittaustuloksia.



Kuva 2. Hokuyo URG-04LX-UG01

Tarkoituksena on käyttää 2D LiDAR-laitteiston palauttamia etäisyysarvoja yhdistettynä VR-ohjaimeen. LiDAR-laitteisto on kiinnitetty VR-ohjaimeen, josta saadaan sijainti- ja orientaatiotiedot, jotka voidaan yhdistää LiDARin palauttamiin etäisyyksiin. Näin saadaan muodostettua pistepilvi 2D LiDAR-laitteella käyttämättä ylimääräisiä moottoreita kameran pyörittämiseen.

2.1 LiDAR-käyttötarkoitukset

Kun puhutaan LiDAR-laitteista sekä niiden käyttötarkoituksista, yleisesti tarkoitetaan 3D sensoreita, joilla on mahdollista saada dataa kolmessa ulottuvuudessa. LiDAR-laitteet voidaan jakaa kahteen pääluokkaan: ilmassa toimiva LiDAR sekä maanpäällinen LiDAR. Ilmassa oleva LiDAR-laitteisto voi olla asennettuna esimerkiksi helikopteriin tai lennokkiin, josta sillä voidaan kerätä dataa. Kun ilmassa oleva LiDAR-laitteisto aktivoidaan, se lähettää valoa maata kohti. LiDAR-laitteisto mittaa valopulssien lähettämisen ja palaamisen välistä aikaa, josta voidaan laskea tarkat etäisyydet. Maanpäällisiä LiDAR-laitteistoja voidaan asentaa liikkuviin autoihin tai kolmijalkoihin, josta ne saavat kerättyä tarkkoja datapisteitä. Maanpäällisiä LiDAR-laitteistoja

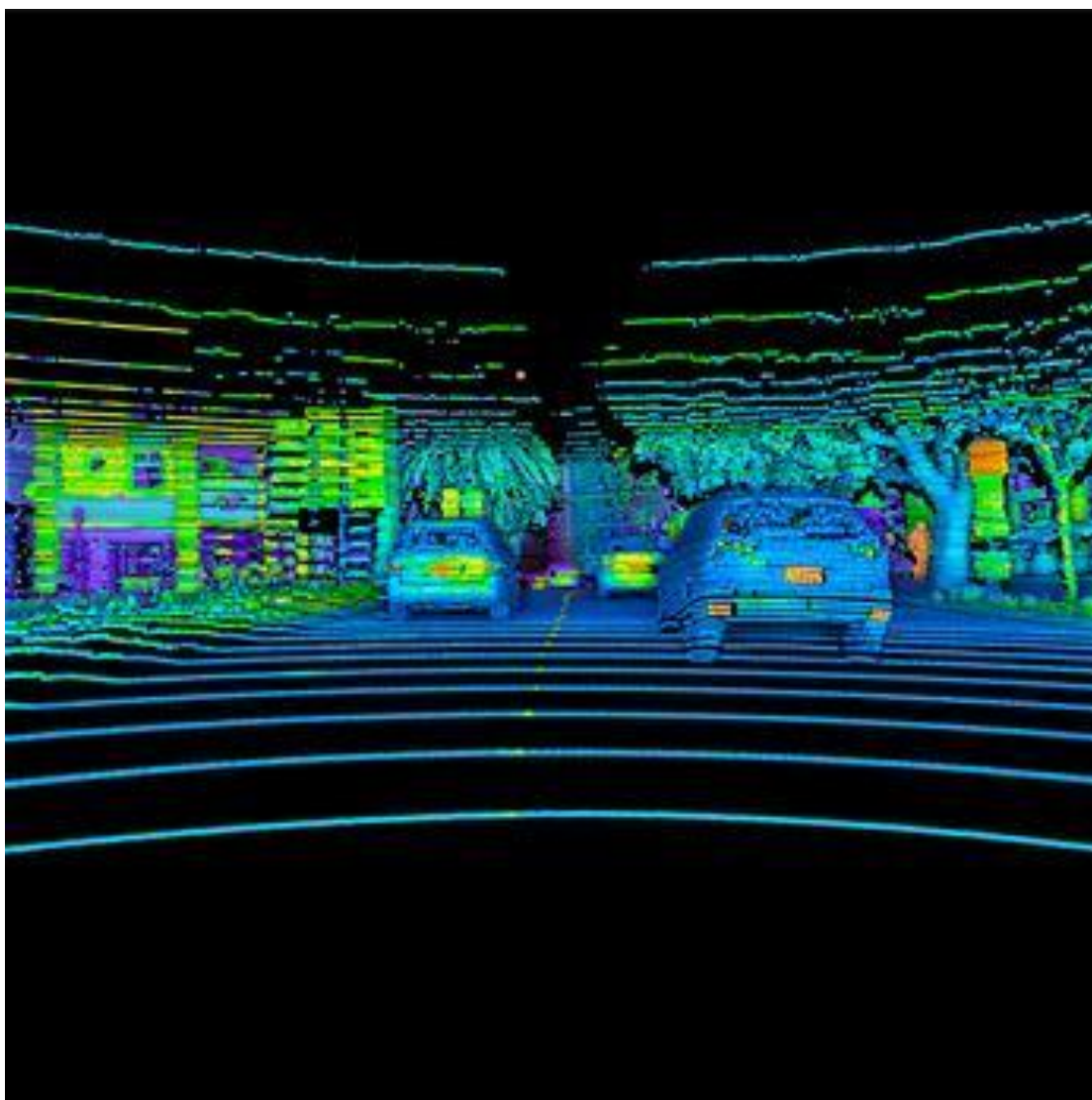
voidaan käyttää esimerkiksi moottoriteiden kuvaamiseen tai pistepilvien keräämiseen rakennuksien sisä- tai ulkopuolelta. Näiden järjestelmien avulla tutkijat ja ammattilaiset voivat tutkia luonnollisia ja ihmisen luomia ympäristöjä sekä tarkasti että joustavasti. [3.]

Digitaaliset korkeusmallit (DEM) ovat yksi ilmassa toimivien LiDAR-laitteistojen käyttötarkoituksista. DEM-malleja käytetään luomaan 3D-esitys maaston pinnoista. Ennen DEM-malleja data saatiin maan mittauksista sekä kuvamittauksilla, mutta mallien muodostaminen näiden tekniikoiden avulla on suhteellisen hidasta verrattuna LiDAR-laitteistoon. [4.] Esimerkki DEM-mallista Suomen maanmittauslaitokselta on nähtävissä Kuva 3.



Kuva 3. Jäätikköjoen kasaama reunamuodostuma ja sen rinteillä näkyviä rantamuodostumia ensimmäisellä Salpausselällä Lohjalla, esitettynä Maanmittauslaitoksen (a) maastokartalla, (b) KM25-korkeusmallilla ja (c) KM2-korkeusmallilla [4].

Maassa toimivilla LiDAR-laitteistoilla tunnetuin käyttötarkoitus on autonomisten autojen navigointi sekä törmäyksenestosenorit. Koska LiDAR-laitteisto pystyy prosessoimaan dataa nopeasti sekä tarkasti, se on mahdollistanut navigoinnin autonomisten ajoneuvojen ohjausjärjestelmiin. Törmäyksenestossa LiDAR-sensorit määrittävät esteiden tarkan sijainnin ympäröivässä ympäristössä ja tuottavat tietoja, jotka ohjaavat ajoneuvoja oikeaan suuntaan iskujen välttämiseksi. Tämä tekniikka on erityisen hyödyllinen autonomisten autojen kehityksessä. [5.] Esimerkki autonomisen auton keräämästä datasta on nähtävissä Kuva 4.

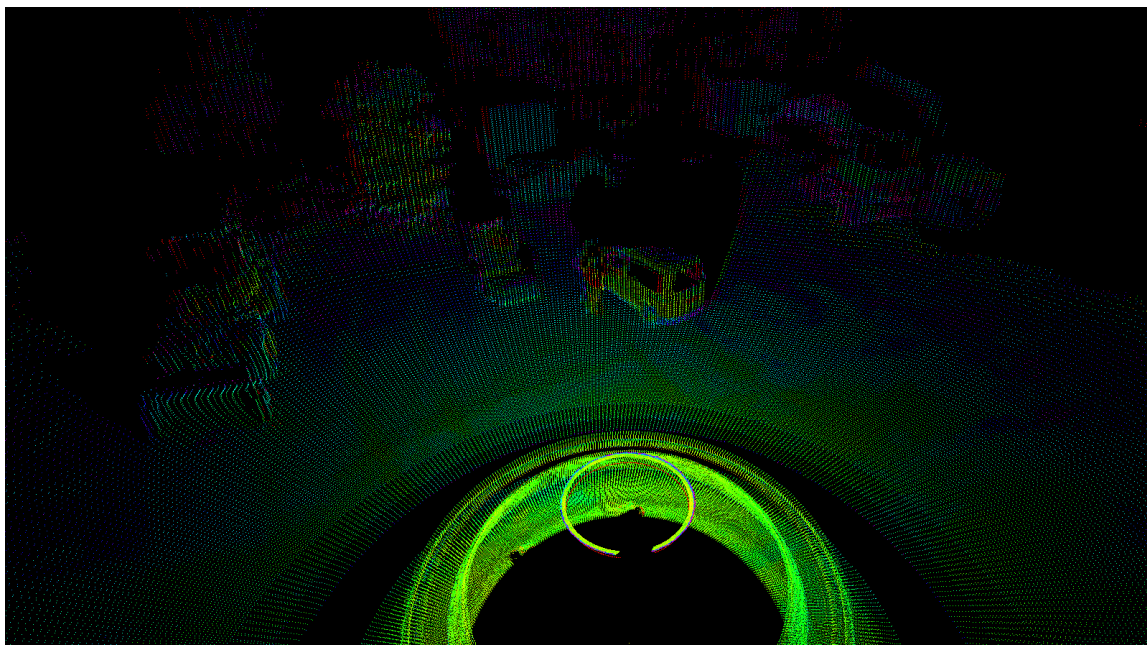


Kuva 4. LiDAR-näkymä autonomisella autolla [5].

2.2 Pistepilvi

Pistepilvet ovat datasettejä, joihin on usein kerätty tuhansia tai jopa miljoonia pisteitä. Pistepilvet voidaan määrittää 3D-koordinaatistoon, jossa pisteillä kuvataan paikkatieteellistä dataa, jossa jokainen piste vastaa tarkkaa kolmiulotteista sijaintia suhteessa muihin pistepilven pisteisiin. Pistepilviä voidaan kerätä käyttäen monia eri laitteita, joilla mahdollistetaan erikokoisten ja -tarkkuuksien pistepilvien luominen käyttötarkoituksesta riippuen. Nämä datapisteet voidaan määrittää niille määrätyn koordinaatiston mukaan. Yksittäisiin pisteisiin voidaan 3D-koordinaatin lisäksi tallentaa väridataa. Pistepilven pisteiden avulla voidaan määrittää ympäristön topologiaa, yksityiskohtia, määrittää geometriaa annetuista pisteistä sekä muodostaa 3D-pintoja datan

perusteella. Kuvassa Kuva 5 voidaan nähdä 3D-pistepilvi visualisoituna Unity3D-pelimoottorissa. [6.]



Kuva 5. 3D-pistepilvi visualisoitu

Pistepilville on monia erilaisia käyttötarkoituksia eri aloilla. Yleisin käyttötarkoitus pistepilville on luoda niistä 3D-malleja. Rakennusalalla laserskannauksella luotuja malleja rakennuksista voidaan hyödyntää etäisyyksien, alueiden sekä tilavuuksien mittaamiseen. Nämä tiedot ovat usein hyödyllisimpiä rakennus- sekä entisöintiprojekteissa. Objektien muotojen määrittäminen voi olla hyödyllistä esimerkiksi valmistettujen osien tarkastelua, metrologiaa tai laaduntarkastusta varten. Pistepilviä käytetään myös ennustavaan kunnossapitoon monilla eri aloilla. Esimerkkinä ennustavaan kunnossapitoon nähdään pistepilviä käytettävän tie-, kaivos- sekä putkitöissä. Näissä tapauksissa pistepilvestä voidaan erotella dataa, jonka avulla analysoidaan, tarvitaanko toimenpiteitä käytettävyyden tai turvallisuuden suhteen. [6.]

Pistepilviä käytetään myös metsän hoitoon sekä ylläpitoon skannaten metsää ilmasta tai maasta käsin, jonka avulla voidaan pistepilvestä mitattuna nähdä metsän korkeuksia paremmin kuin tavanomaisella kenttähavainnoinnilla. Pistepilvestä voidaan näin saada selville dataa puista, kuten niiden korkeus ja ympärysmitta, jonka avulla saadaan luotua tarkkoja karttoja metsistä. [7.]



Kuva 6. Kuvapohjainen pistepilvi [8].

2.3 Verkottaminen

Pistepilvi tullaan välittämään 5G-yhteyden välityksellä muille käyttäjille. Suunnittelin opinnäytetyöhön UDP-yhteyden sekä MQTT brokerin, jolla LiDAR-data saataisiin välitettyä kaikille käyttäjille, kun uutta dataa on saatavilla. Päädyin kuitenkin käyttämään opinnäytetyössä pohjana Nokian aikaisempaa projektia, joka käyttää Mirror Networking stackiä.

Mirror on korkean tason verkkotoiminnallisuuden kirjasto Unity3D-pelimootorille, joka tarjoaa käyttäjälle eri tapoja tehdä peliominaisuuksia moninpeleihin. Mirrorin avulla istunnon ylläpitäjä tulee olemaan LiDAR-skannaaja, joka käyttää kehittämäni työkalua tehdäkseen LiDAR-skannauksen moninpeli-istunnossa. Skannaus prosessoidaan ja siitä muodostetaan pistepilvi, joka tämän jälkeen verkotetaan Mirrorin avulla muille käyttäjille. Pistepilvi visualisoidaan loppukäyttäjillä samalla tavalla, sekä pistepilveä päivitetään aina, kun uutta dataa on saatavilla.

3 Tekninen toteutus

Tämän työn tarkoituksena on tutkia datan hankintaa LiDAR-laitteistolta, datan käsittelyä pistepilviksi sekä pistepilvien siirtämistä Unity3D-loppukäyttäjälle 5G-yhteyden välityksellä, jossa data voidaan visualisoida. Toteutus tulee käyttämään SerialPort-yhteyttä datan hankkimiseen LiDAR-laitteistolta sekä Mirror-verkkotoiminnallisuuskirjastoa datan välittämiseen loppukäyttäjille. Tavoitteena on luoda datan välittämiseen ja prosessointiin käytettävät funktionallisuudet sekä saada datan välitysprosessi optimoitua niin, että dataa saadaan välitettyä pienellä viiveellä LiDAR-laitteelta loppukäyttäjille. Tavoitteena on saada data kulkemaan lähes reaaliajassa, jonka jälkeen data voidaan visualisoida.

Visualisointiin tulee käyttää LiDAR-laitteistolta vastaanotettua dataa, jotka visualisoidaan Unity3D-pelimootorissa 3D-pisteinä. Visualisointiin tullaan käyttämään yksinkertaista toteutusta, jossa pisteitä voidaan lisätä ajon aikana. Tavoitteena on rajoittaa pisteitten määrää laitteistosta riippuen, jotta suorituskyky ei kärsisi.

3.1 Suunnittelu

Opinnäytetyön tekninen toteutus voidaan jakaa kahteen pääosioon, jotka ovat datan prosessointi sekä datan visualisointi.

Datan prosessoinnin osuus jakautuu useampaan pienempään kokonaisuuteen. Ensimmäisenä raakadata on saatava LiDAR-lähteestä, jotta sitä voidaan alkaa prosessoimaan. Kun data on saatu LiDAR-laitteistolta, se voidaan muuntaa bittimuotoon, jotta sitä on kevyempi siirtää sekä käsitellä. LiDAR-laitteisto tulee olla yhteydessä pisteeseen, josta saadaan lähetettyä dataa loppukäyttäjälle. Alkuperäinen suunnitelma opinnäytetyössä oli toteuttaa ensimmäinen vaihe UDP-tietoliikenneprotokolla datan välittämiseen loppukäyttäjälle, mutta opinnäytetyötä tehdessä tästä suunnitelmasta luovuttiin. UDP-datanvälitys haluttiin myös päivittää käyttämään RTP over UDP -tietoliikenneprotokolla, jotta dataa saadaan siirrettyä jatkuvasti. Viimeisimpänä vaiheena oli luoda MQTT broker, jonka tehtävänä on pitää listaa käyttäjistä, jotka haluavat vastaanottaa LiDAR-laitteiston dataa, ja näin välittää saatu data kaikille listan kuuntelijoille.

Datan prosessointi aloitetaan muodostamalla SerialPort-yhteys tietokoneen sekä 2D LiDAR-laitteiston välille. Kun yhteys on saatu muodostettua, voidaan LiDAR-laitteelta alkaa pyytämään

etäisyyksiä. Käytetyn 2D LiDAR-laitteiston näköalue on noin 240° laitteen edessä, josta voidaan laskea pisteiden koordinaatit laitteen näköalueen ja saatujen etäisyyksien avulla.

Datan visualisointi voidaan myös jakaa pienempiin kokonaisuuksiin, joista saatu LiDAR-data muunnetaan Unity3D-pelimoottorin käyttämään muotoon ja visualisoidaan pistepilville optimaalisella tavalla. Ensin LiDAR-data on tuotava Unity3D-pelimoottoriin. Tämän jälkeen raakadatasta prosessoidut koordinaatit voidaan muuntaa Vector3-pisteiksi, joita voidaan käyttää visualisointiin. Unity3D-pelimoottorissa voidaan luoda peliobjekteja sekä mesh-objekteja, joille voidaan antaa saadut Vector3-pisteet uusina vertekseinä. Visualisointiin voitaisiin käyttää datastruktuuria, kuten kasipuu, joka mahdollistaa paremman optimoinnin visualisoinnille, mutta rajoitettu aika ei riitä kaiken rakentamiseen. Edellä mainittu datastruktuuri on kehityskohteissa ensimmäisenä, jos lopputyötä halutaan kehittää eteenpäin sen valmistuttua. Datastruktuurilla voidaan mahdollistaa nopeampaa pisteiden etsimistä, päivittämistä, lisäämistä sekä visualisointia, koska kaikkia pisteitä ei tarvitse piirtää, vaan vain loppukäyttäjälle näkyvät pisteet.

3.2 Datan prosessointi

Alkuperäinen suunnitelma oli toteuttaa UDP-yhteys ja videonsiirtoprotokolla, jolla LiDAR-data saadaan välitettyä skannaavalta käyttäjältä muille käyttäjille. Opinnäytetyön aikana ilmenneistä haasteista johtuen jouduin jättämään datan verkon yli välittämisen Mirror network stackille, joka on Unity3D-pelimoottorin tarjoama kolmannen osapuolen kehittämä verkkotoiminnallisuustyökalu. Nokialla oli projekti, joka käyttää kyseistä työkalua, jota sain käyttää opinnäytetyössäni pohjana.

Datan prosessoinnissa on otettava huomioon, että yhteyden muodostamisen jälkeen LiDAR-laitteelta voidaan pyytää etäisyysdataa, joka vastaanotetaan koodatussa muodossa. Saatu data sisältää aikaleiman sekä jokaista tunnistusalueen kulmaa vastaavan etäisyyden, joka on laskettava ja muunnettava yhtenäiseen koordinaatistoon ennen kuin tuloksia voidaan käyttää.

3.3 LiDAR-yhteyden luominen

Useimmat LiDAR-laitteet käyttävät Ethernet-verkkokaapelia, jonka avulla yhteys saadaan muodostettua LiDAR-laitteiston ja tietokoneen välille. Opinnäytetyössä käytetty LiDAR-malli

käyttää Ethernet-kaapelin sijaan SerialPort-yhteyttä, jonka avulla se on yhteydessä tietokoneeseen.

Aloitin yhteyden muodostamisen prosessin luomalla konsoliohjelman, joka avaa ensin SerialPort-yhteyden annettuun porttiin käyttäjän antamalla portin nimellä ja siirtonopeudella. Kun SerialPort-yhteys on avattu, voidaan alkaa kommunikoida LiDAR-laitteiston kanssa. Kommunikointi tapahtuu lähettämällä SerialPortiin viestejä tekstimuodossa LiDAR-valmistajan määrittämällä formaatilla. Ensimmäiset tarvittavat komennot ovat SCIP2, joka vaihtaa LiDAR-laitteiston kommunikointiprotokollaa. Protokollan vaihdon lisäksi tarvitaan MD-komentoa, jolla kerrotaan LiDAR-laitteistolle, että mittaaminen halutaan aloittaa. MD-komentoon annetaan myös mukaan haluttu aloitus- sekä lopetusaskel, jotta laitteisto tietää, kuinka monta mittausta halutaan suorittaa. Yhteyden muodostamisesta voidaan nähdä esimerkki kuvassa 7.

```

2 references
private void OpenConnection()
{
    urg = new SerialPort(portName, baudrate);
    urg.NewLine = "\n\n";
    urg.Open();
    isConnected = true;
}

1 reference
private void StartReceiving()
{
    if (!isConnected)
        OpenConnection();

    urg.Write(SCIP_Writer.SCIIP2());
    urg.ReadLine(); // ignore echo back
    urg.Write(SCIP_Writer.MD(start_step, end_step));
    urg.ReadLine(); // ignore echo back

    receive = true;
}

1 reference
private void GetData()
{
    // Communicate with LiDAR
    OpenConnection();
    StartReceiving();

    while (receive)
    {
        List<long> distances = new List<long>();
        long timestamp = 0;

        if (urg != null)
            receivedData = urg.ReadLine();

        UrgData data = URG.DecodeData(receivedData, ref timestamp, ref distances);

        lock (lockObject)
        {
            queue.Enqueue(data);
        }
    }

    CloseConnection();
}

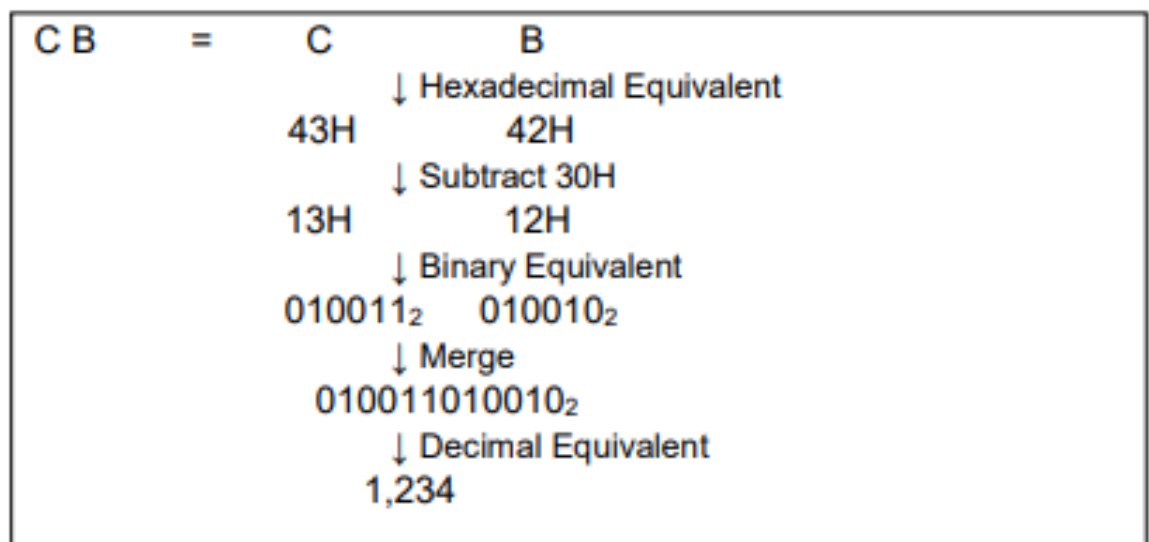
```

Kuva 7. Yhteyden muodostaminen

Kuvassa 7 voidaan nähdä edellä mainitun prosessin suorittaminen. Prosessi on suunniteltu ajettavaksi toisella säikeellä, jossa uutta dataa saadessa se lisätään jonoon, joka voidaan käsitellä päivitys while-silmukassa ja lähettää muille käyttäjille. Siirtämällä LiDAR-kommunikoinnin omalle säikeelle voidaan säästää prosessointitehoa muulle ohjelmalle. Unity3D-pelimoottori käsittelee

kaikki CPU-toiminnallisuudet pääsäikeellä, jonka täytyy tässä tapauksessa olla tietoinen ainoastaan, kun uutta LiDAR-dataa on jonossa saatavilla. Pysin pitämään LiDAR-kommunikoinnin erillään muusta toiminnallisuudesta, jotta sitä voidaan käyttää missä tahansa sovelluksessa tulevaisuudessa.

LiDAR-laitteistolta pyydetty data on muunnettu koodattuun muotoon, jota käytetään videonsiirtoon, siten voidaan vähentää välitysaikaa sensorin ja tietokoneen välillä. Saatu data on muunnettava takaisin ihmiselle luettavaan muotoon, jotta siitä voidaan muodostaa pistepilvi. Laitteiston dokumentaatiossa on mainittuna useampi koodaustekniikka, jota voidaan käyttää riippuen datan määrästä. Koska opinnäytetyössä käytetyssä 2D LiDAR-laitteistossa ei ole valon intensiteetti- tai väriarvoja mittausdatan mukana, voidaan käyttää kahdenkirjaimen koodaustekniikkaa, koska saatu data on aina alle 12 bitin kokoinen. Jotta saaduista kirjaimista voidaan lukea etäisyydet desimaalilukuina, on dataa prosessoitava. Kirjaimet muunnetaan ensin heksadesimaaleiksi, jonka jälkeen niistä vähennetään heksadesimaali "30H" arvo. Tämän jälkeen voidaan lukea heksadesimaalien bitteinä vastaavat arvot, jotka voidaan yhdistää toisiinsa muodostamaan 12-bittinen arvo. Viimein voin lukea saadun bittiarvoa vastaavan desimaaliluvun, joka vastaa LiDAR-laitteelta saatua etäisyyttä. Prosessista on nähtävissä esimerkki kuvassa 8.



Kuva 8. LiDAR-datan muuntaminen

3.4 Visualisointi

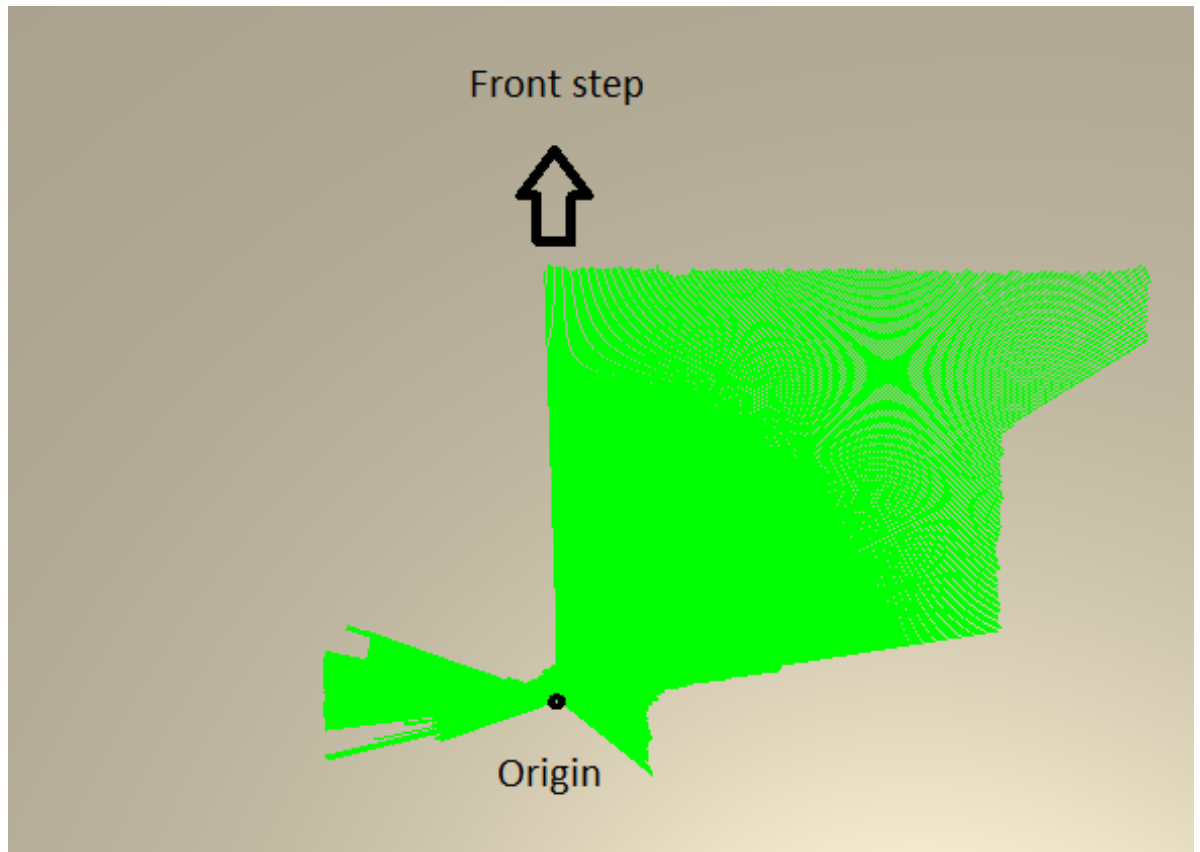
Visualisoinnin tarkoituksena on saada näytettyä jokainen datapiste loppukäyttäjälle. Myös pistepilven pisteiden määrän on tarkoituksena olla päivitettävissä, kun uutta dataa saadaan LiDAR-laitteistolta. Datan visualisointiin Unity3D-pelimoottori tarjoaa monia eri tekniikoita käytettäväksi, joista tulen esittelemään muutaman. Ennen kuin dataa voidaan alkaa visualisoimaan, se on muunnettava pelimoottorin käyttämään muotoon. Unity3D-pelimoottori käyttää float3 muuttujia peliobjektien XYZ-koordinaattien tallentamiseen. Jokaisella peliobjektilla on Transform-komponentti, joka tallentaa kyseisen peliobjektin sijainnin, orientaation sekä skaalan.

Opinnäytetyön tarkoituksena oli saada ensin tehtyä visualisointi Unity3D-pelimoottorissa niin, että pisteitä voidaan piirtää samanaikaisesti näkyville jopa miljoonia. Tähän tarkoitukseen tullaan käyttämään pelimoottorin standarditapaa piirtää objekteja ruudulle, eli käyttäen MeshRenderer- ja MeshFilter-komponentteja.

3.4.1 Raakadata koordinaatistoon

Ennen kuin voidaan muodostaa visualisointia, LiDAR-data on muunnettava Unity3D-pelimoottorin käyttämään koordinaatistoon. Kun raakadata vastaanotetaan LiDAR-laitteistolta, siitä voidaan purkaa luettavaksi jokaista kulmaa vastaava etäisyys. Käytetyssä LiDAR-laitteessa etäisyyksiä saadaan yhdellä skannauksella 768, joista ensimmäiset 43 sekä viimeiset 43 mittausta eivät ole laitteen tunnistusalueen sisäpuolella, joten ne palauttavat aina staattisen etäisyyden. Tästä johtuen nämä mittaukset voidaan jättää huomioimatta.

Unity3D-pelimoottorin yksiköt ovat metreinä, mutta vastaanotetut etäisyydet ovat millimetrejä. Yksiköt on ensin muunnettava samaan yksikköön, jotta saadaan realistinen visualisointi tuloksista. Tämän jälkeen kulma saadaan käymällä läpi jokainen laitteistolta saatu etäisyys suhteessa LiDAR-laitteen tunnistusalueeseen. Kun kulma on saatu, matematiikkakirjaston sini- ja kosini funktioiden avulla saatua kulmaa sekä etäisyyttä käyttäen voidaan viimein saada pistepilvikoordinaatit. Saadut koordinaatit vastaavat nyt LiDAR-laitteiston näkymää 2D-tasolla, joka on tuotu näkyville Unity3D-pelimoottorin editoriominaisuuksia käyttäen ja on nähtävissä kuvassa 9.



Kuva 9. 2D LiDAR-laitteiston näkyvyys

Kun data on saatu muunnettua koordinaatistoon käytettäväksi, voidaan ottaa huomioon VR-laitteisto. Kun LiDAR-laitetta siirretään VR-ohjaimen mukana, on saadut etäisyydet koordinaatistoon siirrettävä VR-ohjaimen koordinaatistoon. Tämä toiminto on suhteellisen yksinkertainen toteuttaa. Ensin LiDAR-laitteistolta saatu koordinaatti voidaan kertoa VR-ohjaimen orientaatiolla, jonka jälkeen voidaan tulokseen lisätä ohjaimen erotus keskipisteestä.

3.4.2 Virheenkorjaus

Jotta LiDAR-data voidaan yhdistää VR-ohjaimelta saatuun sijainti- sekä orientaatiodataan, LiDAR-laitteistoa on liikutettava samassa suhteessa VR-ohjaimen kanssa. Tätä ongelmaa lähdin ratkaisemaan kehittämällä 3D-mallin, joka toimii pidikkeenä LiDAR-laitteelle ja yhdistetään VR-ohjaimeen. Liittimen avulla voidaan välttää LiDAR-laitteen ja VR-ohjaimen välille muodostuvia inhimillisiä virheitä. Kuvassa 10 Kuva 10. VR-ohjainadapteri voidaan nähdä ensimmäinen 3D-tulostettu versio adapterista, jota voidaan käyttää LiDAR-laitteiston yhdistämiseen VR-

ohjaimeen. Kuvassa käytetty malli oli helpotus teknologian testaamisessa ja nopeutti opinnäytetyön prosessia huomattavasti.



Kuva 10. VR-ohjainadapteri

3.4.3 Pistepilven visualisointi

Unity3D-pelimoottori käyttää normaalisti MeshFilter- sekä MeshRenderer-komponentteja piirtääkseen peliobjekteja näkyville. MeshFilter-komponentti pitää sisällään mesh-tietoa eli dataa objektista, joka halutaan piirtää näkyville. MeshRenderer-komponentti sen sijaan käyttää mesh-dataa geometrian piirtämiseen näkyville peliobjektin Transform-komponentin määrittelemään sijaintiin, orientaatioon sekä skaalaan. Unity3D-pelimoottori voi piirtää tuhansia objekteja näkyville samanaikaisesti, mutta tämä kuormittaa CPU:ta, joka vaikuttaa suurella kuormituksella suorituskykyyn.

Visualisointiin on mahdollista hyödyntää datastruktuureita, kuten kasipuu, sekä erilaisia piirtotekniikoita, kuten View-Frustum Culling, joilla voidaan optimoida pistepilvien visualisointia vieläkin pidemmälle. Tällaisessa tilanteessa voitaisiin käyttää kasipuu-datastruktuuria, jolla voitaisiin rajoittaa samanaikaisesti piirrettävien pisteiden määrää, kun taas Viewing-Frustum Culling prosessilla voidaan jättää piirtämättä objektit, jotka jäävät kameran näkymän ulkopuolelle. Nämä tekniikat eivät sisälly tähän opinnäytetyöhön, mutta ovat mahdollisia lisäyksiä, joilla opinnäytetyön tulosta voidaan optimoida pidemmälle.

Tässä opinnäytetyössä käytetään Unity3D-pelimoottorin MeshRenderer- ja MeshFilter-komponentteja, joide kanssa käytetään pistepilvestä generoitua mesh-objektia. Näin tehtäväksi

jää ottaa saadut Vector3-sijaintitiedot pistepilvestä ja muodostaa näistä tiedoista mesh-objekti. Unity3D-pelimoottorissa tämä prosessi voidaan suorittaa CPU:lla hyvin yksinkertaisesti kuvan 11 osoittamalla tavalla.

```
1 reference
private Mesh CreateMesh(VertexData[] data)
{
    Mesh mesh = new Mesh();
    Vector3[] vertices = new Vector3[data.Length];
    int[] indecies = new int[data.Length];
    Color[] colors = new Color[data.Length];

    for (int i = 0; i < data.Length; i++)
    {
        vertices[i] = data[i].position;
        colors[i] = data[i].color;
        indecies[i] = i;
    }

    mesh.indexFormat = UnityEngine.Rendering.IndexFormat.UInt32;
    mesh.vertices = vertices;
    mesh.colors = colors;
    mesh.SetIndices(indecies, MeshTopology.Points, 0);
    mesh.uv = new Vector2[data.Length];
    mesh.normals = new Vector3[data.Length];

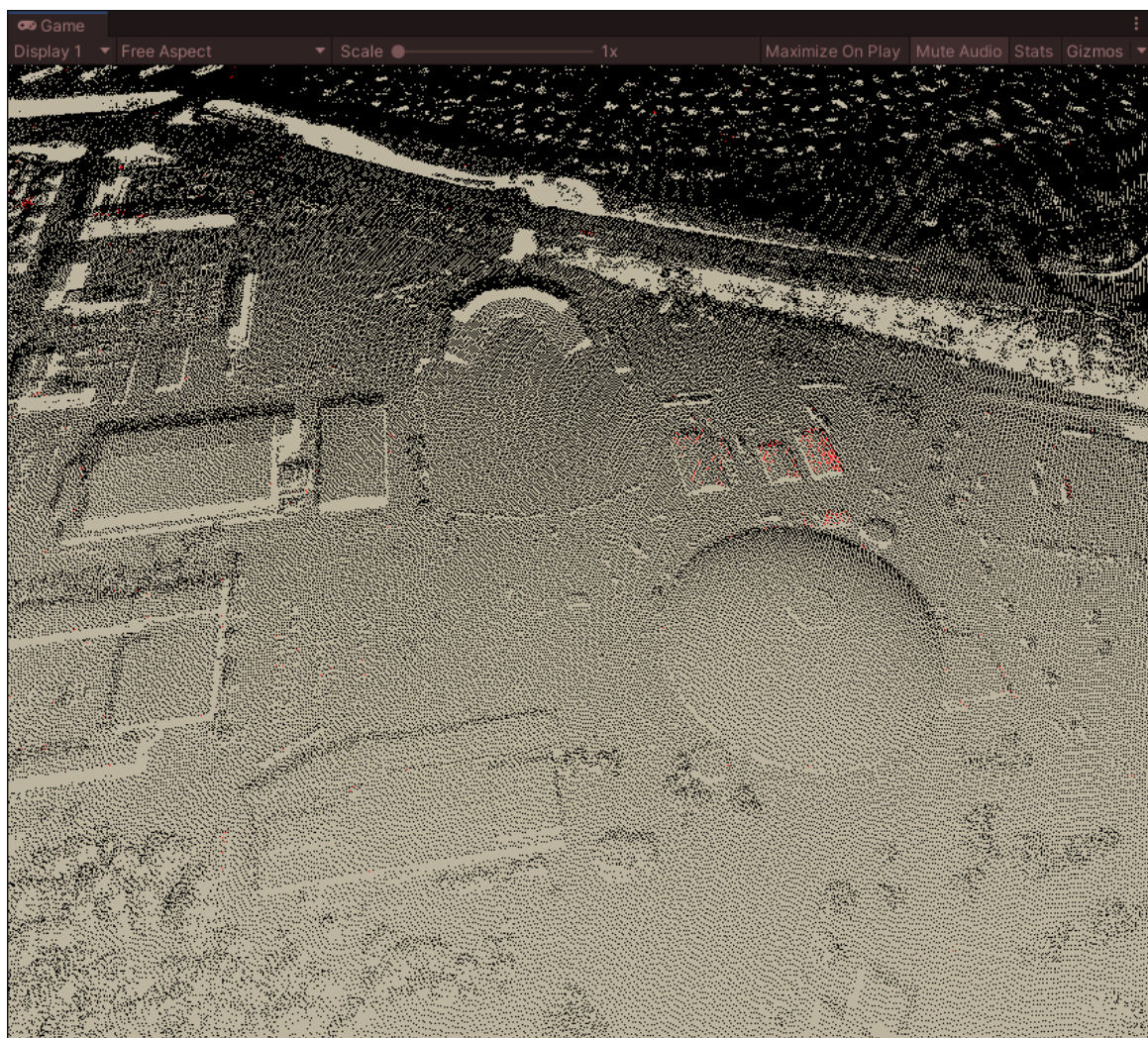
    return mesh;
}
```

Kuva 11. Mesh-objektin muodostaminen

Mesh-objektille voidaan tämän jälkeen syöttää verteksi-, väri- ja indeksisarake. Verteksien lisäksi mesh-objektille voidaan antaa verteksien värit, indeksit, kolmiot ja tekstuurikoordinaatit, joita voidaan käyttää eri tarkoituksiin. Pistepilveä visualisoidessa ei kuitenkaan haluta käyttää kolmioita pistepilven geometrian piirtämiseen, vaikka se on perinteinen tapa piirtää peliobjekteja näkyville. Tämän sijaan Unity3D-pelimoottorissa annettaessa mesh-objektille indeksit voidaan määrittää mesh-objektille haluttu topologia. Topologian avulla voidaan määrittää, miten halutaan piirtää näkyville mesh-geometria, jonka vaihtoehtoihin kuuluvat kolmiot, ruudut, viivat, viivanosat sekä pisteet, joista kolmiot ovat perinteinen tapa. Pistepilveä visualisoidessa halutaan

käyttää pistevisualisointia. Tämän avulla MeshRenderer-komponentti tietää, ettei kolmioita annettujen verteksin välille tulla piirtämään, mikä säästää prosessointitehoa.

Muita huomioitavia asioita kuvassa 11 on mesh-objektin indeksiformaatin muuttaminen. Indeksiformaatti on perinteisesti 16-bittinen, joka tarkoittaa, että mesh-objektille voidaan antaa maksimissaan 65535 verteksiä. Indeksiformaatti voidaan muuttaa 32-bittiseksi, joka mahdollistaa mesh-objektille jopa neljä miljoonaa verteksiä. GPU-tuki 32-bittiselle indeksiformaatille ei ole tuettu jokaisella alustalla, kuten useilla mobiililaitteilla. Kun mesh-objekteja on vain yksi, CPU:n ei tarvitse prosessoida useiden piirrettävien objektien datan lähettämistä GPU:lle, jotta ne saadaan piirrettyä käyttäjän näytölle. Näin on vain yksittäinen objekti, jossa mesh-data sisältää koko pistepilven datan pisteinä. Tämä data voidaan sitten antaa varjostimelle, jossa jokaiseen verteksiin voidaan piirtää pikselipiste ja piirretään ne näytölle kuvan 12 osoittamalla tavalla.



Kuva 12. Yksittäinen mesh-objekti

Kuvassa 12 käytetty data on Suomen maanmittauslaitokselta, ja se on saatavilla kaikille. Data sisältää yli viisi miljoonaa pistettä, mutta datassa ei valitettavasti ole saatavilla väridataa, joka voitaisiin visualisoida verteksivarjostimen avulla. Kun pisteitä alkaa kertymään paljon yksittäiseen pistepilveen, sen päivittäminen on raskasta suorittaa CPU:lla. On siis järkevää asettaa suurin verteksiraja yksittäiselle pistepilvelle, jotta suorituskyky ei kärsi pistepilviä päivittäessä.

Kuvan 12 toteutus ei käytä lähteenään reaaliajassa saatua LiDAR-dataa, vaan data luetaan suoraan tiedostosta. Tiedosto on LAS-formaatissa, joka tarkoittaa että se on muunnettava ensin ASCII-muotoon, jonka jälkeen data voidaan lukea Unity3D-pelimoottorin käyttämiin koordinaatteihin. Kun data on muunnettu ASCII-muotoon, voidaan kuvan 13 osoittamalla tavalla lukea datasta koordinaatti ja väriarvo, jos väri on saatavilla.

```
VertexData[] result = new VertexData[limit];
using (StreamReader reader = new StreamReader(Application.dataPath + filePath + fileExtension))
{
    string[] buffer;
    for (int i = 0; i < limit; i++)
    {
        buffer = reader.ReadLine().Split();

        // Read position buffer
        result[i].position = new Vector3(
            float.Parse(buffer[0], NumberStyles.Any, CultureInfo.InvariantCulture),
            float.Parse(buffer[1], NumberStyles.Any, CultureInfo.InvariantCulture),
            float.Parse(buffer[2], NumberStyles.Any, CultureInfo.InvariantCulture));

        // Read color buffer
        if (buffer.Length >= 5)
            result[i].color = new Color(
                int.Parse(buffer[3]) / 255,
                int.Parse(buffer[4]) / 255,
                int.Parse(buffer[5]) / 255);
        else
            result[i].color = Color.cyan;
    }
}

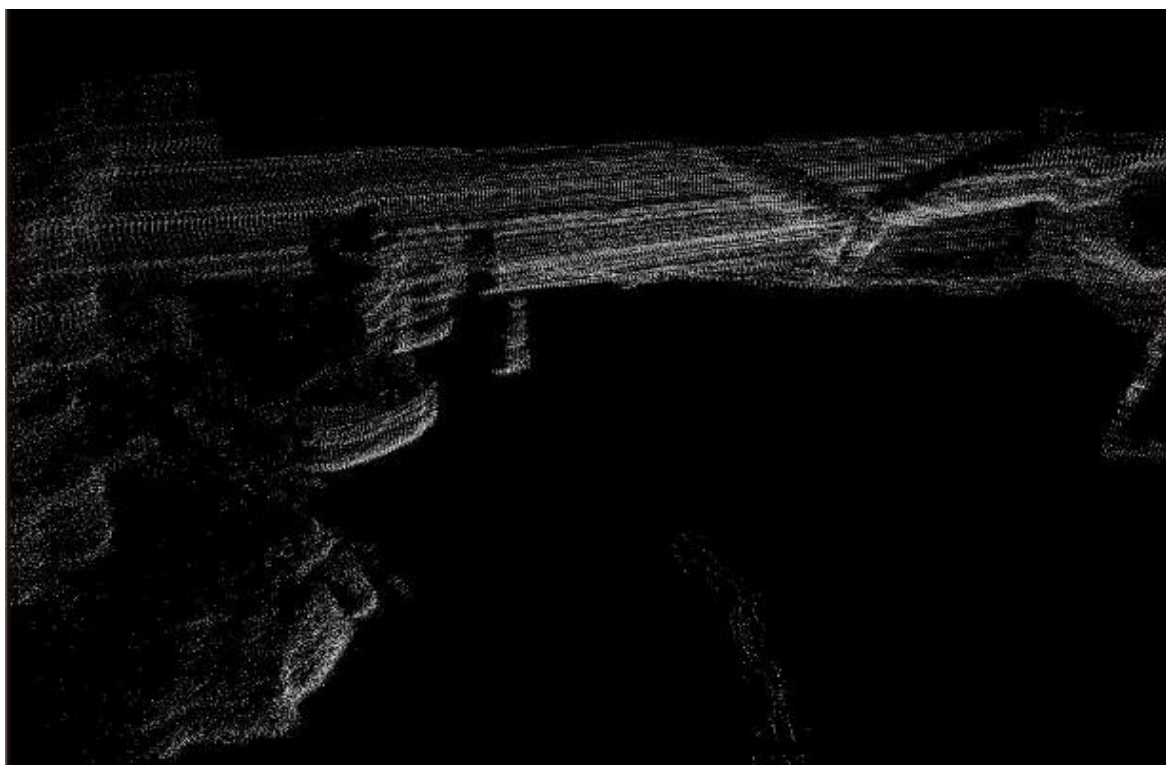
return result;
```

Kuva 13. Datan lukeminen tiedostosta

4 Lopputulos

Opinnäytetyön aihe oli mielestäni erittäin mielenkiintoinen ja motivoi minua kehittämään ohjelmistoa moniin eri käyttötarkoituksiin. Opinnäytetyössä haluttiin tuoda esille, kuinka voidaan muodostaa 3D-pistepilvidataa käyttäen halvempaa LiDAR-laitetta yhdistettynä sijainti- ja orientaatiotalähteeseen, joka tässä tapauksessa oli VR-laitteisto. Opinnäytetyön lopputulosta voidaan käyttää moniin eri tarkoituksiin pistepilvigenereoinnin avulla, sekä ohjelmistoa on lopputyön jälkeen helppo kehittää eteenpäin. Laitteiden vaihtaminen on myös mahdollista, ja siten voidaan mahdollistaa erilaisten datojen kerääminen kehitetyllä ohjelmistolla, kuten LiDAR-laitteiston päivittämisellä voidaan mahdollistaa värien, suurempien pistemäärien sekä muiden tietojen kerääminen ympäristöstä. Opinnäytetyön lopputulokseen on mahdollista jatkokehittää paljon yksinkertaisia ominaisuuksia, joilla mahdollistetaan optimaalisempi käyttökokemus ja laajempi käyttötarkoitus. Opinnäytetyön tulosta voidaan käyttää ympäristön skannaamiseen sekä skannatun pistepilven siirtämiseen useiden käyttäjien välillä. Manuaalisesti skannatut pistepilvet ovat kuitenkin kohtuullisen pienikokoisia ja niiden datan määrä on myös huomattavasti vähäisempää, kuin sen alkuperin suunniteltiin olevan. Vähäisempi pistemäärä mahdollistaa suurempia skannauksia ilman suoritusongelmia, mikä tuo opinnäytetyön tulokselle paremman käyttökokemuksen.

2D LiDAR-laitella saadaan oleellinen tieto LiDAR-laitteiston edessä oleviin esteisiin ja VR-ohjaimesta saadaan tarvittavat puuttuvat tiedot, jotka ovat sijainti sekä orientaatio. Yhdessä laitteilla saadaan muodostettua skanneri, jolla voidaan muodostaa pistepilvi päivittämällä saadut etäisyydet pisteinä lopulliseen pistepilveen. Pistepilven datan määrän kasvaessa on hyödyllistä hajauttaa pistepilvikokonaisuus pienempiin osiin, jottei CPU:lla tarvitse generoida uutta mesh-objektia, jossa jokainen piste pitää päivittää uudelleen. Uusien pisteiden lisääminen ei näin tuo suurta määrää uutta prosessoitavaa tietoa GPU:lle, mikä mahdollistaa miljoonien pisteiden visualisoinnin vähäisellä vaivalla ja prosessoinnilla. Kuvassa 14 voidaan nähdä LiDAR- ja VR-laitteistosta hankitun datan tulos yhdistettynä toisiinsa.



Kuva 14. LiDAR- ja VR-laitteistokannaus

Kuvan 14 lopputuloksessa on parannettu näkyvyyttä värjäämällä pistepilven pisteet etäisyydestä riippuen. Pisteitä on skannattu yli miljoona, mikä ei tuottanut visualisoidessa suoritusongelmia Oculus Quest -laitteella. Oculus VR-laitteistoa käytettiin LINK-kaapelin välityksellä, jonka avulla voitiin välttää android buildin tekemistä ja lataamista laitteistolle. Skannattu ympäristö on selkeästi nähtävissä ja vastaa tarkasti oikeita ympäristön pintoja. Käyttämällä LiDAR-laitteistoa, joka palauttaa valon intensiteetin tai väriarvon visualisointiin, tulosta saataisiin parannettua sekä pisteiden etäisyys tulisi entistäkin paremmin esille.

Mittauksen tuloksien visualisointiin on käytetty materiaalia MeshRenderer-komponentissa, joka käyttää Unity3D-pelimoottorin partikkelimateriaalia pohjana. Partikkelimateriaaleilla on ominaisuus värjätä yksittäisiä verteksejä, joilla voidaan yksittäisille pisteille antaa omat väriarvot ja tuoda ne esille pistepilvien muodossa. Vaikkei käytetyllä LiDAR-laitteistolla saadakaan väriarvoja datan mukana, opinnäytetyön visualisoinnin lopputuloksella värien antaminen jokaiselle pisteelle on mahdollista.

Skannaukseen ei ole pakollista käyttää VR-laitteistoa, vaan visualisaation komponentti tarvitsee tässä tilanteessa viittauksen Transform-komponenttiin, joka vastaa LiDAR-laitteiston sijaintia oikeassa maailmassa. Tällä tavalla LiDAR-laitteen sijainti- ja orientaatiolähteeksi voidaan helposti vaihtaa esimerkiksi lennokka, josta saadaan vastaavat tiedot haettua. Lennokin avulla voidaan

visualisointia käyttää DEM-mallien luomiseen 2D LiDAR-laitteistoa lähteenä käyttäen, joka voi joissain tilanteissa olla käytännöllisempi vaihtoehto.

4.1 Kehitysideat ja haasteet

Kokonaisuudessaan opinnäytetyö sisälsi monia erimuotoisia haasteita minulle kehittäjänä, mutta myös teknologiana, jota en ollut aikaisemmin käyttänyt. Tutustuminen laitteistoon ja jo olemassa oleviin teknologioihin, jotka käyttävät LiDAR-laitteita hyödykseen monin eri tavoin, vei oman aikansa opinnäytetyötä suunnitellessa. LiDAR-laitteistot eivät ole halvimasta päästä, jonka takia harrastelijoilla ei välttämättä ole mahdollisuutta saada vastaavaa laitetta käyttöönsä. Tästä johtuen ohjelmointiin tarkoitettuja lähteitä oli hyvin rajatusti saatavilla käyttötarkoitusta varten, mikä hidasti kehitystä alkuvaiheessa. Tämän lisäksi käytin opinnäytetyön alkuvaiheessa aikaa tutkiessani pistepilvidatan siirtämistä sekä formatointia LAS- ja LAZ-tiedostoiksi sekä tiedostojen purkamista takaisin luettavaan pistepilvimuotoon, jota ei ikinä päädytty käyttämään.

Jotta LiDAR-laitteiston kanssa voitiin alkaa kommunikoida, oli asennettava laitteiston valmistajan ajurit, jotta laitetta voidaan käyttää. Nämä ajurit eivät sisältäneet kehittäjän allekirjoitusta, joka Windows-käyttöjärjestelmällä tarkoittaa, että Windows on käynnistettävä turvatussa tilassa ja ohjelmien allekirjoittamisen tarkastaminen on manuaalisesti otettava pois käytöstä, jotta ajurit voidaan asentaa onnistuneesti.

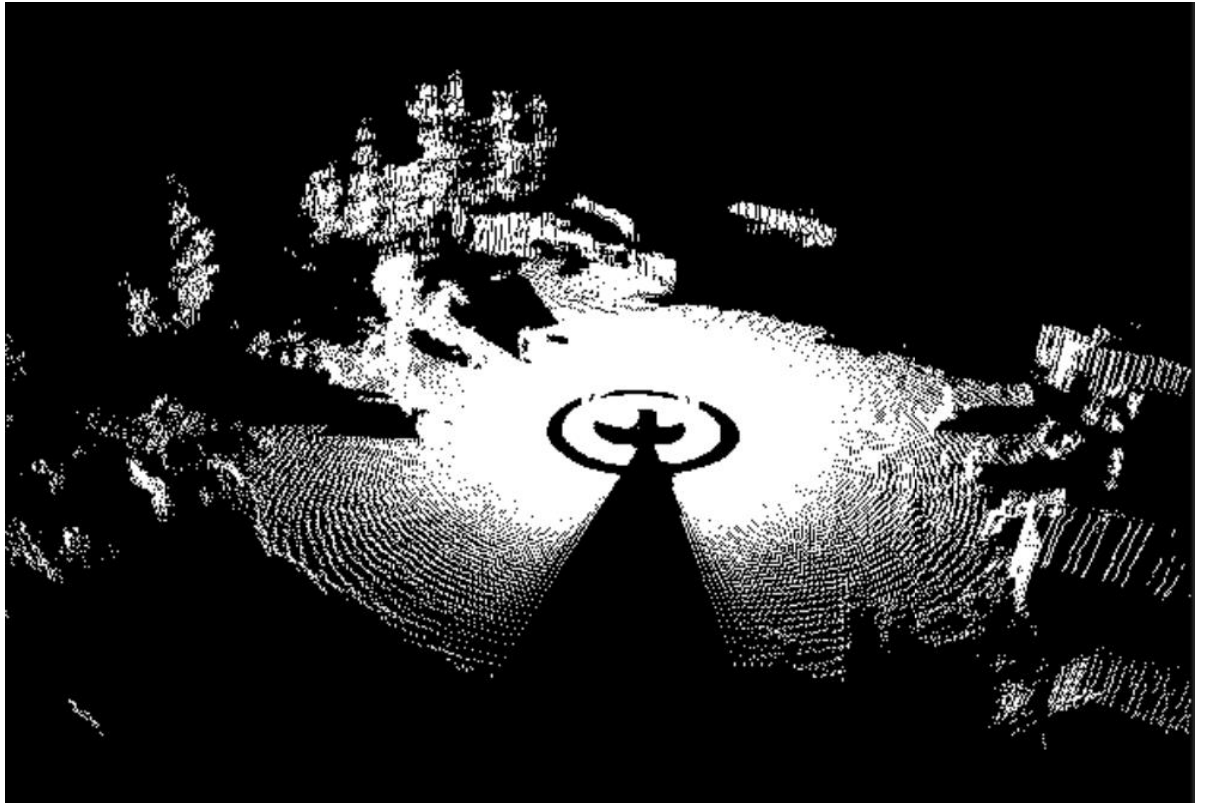
Opinnäytetyöhön olisi voitu rakentaa yleinen LiDAR-laitetuki, joka mahdollistaisi ympäristödatan hankinnan laitteen vaihtamisen helposti ilman muutoksia lähdekoodiin, mutta tämä ei valitettavasti ollut mahdollista, koska osa malleista toimii käyttäen Ethernet-yhteyttä eikä minulla ollut vastaavaa laitetta, jolla testata tätä toiminnallisuutta. Opinnäytetyön lopputulokseen voidaan myös myöhemmin lisätä käyttöön datastrukturi, kuten kasipuu, jonka avulla voidaan datan määrää rajoittaa sekä käsitellä helpommin. Datastruktuurin avulla voidaan rajoittaa pisteiden maksimimäärää, jota visualisoidaan kerralla, sekä välttää suurien pistekeskittymien muodostumista, millä voidaan selkeyttää visualisointia. Myös 5G-yhteyden hyödyntäminen opinnäytetyössä tuo paremman lopputuloksen loppukäyttäjille, kun dataa voidaan siirtää sekä käsitellä nopeammin sekä tehokkaammin.

Yksi opinnäytetyön suurin ongelmakohta on pistepilven visualisoinnin päivittäminen reaaliajassa. Kun uutta LiDAR-dataa on saatavilla päivittävällä säikeellä, se prosessoidaan ja lähetetään kaikille käyttäjille, jossa käyttäjille visualisoidaan saadut datat. Kun pisteitä saadaan lisää on pistepilven

käyttämä mesh-objekti generoitava uudelleen uusien pisteiden lisäämiseksi, jotta visualisointi näyttää jokaisen pisteen. Tämä prosessi tapahtuu CPU:lla ja hidastuu pisteiden määrän kasvaessa, mutta päivittäminen voidaan hajauttaa useammalle ruudunpäivitykselle, jolla voidaan helpottaa CPU:n prosessia. Ratkaisuna voisi toimia esimerkiksi yksi datastrukturi koko pistepilvelle, jota esimerkiksi Unity3D-pelimoottorin varjostimella voidaan käydä läpi ja päivittää annetut pistepilven arvot, johon voitaisiin antaa sijainti sekä väri. Tällä tavalla CPU:lla ei tarvitsisi tehdä muuta kuin lisätä pisteitä ylläpidettävään datastrukturiin, jonka jälkeen kehitetty varjostin tai laskentavarjostin voisi prosessoida visualisoinnin lähes reaiajassa.

Yhtenä tavoitteista oli saada mahdollisuus prosessoida kerätty pistepilvi ja kolmioittaa saatu 3D mesh-objekti käyttäen pinnan jälleenrakennusprosessia, jos idealle jäi pistepilvi visualisoinnin jälkeen aikaa. Valitettavasti pienen tavoitteen muutoksen jälkeen opinnäytetyössä ei jäänyt riittävästi ylimääräistä aikaa lisätä tällaista ominaisuutta lopputulokseen, mutta vastaava elementti voitaisiin helposti lisätä opinnäytetyön lopputulokseen jälkeenpäin. Pistepilvien kolmioitumiseen mesh-objektiksi on olemassa jo monia eri algoritmeja, joita voidaan käyttää ja joilla voidaan tuoda opinnäytetyön tuloksesta vielä enemmän hyödyllisiä tuloksia loppukäyttäjille. Visualisointi kolmiodussa mesh-objektissa tuo myös paremmin ilmi objektien muotoa ja pistepilven kokonaisuutta, kun yksittäiset pisteet eivät ole vain pikselin kokoisia, vaan muodostavat yhdessä tarkempia kokonaisuuksia.

Syvyyden erottaminen yhdenvärisistä pisteistä voi olla hankalaa suuremmissa pistepilvissä, missä pisteitä voi olla lähes päällekkäin eri etäisyyksillä. Tätä ongelmaa lähestyin käyttäen pisteiden etäisyyttä suhteutettuna mittauslaitteen sijaintiin. Visualisoidessa pisteitä ilman väridataa käytin yksittäistä väriä, joka tässä tapauksessa on valkoinen, ja värjäsin pisteet valkoisesta mustaan saadun etäisyyden perusteella. Kauimmainen piste on aina värjättyä mustaksi ja lähimmäinen valkoiseksi, mikä mahdollistaa etäisyyden erottamisen huomattavasti selkeämmin verrattuna mittaukseen ilman värin muokkaamista. Tästä voidaan nähdä esimerkki kuvassa 15, jossa on käytetty täysin valkoisia pisteitä visualisointiin.



Kuva 15. Valkoiset pisteet

Kuvassa 15 on hankalaa erottaa objekteja ja muotoja toisistaan, kun visualisointia katselmoidaan keskeltä. Kuvassa 15 on Oulun yliopistolta saatu LiDAR-laitteiston data, jonka sain käyttööni visualisointia testatessani, joka auttoi havaitsemaan kyseisen ongelman jo varhaisessa vaiheessa. Testidatassa oli mukana valon intensiteettiarvoja, joita käytin kyseisen testidatan värittämiseen kuvassa 5, mutta tiesin, ettei käytössäni olevalla LiDAR-laitteella ole samaa ominaisuutta, joka mahdollisti korjauksen valmistamisen ennen laitteen testaamista.

Tähän kehitin yksinkertaisen tavan tehdä pistepilvvisualisoinnista hallitumman ja helpommin ymmärrettävän käyttämällä pisteiden etäisyyksiä syvyyden selkeyttämiseen, joka on nähtävissä kuvassa 16.



Kuva 16. Etäisyyden mukaan värjätyt pisteet

Kuvan 16 visualisointi on huomattavasti selkeämpi, koska jokainen piste ei ole räikeän valkoinen, vaan värit luovat sulavan etäisyyssefektin, jonka avulla pistepilven muodot ja objektit ovat näin helpompi erottaa toisistaan.

4.2 Asiakkaan palaute

Sain Nokia Networksiltä runsaasti positiivista palautetta lopputyön tuloksista, joita esittelin aina kun uusiin tavoitteisiin päästiin. Kun uusiin tavoitteisiin päästiin, jatkettiin keskustelemalla, mihin on järkevintä keskittyä seuraavaksi ja mitkä aiheet ovat keskeisimpiä opinnäytetyön tuloksille. Nokialla pidettiin oma-aloitteisuudestani, mikä näkyi monissa eri opinnäytetyön vaiheissa. Kun sain tietää, ettei LiDAR-laitteisto ollutkaan soveltuva alkuperäiseen opinnäytetyön ideaan, sovelsin käyttötarkoitusta yhdistettäväksi VR-laitteistoon. Oli minun ideani kehittää 3D-tulostettava adapteri VR-ohjaimen ja LiDAR-laitteiston välille virheenkorjausta varten, mikä oli myös hyvin pidetty ideana.

Yksi alkuperäisistä tavoitteista oli päästä käyttämään laskentavarjostimia opinnäytetyössä datan prosessointiin tai mesh-objektin kolmioittamiseen siihen kehitetyn algoritmin avulla. Tätä vaihetta ei päästy kuitenkaan toteuttamaan opinnäytetyöhön, vaan visualisointi tapahtuu

toistaiseksi pisteinä ja geometrian piirtämiseen käytetään Unity3D-pelimoottorin partikkelimateriaalia. Halusin pitää laskentavarjostimet jossain muodossa mukana opinnäytetyössä, ja niille löydettiin oma paikka opinnäytetyön kehitysvaiheissa, mutta vaiheeseen ei jäänyt tarpeeksi aikaa toteuttaa loppuun asti.

Valmis lopputyö sai runsaasti palautetta onnistuneesta toteutuksesta ja työn yhtenäisyydestä. Demolla saatiin aikaiseksi paljon keskustelua, mihin eri käyttötarkoituksiin sitä voidaan käyttää ja kuinka paljon muutoksia on jatkokehityksessä tehtävä, jos halutaan vaihtaa demossa käytettyä laitteistoa. LiDAR-laitteen vaihtaminen ei tuota opinnäytetyössä mitään ongelmia, kunhan data tuodaan samassa muodossa Unity3D-pelimoottorille. Myös VR-laitteisto voidaan vaihtaa esimerkiksi lennokkiin, josta voidaan ottaa 2D dataa ilmasta käsin, jonka jälkeen datasta voidaan muodostaa DEM-malli moniin eri käyttötarkoituksiin.

5 Yhteenveto

LiDAR-laitteiden avulla on mahdollista kerätä ympäristöstä suuria määriä dataa, jota voidaan katselmoida ja käsitellä moniin eri käyttötarkoituksiin. Sovelluskehittämisessä tällainen teknologia on äärimmäisen hyödyllistä, ja käyttötarkoituksia on lähes loputtomasti. On myös mahdollista käyttää halvempia 2D LiDAR-laitteistoja soveltaen, jonka avulla on mahdollista muodostaa esimerkiksi 3D-pistepilviä ympäristöstä.

Pistepilvien käyttö on usein hyödyllinen lähestymistapa, jos visualisointiin haluttu data saadaan pisteiden muodossa tai dataa siirretään jatkuvasti. Pistepilvien käyttäminen suurien datamäärien visualisointiin on myös suosittu vaihtoehto, koska pisteiden piirtäminen on halvempaa kuin kolmioiden muodostaminen ja piirtäminen pisteiden välille. Pistepilvien avulla saadaan samalla tavalla tuotua käyttäjille esille objektien pintojen muodot sekä etäisyydet toisistaan, mutta CPU ja GPU joutuvat prosessoimaan dataa vähemmän visualisoinnin muodostamiseksi.

Mesh-objektin käyttäminen pistepilvien visualisointiin on toimiva vaihtoehto, jonka avulla voidaan piirtää näkyville miljoonia pisteitä. MeshRenderer- ja MeshFilter-komponenttien avulla pistepilvien visualisointi ja päivittäminen voidaan pitää yksinkertaisena sekä kustannustehokkaana CPU:lle ja GPU:lle. Pisteitä voidaan prosessoida tiedostoista käsin lukemalla tai reaaliaikaisen yhteyden välityksellä, joka tuo monikäyttöisen tarkoituksen opinnäytetyön visualisoinnin lopputulokselle.

Jos pistepilvivisualisointia halutaan käyttää pintojen tarkasteluun tai muihin käyttötarkoituksiin, on käyttäjänä otettava huomioon sovelluksen rajoitukset onnistuneen käyttökokemuksen saamiseksi. Pistepilvivisualisointi soveltuu objektien tutkimiseen ulkopuolelta käsin, mutta koska pintoja ei piirretä pisteiden välille, voi syvyyden erottaminen olla hankalampaa perinteiseen geometriaan verrattuna. Myös liian suuri pisteiden määrä voi heikentää käyttökokemusta, jos CPU ja GPU eivät kykene piirtämään kaikkia pisteitä kerralla ilman suoritusongelmia. Myös suuret pistekeskittymät voivat hankaloittaa objektien tutkimista, jolloin pistepilvivisualisointi ei välttämättä ole yhtä hyvä lähestymistapa normaaliin kolmioitettuun geometriaan verrattuna.

Lähteet

1. Sharma, B. (2020, 10 16). *What is LiDAR technology and how does it work?* Haettu osoitteesta <https://www.geospatialworld.net/blogs/what-is-lidar-technology-and-how-does-it-work/>
2. Hokuyo Automatic. (2009, 8 20). URG-04LX_UG01_Specification.pdf Haettu sivustolta Hokuyo Automatic URG-04LX internetosoite: <https://www.hokuyo-aut.jp/search/single.php?serial=166>
3. NOAA. (2021, 2 26). *What is lidar?*. Lidar — Light Detection and Ranging — is a remote sensing method used to examine the surface of the Earth. Haettu 17.03.2021, sivustolta National Ocean Service website internetosoite: <https://oceanservice.noaa.gov/facts/lidar.html>
4. Maanmittauslaitos. (2018). *Korkeusmallit*. Haettu osoitteesta <https://www.maanmittauslaitos.fi/tutkimus/teematietoa/korkeusmallit>
5. Domke, C. (2020, 8 3). *LiDARs for self-driving vehicles: a technological arms race*. Haettu osoitteesta <https://www.automotiveworld.com/articles/lidars-for-self-driving-vehicles-a-technological-arms-race/>
6. Doggett, S. (2020, 3 1). What are Point Clouds, And How Are They Used? Haettu osoitteesta <https://www.dronegenuity.com/point-clouds/>
7. Maanmittauslaitos. (2018). *Smart pointclouds*. Haettu osoitteesta <https://www.maanmittauslaitos.fi/en/research/smart-pointclouds>
8. Liang, X. (2015, 5 18). Forest Data Collection Using Terrestrial Image-Based Point Clouds From a Handheld Camera Compared to Terrestrial and Personal Laser Scanning. Haettu sivustolta Institute of Electrical and Electronics Engineers internetosoite: <https://ieeexplore.ieee.org/document/7109840>