

# **Volumetrinen videoiden kuvausjärjestelmän kehittäminen**

Ari-Pekka Oinonen  
Jani Seppälä

Opinnäytetyö  
Maaliskuu 2021  
Luonnontieteiden ala  
Tradenomi (AMK), tietojenkäsittelyn tutkinto-ohjelma

Tekijä(t) Oinonen, Ari-Pekka Seppälä, Jani	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Maaliskuu 2021
	Sivumäärä 47	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi <b>Volumetristen videoiden kuvausjärjestelmän kehittäminen</b>		
Tutkinto-ohjelma Tietojenkäsittelyn tutkinto-ohjelma		
Työn ohjaaja(t) Niko Kiviaho		
Toimeksiantaja(t) Digi & Game Center		
Tiivistelmä <p>Opinnäytetyön tavoitteena oli perehtyä volumetrisen videokuvauksen teknologiaan sekä tutkia mitä vaaditaan volumetristen videoiden kuvausjärjestelmän kehittämiseen. Kerättyjen tietojen perusteella kehitettiin toimeksiantajalle proof of concept –tyylinen demo volumetristen videoiden kuvausjärjestelmästä. Toimeksiantajan tärkeimmät vaatimukset järjestelmälle olivat liikuteltavuus ja helppokäyttöisyys. Opinnäytetyö toteutettiin kehittämistutkimuksena ja toimeksiantaja oli varannut kehittämistyölle 11 000 euron budjetin laitteita ja ohjelmistoja varten.</p> <p>Tutkimus aloitettiin tutustumalla volumetrisen videokuvauksen teoriaan ja aikaisempaan tutkimustyöhön. Aihe oli vielä hyvin uusi eikä siitä ollut tehty vielä paljon tutkimusta. Tämän jälkeen tutustuttiin olemassa oleviin volumetristen videoiden kuvausohjelmiin ja kameroihin sekä tilattiin tarvittava laitteisto.</p> <p>Kehitystyön aikana havaittiin, että olemassa olevilla kuvausohjelmilla ei saatu toimeksiantajan käyttöön soveltuvia tuloksia, joten päädyttiin kehittämään demo omasta kuvausohjelmistosta. Ohjelmisto kehitettiin Unity-pelimootorilla. Kehitystyössä hyödynnettiin OpenCV-kirjastoa sekä kameravalmistajan omaa ohjelmistokehityspakettia.</p> <p>Tutkimuksen tuloksena saatiin kehitettyä kuvausjärjestelmä, jolla pystyi kuvaamaan ja toistamaan volumetrisia videoita. Videoiden toistamista varten kehitettiin Unity-lisäosa, jolla järjestelmällä kuvatun videon pystyi liittämään helposti Unity-projekteihin. Lisäksi kehitettiin proof of concept –videosoitinsovellus, josta tehtiin työpöytäversio sekä internetsivuille upotettava WebGL-versio. Vaikka kehitetyllä kuvausjärjestelmällä pystyi kuvaamaan volumetrisia videoita niin suuret tiedostokoot ja laitteistovaatimukset sekä videoiden alhainen laatu rajoittivat niiden käyttömahdollisuuksia.</p>		
Avainsanat Volumetric capture, volumetrinen video, ohjelmistokehitys, kehittämistutkimus		
Muut tiedot		

Author(s) Oinonen, Ari-Pekka Seppälä, Jani	Type of publication Bachelor's thesis	Date March 2021 Language of publication: Finnish
	Number of pages 47	Permission for web publication: x
Title of publication <b>Developing a system for volumetric video capture</b>		
Degree programme Business Information Technology		
Supervisor(s) Kiviaho, Niko		
Assigned by Digi & Game Center		
Abstract  <p>The aim of the thesis was to study the theory and requirements of volumetric video capture and to build a proof of concept –demo of a volumetric video capture system for the client. The client's main requirements for the volumetric video capture system were portability and ease of use. The thesis was conducted as design-based research. The client had prepared a budget of 11 000 euros for purchasing the necessary equipment and software for the research.</p> <p>The research was started by studying earlier studies conducted on the subject and the theories and technologies behind the subject. The subject matter was still relatively new and there were not many previous studies conducted on volumetric video capture. After the initial research, the already existing volumetric video capture solutions were examined, and the required hardware was purchased.</p> <p>During the research it was discovered that the existing volumetric video capture software were not suitable for the client's needs, so a decision was made to develop a proof of concept –demo of a proprietary software for the volumetric video capture system. The software was developed using the Unity game engine, OpenCV library and the camera manufacturers' own software development kit.</p> <p>The result of the research was a volumetric video capture system which enabled recording and playback of volumetric videos. A Unity plugin was developed so that users can easily add volumetric videos to their Unity projects. A desktop- and a WebGL versions of a proof of concept –video player were also developed. While it was possible to record volumetric videos using the developed system the usage was limited by low quality, large file sizes and relatively high hardware requirements.</p>		
Keywords/tags Volumetric capture, volumetric video, software development, design-based research		
Miscellaneous		

## Sisältö

<b>1</b>	<b>Johdanto .....</b>	<b>4</b>
<b>2</b>	<b>Tutkimusasetelma .....</b>	<b>4</b>
2.1	Tutkimuksen tavoitekuvaus ja tutkimuskysymykset.....	5
2.2	Volumetristen videoiden kuvausjärjestelmän vaatimukset.....	6
2.3	Kehitystyön lähtökohdat .....	6
2.4	Opinnäytetyön rajaus .....	7
<b>3</b>	<b>Volumetrinen videotuotanto .....</b>	<b>7</b>
3.1	Volumetric capture ja volumetrinen video .....	7
3.2	Fotogrammetria ja volumetric capture .....	8
3.3	Syvyyskamerat .....	9
3.3.1	Kulkuaikatekniikka (Time of Flight) .....	10
3.3.2	Valokuviotekniikka (Structured Light) .....	11
3.3.3	Yleisiä syvyyskameroissa esiintyviä ongelmia .....	11
3.4	Volumetristen videoiden kuvausprosessi .....	13
3.4.1	Kameroiden kalibrointi .....	13
3.4.2	Kuvausympäristön kalibrointi.....	13
3.4.3	Kuvaaminen .....	14
3.4.4	Kuvatun materiaalin prosessointi.....	15
3.4.5	Vienti ja toistaminen .....	16
3.5	Volumetric capture markkinakatsaus .....	17
<b>4</b>	<b>Volumetristen videoiden kuvausjärjestelmän kehittäminen .....</b>	<b>18</b>
4.1	Laitteisto .....	18
4.1.1	Syvyyskamerat .....	19
4.1.2	Tietokoneet.....	20
4.1.3	Oheislaitteet .....	20
4.2	Olemassa olevat ohjelmistot .....	21
4.2.1	Depthkit .....	22
4.2.2	EF EVE Volumetric Capture.....	26
4.3	Oman ohjelmiston kehittäminen .....	30

	2
4.3.1 Kuvausalueen kalibrointi ja kameroiden synkronointi.....	30
4.3.2 Kameroista tulevan datan tallentaminen.....	31
4.3.3 Äänen tallentaminen .....	32
4.3.4 Kameroilla kuvatun datan siivoaminen ja yhdistäminen .....	32
4.3.5 Tallennetun tiedoston toistaminen.....	35
4.3.6 Volumetrisen videon toimivuus eri alustoilla.....	37
<b>5 Tulokset .....</b>	<b>38</b>
5.1 Kuvausjärjestelmän helppokäyttöisyys .....	38
5.2 Toistaminen .....	39
5.3 Liikuteltavuus ja hinta .....	40
<b>6 Johtopäätökset.....</b>	<b>40</b>
<b>7 Pohdinta.....</b>	<b>42</b>
7.1 Tulokset ja jatkokehitys.....	42
7.2 Tulosten luotettavuus .....	44
<b>Lähteet .....</b>	<b>46</b>

## **Kuviot**

Kuvio 1. Kuvausjärjestelmää varten hankittua laitteistoa.....	21
Kuvio 2. Tallennusetaisyyden säätäminen Depthkitissä.....	23
Kuvio 3. Näkymä Depthkitin editointi-ikkunasta.....	24
Kuvio 4. Syvyys- ja väridata yhdistettynä yhdeksi videoksi .....	25
Kuvio 5. Kuvattu kohde ennen reunojen siistimistä ja siistimisen jälkeen.....	33
Kuvio 6. Vokselikoon vaikutus pistepilven laatuun .....	34
Kuvio 7. 3DVideo-tiedoston rakenne.....	35
Kuvio 8. Näkymä videosoitinkomponentista Unity-pelimoottorissa .....	36
Kuvio 9. Näkymä videosoitinsovelluksesta .....	39

## Käsitteet

**ArUco:** OpenCV kirjastosta löytyvä merkkien tunnistukseen tarkoitettu moduuli.

**Gltf (GL Transmission Format):** Khronos Groupin kehittämä avoin tiedostoformaatti kolmiulotteisille malleille ja ympäristöille.

**IndexedDB:** Verkkoselainten käyttämä suurille datamäärille tarkoitettu tallennustila.

**Kehys (engl. frame):** Videokameran tallentama yksittäinen kuva.

**Konenäkö (engl. machine vision):** Automaattiseen kuvan tulkitsemiseen perustuva tietotekninen järjestelmä.

**Kuvataajuus (engl. framerate):** Näytölle piirrettyjen kuvien määrä sekunnissa.

**MPEG-työryhmä (Moving Picture Experts Group):** Vuonna 1988 perustettu videonpakkaustapoja suunnitteleva ja standardisoiva työryhmä.

**OpenCV (Open Source Computer Vision Library):** Avoimen lähdekoodin konenäön sekä koneoppimisen ohjelmakirjasto.

**Pelimoottori (engl. game engine):** Videopelien kehityksessä käytettävä ohjelmistokehys, esimerkiksi Unity tai Unreal Engine.

**Pistepilvi (engl. point cloud):** Kokoelma kolmiulotteisessa avaruudessa olevia pisteitä.

**Polygoni (engl. polygon):** Kolmiulotteisessa avaruudessa oleva suljettu kaksiulotteinen tasokuvio.

**Rosbag:** Tiedostoformaatti Robot Operating Systemin hyödyntämän datan tallentamiseen.

**Shader:** Ohjelma, joka määrittää miltä kolmiulotteisen mallin tulisi näyttää erilaisissa valaistuksissa.

**Unity Multiplayer HLAPI (Unity Multiplayer High Level API):** Unity-pelimoottorin verkkopelien kehittämiseen tarkoitettu järjestelmä.

**Unity Job System:** Unity-pelimoottorin moniajon mahdollistava järjestelmä.

**Videokoodekki:** Digitaalisten videoiden pakkaamiseen ja purkamiseen tarkoitettu ohjelma.

**Vokseli:** Kolmiulotteisessa avaruudessa oleva kuutiomainen pikseli.

**VR (Virtual Reality):** Tietokoneella luotu virtuaalinen ympäristö tai elämys.

**WebGL (Web Graphics Library):** Khronos Groupin kehittämä ohjelmointirajapinta interaktiivisen graafisen sisällön esittämiseen verkkoselaimissa.

# 1 Johdanto

Jo vuosien ajan videoiden katseleminen on rajoittunut yksittäisen kameran kuvakulmasta kuvattuun näkymään. Vaikka viime vuosikymmenen aikana yleistyneet 360-asteen videot toivat mahdollisuuden katsella kuvattua ympäristöä vapaasti joka puolelta, niin siitä huolimatta videoiden katselija on ollut lukittuna kameran sijaintiin. Tämä kaikki saattaa kuitenkin muuttua tulevien vuosien aikana sillä VR-laitteistojen kehittyminen ja yleistyminen on luonut kysyntää yhä immersiiivisemmille videoelämyksille.

Kolmiulotteiseen kuvantamiseen tarkoitettujen kameroiden teknologian kehittyminen mahdollistaa jo nyt oikeasti kolmiulotteisten ns. volumetristen videoiden kuvaamisen ja alan pioneerit ovatkin jo kehittäneet erilaisia ratkaisuja volumetristen videoiden tuottamiseksi, mutta aihealue on vielä hyvin uusi ja olemassa olevat ratkaisut ovat vielä melko kokeellisia.

Tämän opinnäytetyön tarkoituksena on rakentaa demo volumetriseen videotuotantoon kykenevästä kuvausjärjestelmästä Digi & Game Centerille. Työn teoriaosuudessa tarkastellaan volumetriseen videotuotantoon käytettävään laitteistoon ja ohjelmistoon liittyvää teoriaa sekä aikaisempaa aiheesta tehtyä tutkimustyötä. Itse kehitystutkimuksessa tutustutaan volumetriseen videokuvaukseen tarvittavaan laitteistoon, olemassa oleviin kuvausohjelmistoihin sekä niillä tuotetun materiaalin toistomahdollisuuksiin. Tämän lisäksi kehitetään proof of concept –prototyyppi omasta kuvausohjelmistosta sekä toisto-ohjelmasta.

## 2 Tutkimusasetelma

Tässä luvussa käydään läpi opinnäytetyön tutkimusongelma, -kysymykset, vaatimukset, lähtökohdat sekä rajaus.

## Toimeksiantaja

Opinnäytetyön toimeksiantaja on Digi & Game Center. Digi & Game Center on jyväs-kyläläinen yhteisö ja ekosysteemi, joka kokoaa yhden brändin alle keskisuomalaisia peli- ja digialan osaajia ja luo puitteet laadukkaalle ja monipuoliselle peli- ja digipalveluiden tuotannolle sekä yhteistyölle niin yritysten kuin oppilaitosten välillä Keski-Suomessa ja kansainvälisesti.

### 2.1 Tutkimuksen tavoitekuvaus ja tutkimuskysymykset

Tämän opinnäytetyön tavoitteena on luoda Digi & Game Centerille demo volumetrisen videokuvaukseen soveltuvasta tuotantotilasta. Tavoitteena on tehdä tuotantotilan laitteistosta liikuteltava, jotta sillä pystytään kuvaamaan tarvittaessa myös Digi & Game Centerin ulkopuolella. Tavoitteen pohjalta voidaan määritellä seuraavat tutkimuskysymykset, joihin pyritään vastaamaan:

- Mitä on volumetrinen videokuvaus?
- Mitä laitteita ja ohjelmistoja vaaditaan volumetrisen videokuvaukseen soveltuvan kuvausjärjestelmän rakentamiseen?
- Miten volumetristä videota kuvataan?
- Miten volumetristä videota toistetaan?

Ensimmäinen kysymys vastaa tutkimuksen teoriaosaan liittyvistä asioista. Toinen, kolmas ja neljäs kysymys pyrkivät vastaamaan itse opinnäytetyön tavoitteena olevaan tutkimusongelmaan.

Bisterin (2019) mukaan toimintatutkimus on tutkimus, jossa pyritään saamaan aikaan muutosta, puututaan asioiden kulkuun sekä kehitetään tutkittavaa kohdetta parempaan suuntaan ja sen tuloksena syntyy tuote tai sen osa. Tapauksissa, joissa toimintatutkimuksella on kehittämisenäkökulma ja sillä pyritään saamaan aikaan korjausta, parannusta tai tehostumista, voidaan puhua kehittämistutkimuksesta. (Bister 2019, 42–44.)



Koska tämän opinnäytetyön kirjallisen teoriaosuuden lisäksi kehitetään toimeksiantajalle konkreettinen demo volumetriseen videokuvaukseen soveltuvasta järjestelmästä sekä ohjelmistosta, toteutetaan opinnäytetyö kehittämistutkimuksena.

## 2.2 Volumetrinen videoiden kuvausjärjestelmän vaatimukset

Tässä käydään läpi opinnäytetyön tutkimuksen tuloksena toteutettavan volumetrinen videoiden kuvausjärjestelmän vaatimukset. Toimeksiantajan asettamat tärkeimmät vaatimukset järjestelmälle ovat:

- Liikuteltavuus
- Helppokäyttöisyys
- Budjetti: 11 000 €

Liikuteltavuus mahdollistaa sen, että asiakas pystyy kuvaamaan volumetrisia videoita tarvittaessa myös Digi & Game Centerin ulkopuolella. Tästä syystä laitteiston tulee olla helposti kasattavissa ja purettavissa, sekä suhteellisen kompakti kokonaisuus.

Toinen vaatimus tuotantotilalle on sen helppokäyttöisyys. Tämä on erityisen tärkeää siitä syystä, että Digi & Game Centerillä on useita eri yrityksiä, jotka käyttävät studioa jonne järjestelmä on tarkoitus sijoittaa, joten järjestelmästä tulisi rakentaa sellainen, että yritykset pystyvät käyttämään sitä mahdollisimman itsenäisesti.

Toimeksiantaja on varannut tätä projektia varten 11 000 euron budjetin, jolla on tarkoitus hankkia tätä järjestelmää varten tarvittavat laitteistot ja ohjelmistot.

## 2.3 Kehitystyön lähtökohdat

Koska volumetriset videot ovat vielä hyvin uusi ja tuntematon aihealue sekä toimeksiantajalle, että opinnäytetyön tekijöille on tutkimuksessa ensin perehdyttävä aiheen teoriaan. Tämän teorian pohjalta lähdetään kartoittamaan markkinoilla olevia laitteistoja, ohjelmistoja sekä mahdollisia valmiita ratkaisuja.

Toimeksiantajalla on kuvausjärjestelmää varten varattuna tuotantotila Digi & Game Centerille, mutta kaikki tarvittavat laitteistot ja ohjelmistot tullaan hankkimaan projektin aikana ja hankintoihin käytetään toimeksiantajan varaamaa budjettia.

Projekti alkaa toukokuussa 2020 laitehankintojen ja teoriaosuuden osalta ja itse järjestelmän kehittäminen aloitetaan, kun kaikki tarvittavat laitteet ja ohjelmistot on saatu tilattua. Opinnäytetyön tekemiseen on varattu noin 400 tuntia aikaa ja sen on tarkoitus valmistua vuoden 2020 loppuun mennessä.

## 2.4 Opinnäytetyön rajaus

Opinnäytetyössä selvitettiin mitä laitteita ja ohjelmistoja tarvitaan volumetriseen videokuvaukseen soveltuvan järjestelmän rakentamiseen. Lisäksi vertailtiin kahta olemassa olevaa kuvausohjelmistoa sekä kehitettiin prototyyppi omasta kuvausjärjestelmästä, jota varten kehitettiin myös proof of concept – tyylinen demo omasta kuvausohjelmistosta. Tämä työ kattaa siis koko volumetrisen videon tuotantoprosessin kehityksen kuvauksesta toistamiseen. Projektin kokeellisen luonteen ja aihealueen uutuuden vuoksi kehitystyö rajattiin toimivan prototyypin kehittämiseen.

# 3 Volumetrinen videotuotanto

Tässä kappaleessa käydään läpi volumetriseen videotuotantoon liittyvää teoriaa, sekä keskeisimmät käsitteet. Koska volumetrinen videotuotanto on vielä suhteellisen uusi aihealue, joka kehittyy jatkuvasti, niin opinnäytetyössä on käytetty pääasiassa vain tuoreita, noin muutaman vuoden takaisia lähteitä.

## 3.1 Volumetric capture ja volumetrinen video

Volumetric capture on teknologia, jossa kameroiden tuottamista kuvista ja sensoreiden tuottamasta syvyysdatasta luodaan kolmiulotteinen malli, jota voidaan sitten

käyttää esimerkiksi pelimoottoreissa sekä erilaisissa virtuaaliympäristöissä. Tällä saadaan aikaiseksi vahva psykologinen vaikutus käyttäjään. Missä tahansa käyttökohteessa, jossa normaalisti käytettäisiin tietokoneella luotua kolmiulotteista mallia, voidaan immersion lisäämiseksi käyttää volumetric capture teknologialla luotua kolmiulotteista mallia. (Porter & Porter 2018, 1–2.)

Volumetric capturella tarkoitetaan yleisesti prosessia, jossa luodaan pistepilvi yksittäisestä kuvasta tai kuvasarjasta ja volumetrisella videolla taas yleensä tarkoitetaan volumetric capturella tallennetun datan toistamista videomuodossa. Molemmilla aihealueilla on kuitenkin ollut ennen ristiriitaisia määritelmiä ja nämä aihealueet kehittyvät ja muuttuvat muutenkin niin nopeaan tahtiin, että alan ammattilaisilla on ollut vaikeuksia päästä termien määritelmistä yhteisymmärrykseen. (Porter & Porter 2018, 4.)

Siinä missä tavallinen video käyttää vain litteitä kaksiulotteisia kuvia ja 360 asteen video käyttää pallonmuotoon projisoitua panoraamakuvaa, on volumetrinen video täysin kolmiulotteinen ja koostuu kolmiulotteisista elementeistä, kuten vokseleista tai polygoneista. Tämä mahdollistaa sen, että käyttäjä pystyy tarkastelemaan kuvattua sisältöä liikuttamalla kameraa vapaasti kuvatussa ympäristössä, kun taas tavallisissa videoissa sekä 360 asteen videossa käyttäjä ei pysty vapaasti liikkumaan kuvatussa ympäristössä vaan on lukittuna katselemaan sisältöä sen kuvanneen kameran sijainnista. (Gopalakrishnan, Han, Pair & Qian 2019, 135.)

Periaatteessa mitä tahansa teknologiaa tai kameraa, joka pystyy tuottamaan syvyysinformaatioita kuvattavasta kohteesta, voidaan käyttää kohteen kolmiulotteiseen mallintamiseen. Pistepilven luominen syvyysinformaatiosta onkin volumetric käsitteen ensimmäinen laajalti hyväksytty määritelmä. (Porter & Porter 2018, 4.)

### 3.2 Fotogrammetria ja volumetric capture

Fotogrammetria on teknologia, joka on saanut alkunsa jo 1800-luvun puolivälissä. Se on kohteen kolmiulotteisessa mittauksessa käytetty tekniikka, jossa kuvattava kohde

kuvataan vähintään kahdesta eri kuvakulmasta. Tämän jälkeen kuvissa olevien kiintopisteiden sijainteja verrataan toisiinsa ja vertailusta saadun informaation perusteella pystytään laskemaan kiintopisteiden sijainti kolmiulotteisessa avaruudessa. Näiden pisteiden perusteella voidaan luoda pistepilvi, jonka perusteella voidaan rakentaa kuvastusta kohteesta kolmiulotteinen malli. Laskennallista fotogrammetriaa onkin hyödynnetty jo vuosikymmenten ajan ja tietotekniikan kehitys on luonut sille useita potentiaalisia käyttökohteita, kun taas volumetric capture on paljon uudempi innovaatio kolmiulotteisen datan keräämiseen. (Porter & Porter 2018, 2.)

Vaikka fotogrammetria ja volumetric capture ovat teknisesti hyvin erilaisia niin niiden kuvastun informaation käsittelyssä käytetyissä prosesseissa löytyy hyvin paljon yhteneväisyyksiä. Molemmista prosesseista hyödynnetään syvyysinformaation perusteella luotua pistepilveä kolmiulotteisen mallin luomiseen, mutta siinä missä fotogrammetriassa syvyysdatan luominen vaatii kuvien määrästä riippuen tuhansien tai jopa miljoonien pisteiden laskemista kuvien perusteella, niin syvyyskamerasta saadaan syvyysinformaatio suoraan kamerassa olevalta syvyysensorilta. (Porter & Porter 2018, 3–4.)

### 3.3 Syvyyskamerat

Tässä osiossa käydään läpi syvyyskameroiden teknologioita ja erotellaan niiden hyviä ja huonoja puolia sekä eri kamerateknologioiden yleisimpiä ongelmia.

Syvyyskamerat ovat kameroita, jotka käyttävät erilaisia teknologioita tallentaakseen syvyysinformaatiota. Nykyään saatavilla olevat syvyyskamerat hyödyntävät monia erilaisia teknologioita syvyysinformaation tuottamiseksi. Näitä ovat muun muassa infrapunavalot, stereokuvaus ja lasermittaus. (Porter & Porter 2018, 4.)

Kolmiulotteista syvyysdataa tuottavat kamerat, jotka käyttävät joko kulkuairotekniikkaa (Time-Of-Flight) tai valokuviota (Structured Light) ovat viime vuosina kehittyneet huomattavasti ja näin ollen niiden suosio on kasvanut. Nykyään nämä teknologiat mahdollistavat kolmiulotteisen datan tallentamisen videolaatuisella kuvataajuudella.

Tämä kehitys on luonut entistä paremmat mahdollisuudet monenlaisien kolmiulotteista konenäköä hyödyntävien ratkaisuiden kehittämiseksi. (Grzegorzec, Koch, Kolb & Theobalt 2013, 2.)

### 3.3.1 Kulkuaikatekniikka (Time of Flight)

Kulkuaikatekniikkaan perustuvat kamerat mahdollistavat hyvän ja tehokkaan tavan reaaliaikaiseen kohteiden ja ympäristöjen kolmiulotteisen geometrisen informaation tallentamiseen. Niiden toimintaperiaatteen takia ne ovat kuitenkin alttiita monenlaisille mittausvirheille. Viimeisen kymmenen vuoden aikana tehtyjen tutkimusten perusteella on havaittu, että mittausvirheitä voivat aiheuttaa muun muassa sensorin lämpötila, piirin suunnittelu sekä kuvausympäristö. (Grzegorzec ym. 2013, 6.)

Kulkuaikatekniikalla toimivat kamerat keräävät yleensä kaksi erilaista mittausta kuvattavasta kohteesta; syvyys- ja voimakkuuskuvan. Voimakkuuskuva vastaa sensorille palaavan valon määrää ja sillä määritellään saadun syvyysdatan laatua sekä luotettavuutta. Yksi tärkeimmistä ja oleellisimmista tavoista, jolla kulkuaikatekniikalla toimivan sensorin kuvanlaatua voidaan parantaa, onkin kameran sensorin kalibrointi. (Grzegorzec ym. 2013, 6.)

Yksi suurimmista negatiivisista puolista kulkuaikatekniikassa on se, että sen tarvitsema valonmäärä on suurempi verrattuna valokuviotekniikkaan, joten sensori lämpee enemmän, jolloin syvyysmittauksessa saattaa esiintyä herkemmin virheitä. Tästä syystä kulkuaikatekniikkaa käyttävä kamera vaatii valokuviotekniikkaa käyttävää kameraa tehokkaamman jäähdytysjärjestelmän. (Kolb, Lefloch & Sarbolandi 2015, 11.)

Toinen kulkuaikatekniikkaa hyödyntäviä kameroita vaivaava ongelma, joka ei vaivaa valokuviotekniikkaa käyttäviä kameroita, on kuvattavan kohteen heijastavuudessa olevien erojen aiheuttama sensorin lähettämän valon intensiteettiin liittyvä etäisyysvirhe. Virhe on herkemmin havaittavissa kohteissa, jotka ovat kameran lähellä, mutta virhe on havaittavissa myös kohteissa, jotka ovat kauempana kamerasta. Tätä virhettä voidaan kuitenkin korjata kalibroimalla. (Kolb & Lindner 2007, 2.)

### 3.3.2 Valokuviotekniikka (Structured Light)

Valokuviotekniikkaa hyödyntävät kamerat rakentavat syvyyskuvan lähettämällä infra-punavalolla tuotetun kuvion kuvattavaan kohteeseen. Kamera sitten vertaa takaisin heijastunutta kuviota kamerassa olevaan esikalibroituun kuvioon ja laskee sen perusteella syvyysinformaation omien algoritmiensa perusteella. (Hsieh, Leang, Lin, Schipf, Tsui & Wang. 2014, 1.)

Valokuviotekniikkaa hyödyntävien kameroiden heikkous kulkuakatekniikka hyödyntäviin kameroihin verrattuna on se, että ne ovat herkempiä kuvattavan kohteen ja sen ympäristön olosuhteiden aiheuttamille mittausvirheille. Koska valokuviotekniikka hyödyntävä kamera vaatii kuvion heijastamista kuvattavaan kohteeseen ei niissä yhtä helposti voida hyödyntää ulkoisia valonlähteitä mittaustulosten parantamiseksi. (Hsieh ym. 2014, 2.)

Toisin kuin kulkuakatekniikkaa hyödyntävät kamerat, valokuviotekniikkaa hyödyntävät kamerat eivät kärsi kuvattavan kohteen heijastavuudessa olevista eroista johtuvista etäisyysvirheistä, koska ne luovat oman kuvionsa kuvattavaan kohteeseen. (Kolb ym. 2015, 14.)

### 3.3.3 Yleisiä syvyyskameroissa esiintyviä ongelmia

Syvyyskamerat keskittävät kameraan heijastuvaa valoa tavallisten kameroiden tapaan, joten ne kärsivät samoista optiikkaan liittyvistä ongelmista kuin mitkä tahansa muutkin kamerat, mutta näiden lisäksi on olemassa muitakin ongelmakohtia, joita ei esiinny tavallisissa kameroissa (Kolb ym. 2015, 10). Alla on esitelty yleisimmät kulkuakatekniikkaa ja valokuviotekniikkaa hyödyntävien syvyyskameroiden ongelmat.

#### **Ympäristön valaistus**

Sekä kulkuakatekniikkaa, että valokuviotekniikkaa hyödyntävät kamerat kärsivät ympäristön liiallisesta valaistuksesta liittyvistä mittavirheistä. Etenkin valokuviotekniikkaa hyödyntävät kamerat ovat erityisen alttiita tälle ongelmalle, sillä liiallinen valaistus saattaa hankaloittaa kameran luoman kuvion havaitsemista. (Kolb ym. 2015, 10.)

### **Useiden laitteiden yhtäaikainen käyttö**

Molemmat kuvaustekniikat hyödyntävät kohteen aktiivista valaistusta syvyys datan luomiseen, saattaa kohteen kuvaaminen usealla syvyyskameralla yhtä aikaa aiheuttaa virheitä kameroiden mittaustuloksissa (Kolb ym. 2015, 10).

### **Lämmönvaihtelu**

Lämpötilan vaihtelu saattaa vaikuttaa sensorin tuottamiin mittaustuloksiin. Pitkään kuvatessa sensori lämpenee ja tämän seurauksena samasta kohteesta saadut mittaustulokset saattavat sensorin lämmettyä poiketa alkuperäisistä tuloksista. (Kolb ym. 2015, 11.)

### **Systemaattinen mittausrvirhe**

Systemaattiset mittausrvirheet voivat johtua monista eri tekijöistä. Ne voivat johtua muun muassa syvyyskameran sensorin riittämättömästä kalibroinnista tai siitä, että sensorin resoluutio on liian rajallinen kohteen etäisyyden tarkkaan määrittelyyn. (Kolb ym. 2015, 11.)

### **Syvyysdatan epäyhtenäisyys**

Kuvattavan kohteen yksittäisissä mittaustuloksissa saattaa esiintyä muun syvyysdatan kanssa epäyhtenäisiä mittaustuloksia. Valokuviotekniikkaa hyödyntävissä kameroissa tätä saattaa ilmetä kuvattavan kohteen raja-alueilla, joilla osa kohteista ei tule valokuvion valaisemaksi ja tästä syystä niillä alueilla syvyysdata jää puutteelliseksi. Kulkuakatekniikkaa hyödyntävissä kameroissa eri etäisyyksiltä heijastuvat signaalit saattavat aiheuttaa päällekkäisiä signaaleja, mikä johtaa virheellisiin mittaustuloksiin. Näitä mittavirheitä kutsutaan myös lentäviksi pikseleiksi. (Kolb ym. 2015, 12.)

### **Valon hajonta**

Koska molemmat tekniikat heijastavat valoa kuvattavaan kohteeseen ja mittaavat kohteesta palautuvaa valoa, niin valoa heijastavat ja läpikuultavat kohteet voivat aiheuttaa mittavirheitä. Tämä johtuu siitä, että valo ei heijastukaan suoraan takaisin sensorille vaan saattaa heijastuksen tai läpikuultavuuden takia kulkea pidemmän matkan takaisin sensorille tai heijastua muualle ja tämä voi myös aiheuttaa epäyhtenäisyyksiä syvyysdatassa. (Kolb ym. 2015, 13–14.)

### **Liikkuva kohde tai kamera**

Kuvatessa liikkuvaa kohdetta tai liikkeessä olevalla kameralla saattaa mittaustuloksissa ilmetä liikkeestä johtuvia mittausrvirheitä (Kolb ym. 2015, 15).

## **3.4 Volumetristen videoiden kuvausprosessi**

Tässä kappaleessa kerrotaan volumetristen videoiden kuvausprosessin kulusta ja työvaiheista. Prosessin kulussa ja yksityiskohdissa saattaa olla eroavaisuuksia käytetyistä laitteista ja ohjelmistoista riippuen, mutta peruseriaatteet pysyvät samankaltaisina.

### **3.4.1 Kameroiden kalibrointi**

Syvyyskameroiden sensoreiden mittaustulosten laatu saattaa ajan ja käytön myötä heikentyä. Muun muassa altistus suurille lämpötilanvaihteluille, iskuille ja värinöille saattaa heikentää sensorin kalibrointiastetta. Esimerkiksi Intel RealSense D400-sarjan kameroissa tämä ilmenee yleensä syvyysdatassa olevana kohinana. Kohina on helppo havaita kuvaamalla tasaista pintaa ja tarkastelemalla kuinka paljon korkeuseroja mittaustuloksissa on. Mitä huonompi sensorin kalibrointiaste on, sitä suurempia korkeuseroja voidaan havaita mittaustuloksissa ja ennen pitkää kalibroimattoman sensorin mittaustuloksissa alkaa ilmenemään "reikiä" niissä kohdissa, joista sensori ei enää kykene tuottamaan mittaustuloksia. (Dorodnicov, Grunnet-Jepsen, Khuong, Mulla, Sweetser & Tong 2020.)

### **3.4.2 Kuvausympäristön kalibrointi**

Kohteen täydellinen kolmiulotteinen videokuvaaminen vaatii useiden kameroiden käyttämistä, jotta kuvattavasta kohteesta saadaan tarpeeksi dataa kolmiulotteisen mallin luomiseksi. Jotta jokaisesta kamerasta saatu informaatio pystytään yhdistämään kolmiulotteiseksi malliksi, täytyy jokaisen kameran sijainti ja asento suhteessa kuvattavaan kohteeseen olla tiedossa. (Daras, Papachristou, Zarpalas & Zioulis 2018, 1.)



Perinteiset kalibroitavat käyttävät kuvausalueen keskelle asetettavia tulostettuja kuvioita, joista otetaan kuva kaikilla kameroilla. Näistä kuvista saadun informaation perusteella pystytään määrittelemään missä kohtaan kuviot ovat kunkin kameran ottamassa kuvassa. Yhdistämällä tämä tieto kameran sensorista saatuun syvyysdataan, saadaan tietoa kuvion sijainnista ja asennosta jokaiseen kameraan nähden. Vierekkäisten kameroiden mittaustulokset yhdistämällä pystytään arvioimaan vierekkäisten kameroiden sijainti ja asento toisiinsa nähden. (Daras ym. 2018, 2.)

Tällä tavalla suoritettu kalibrointi tuottaa yleensä tarkkoja kalibrointituloksia, mutta se vaatii myös paljon työtä, koska tarkkojen kalibrointituloksien saamiseksi kuvioita on liikuteltava kuvausalueen sisällä useita kertoja. Ongelmana on myös se, että mitä useampi kamera kuvausalueella on, sitä suurempi on myös mahdollisten kalibrointivirheiden todennäköisyys, koska eri kameroiden kalibrointituloksissa olevat virheet kertaantuvat eri kameroiden kalibrointituloksia yhdistäessä. (Daras ym. 2018, 2.)

Jotkut kalibroitiratkaisut käyttävät kaksiulotteisten kuvioden sijaan kuvausalueelle asetettua rakennelmaa kameroiden sijaintien määrittelemiseksi (Daras ym. 2018, 2). Esimerkiksi Alexiadis, Chatzitofis, Daras, Louizis, Zarpalas, Zioulis ja Zoidi (2017, 800) käyttivät omassa kuvausjärjestelmässään kuvausalueen keskelle asetettua pahvilaatikoista rakennettua rakennelmaa, jossa jokaisen pahvilaatikon kulmiin oli sijoitettu kalibrointiin käytetty kuvio. Daras ja muut (2018, 2–3) käyttivät omassa järjestelmässään vastaavaa tekniikkaa kuvausalueen kalibrointiin, mutta heidän käyttämässään kalibroitiratkaisussa ei rakennelmassa tarvitse olla kalibrointimerkkejä, vaan he hyödynsivät ratkaisussaan tekoälyä, joka kykeni päättelemään kameroiden sijainnin suoraan rakennelman geometrian perusteella.

### 3.4.3 Kuvaaminen

Korkealaatuisten volumetristen videoiden kuvaaminen vaatii tarkkojen kameroiden lisäksi myös tehokkaan tietokoneen. Esimerkiksi Girskaitė (2020a) ilmoittaa EF EVE:n volumetristen videoiden kuvaussovelluksen suositelluiksi järjestelmävaatimuksiksi vähintään 3.0 GHz Intel i9 prosessorin, 16 gigatavua muistia sekä vähintään NVIDIA Geforce GTX 1080-sarjan näytönohjaimen. Heidän sovelluksensa kykenee kuvaamaan

vain kahdella kameralla per tietokone, joten yli kahden kameran käyttö vaatii useamman tietokoneen käyttämistä. (Girskaite 2020a.)

Yksittäinen syvyyskamera tuottaa niin paljon dataa, että useamman syvyyskameran yhteiskäyttö samalla tietokoneella vaatisikin useamman USB-ohjainkortin asentamista tietokoneeseen. Mikäli kamerat ovat yhdistettynä tietokoneeseen USB-jatkojohdoilla, tulee käyttää korkealaatuisia jatkojohtoja ja pidemmillä etäisyyksillä jopa signaalinvahvistimille voi ilmetä tarvetta. (Daras, Domanoglou, Karakottas, Papachristou & Sterzentsenko 2019, 4.)

#### 3.4.4 Kuvatun materiaalin prosessointi

Volumetric capturessa käytetyissä kameroista saatava data jaetaan yleensä kolmeen eri osaan: värikuva, syvyyskartta ja harva pistepilvi. Harva pistepilvi on lista pisteistä kolmiulotteisesta ympäristöstä ja niiden väri-informaatiosta, kun taas syvyyskartta pitää sisällään pelkästään informaation kuvattavan kohteen etäisyydestä kameran sijaintiin suhteutettuna. Harva pistepilvi voidaan myös luoda värikuvasta ja syvyyskartasta saadun informaation perusteella. (Porter & Porter 2018, 5.)

Kun pistepilvi on luotu niin sen perusteella voidaan luoda kolmiulotteinen malli. Mallin luomiseen on kehitetty erilaisia tapoja. Bernardini, Mittleman, Rushmeier, Silva ja Taubin (1999) kehittivät pistepilvestä kolmiulotteisen mallin luomiseen Ball-Pivoting Algoritmin. Tässä algoritmista käyttäjä määrittelee halkaisijan virtuaaliselle pallolle, jota liikutetaan pistepilven alueella, kunnes löydetään kohta, jossa kolme pistepilven pistettä jäävät kaikki pallon sisälle ilman että pallon sisälle jää muita pisteitä. Näiden pisteiden perusteella saadaan luotua kolmion muotoinen polygoni, jota käytetään kolmiulotteisen mallin luomisen aloituspisteenä. Tästä kolmiosta valitaan sitten kaksi kulmaa, joiden ympäri virtuaalista palloa pyöritetään siten, että pallon ulkoreunat ovat jatkuvasti kosketuksissa kulmiin ja kun pallo osuu taas johonkin syvyyskartassa olevaan pisteeseen, syntyy tämän uuden pisteen sekä valittujen kulmien välille uusi kolmio. Tätä prosessia toistetaan, kunnes kaikki mahdolliset kolmion sivut on käyty läpi. Mikäli joiltakin kolmion sivuilta ei löytynyt pisteitä, joihin pallo olisi osunut, niin

pallon kokoa suurennetaan ja prosessi käydään niiltä osin uudestaan läpi. Tätä toistetaan niin kauan, että jokainen kolmio on yhdistetty toisiinsa. (Mittleman ym. 1999, 349.)

Bolitho, Hoppe ja Kazhdan (2006) käyttivät omassa ratkaisussaan lähestymistapaa, jossa hyödynnettiin mallin generoimiseen spatiaalista Poisson-prosessia. Heidän ratkaisunsa käsittelee kaikki pisteet yhtä aikaa hyödyntämättä heuristista avaruudellista osiointia tai yhdistämistä. Tämä tekee siitä erityisen vastustuskykyisen datassa ilmeneville virheille. Käyttämällä kehittämäänsä algoritmia julkisesti saatavilla olevan datan mallintamiseen he saivat aikaisempien yksityiskohtaisempia tuloksia kuin mitä aikaisemmillä tekniikoilla oli ollut mahdollista saavuttaa. (Bolitho ym. 2006, 1.)

#### 3.4.5 Vienti ja toistaminen

Koska volumetriset videot ovat vielä niin uusi aihealue, ei volumetriselle videolle ole tämän opinnäytetyön kirjoitushetkellä olemassa vielä standardeja. Aiheesta on kuitenkin meneillään useita standardisointiprosesseja. ISO:n ja IEC:n MPEG työryhmä on kehittämässä MPEG-I standardia, jossa tulee olemaan määrittäykset arkkitehtuurille, järjestelmille, videolle, äänelle, pistepilville ja myös tarvittava metadata, mittarit ja käyttöliittymät immersivisen mediasisällön verkkopohjaiselle käsittelylle. Prosessi aloitettiin jo vuonna 2015 ja sen on tarkoitus olla teknisesti valmis vuoden 2020 loppuun mennessä. (Boyce, Peng, Stockhammer & Wien 2019, 5–11.)

Koska volumetriselle videolle ei ole standardeja ovat alan yritykset kehittäneet omia ratkaisujaan videoiden vientiin ja toistamiseen. Esimerkiksi EF EVE Creator sovelluksesta on mahdollista tallentaa sovelluksessa luotu volumetrinen video OBJ-, PLY-, PLY.GLTF-, OBJ.GLTF-muodossa (Girskaitė 2020b). Toinen markkinoilla oleva ohjelmisto Depthkit mahdollistaa volumetrisen videon OBJ-muodossa tai heidän omassa MP4-videoformaattissansa (Export Formats 2019). MP4-videoformaatti voidaan toistaa Dethkit.js web-videosoittimellansa tai vaihtoehtoisesti Unitylla heidän omalla lisäosallaan (Depthkit.js 2018; Export Formats 2019).

### 3.5 Volumetric capture markkinakatsaus

Volumetrinen videoiden markkinoiden arvo vuonna 2020 on arviolta 1,4 miljardia Yhdysvaltain dollaria ja niiden uskotaan kasvavan 5,8 miljardin Yhdysvaltain dollarin arvoiseksi vuoteen 2025 mennessä 32,8 prosentin keskimääräisellä vuosittaisella kasvutahdilla. Markkinoiden kasvun merkittävimpiä tekijöinä pidetään erilaisten AR/VR-laitteistojen ja sovellusten yleistymisen, sekä kasvanut kysyntä paremmille katseluelämyksille etenkin urheilu-, peli-, tapahtuma- ja viihdesegmenteillä. Muita suuria markkinasegmenttejä ovat lääketiede, opetus ja koulutus sekä markkinointi. (Volumetric Video Market - Global Forecast to 2025 2020.)

Markkinoiden hallitsevat yritykset ovat Microsoft Corporation, Intel Corporation, Facebook Inc., Google LLC ja 4D View Solutions. Markkinoilla on tapahtunut myös paljon yrityskauppoja ja yritysyhteistyötä. Alan yritykset ovat tehneet useita ratkaisuja etenkin urheilu ja viihdesegmenteille. Muun muassa Microsoft on kehittänyt volumetric capture studion nimeltä Dimension yhteistyössä Hammerheadin kanssa. Helmikuussa 2018 MR Museo Kiotossa Japanissa loi yhdessä HakuHodo-VRAR:n kanssa volumetrinen videon avulla museokierroksen, jossa zen-munkki kertoi museon kävijöille tempelin taideteoksista. Intel on myös kehittänyt elokuvateollisuuden käyttöön soveltuvan volumetric capture studion yhdessä Turner Sportsin kanssa. Intel teki myös yhteistyötä National Football Leaguen kanssa tarjotakseen immersivisiä kohokohtia ja uusintoja vuoden 2019 Super Bowlistä Atlantan Mercedes-Benz stadionilla käyttäen Intelin True View teknologiaa. (Mt.)

Useat nuoret kasvuhakuiset yritykset, alan suuryritykset sekä muutamat tutkimuslaitokset tekevät myös tutkimusta volumetrinen videon teknologioista. Tämä tulee auttamaan volumetrinen video markkinoiden kasvua tulevien vuosien aikana. Alan johtava tutkimuslaitos on Saksassa sijaitseva Fraunhofer Heinrich Hertz Institute. (Mt.)

Urheilu-, tapahtuma- ja viihdesegmenteille tehtyjen ratkaisujen ja sovellusten uskotaan toimivan pääasiallisina volumetrinen videoiden markkinoiden kasvattajina seu-

raavan viiden vuoden aikana ja niiden uskotaan myös kasvavan suurimmalla keskimääräisellä vuosittaisella kasvutahdilla. Tämä johtaa siihen, että ne tulevat olemaan volumetristen videoiden markkinoiden hallitsevia segmenttejä vuoteen 2025 mennessä. (Mt.)

Tampereelle on myös kehitteillä Suomen ensimmäinen volumetriseen video kuvaukseen kykenevä studio. Studio rakennetaan tampereen yliopiston Hervannan kampuksella olevaan immerstiivisten visuaalisten teknologioiden tutkimusinfrastruktuuri CIVIT:n tiloihin. Hanke on saanut rahoitusta Tampereen yliopistolta ja Suomen Akatemialta. (The first Volumetric Capture Studio in Finland to be built in CIVIT 2020.)

## **4 Volumetristen videoiden kuvauskärjestelmän kehittäminen**

Tässä kappaleessa käydään läpi volumetristen videoiden kuvausjärjestelmän kehitystyön vaiheet. Tutkimuksessa käydään aluksi läpi olemassa olevat kuvauslaitteet, perustellaan laitteistohankinnat, vertaillaan olemassa olevia kuvausohjelmistoja sekä tutustutaan niillä kuvatun datan käsittelyyn ja toistamiseen. Tämän jälkeen käydään läpi oman volumetristen videoiden kuvausohjelmiston kehittämistyön vaiheet. Kehitetty ohjelmisto kattaa materiaalin kuvaamisen, siivoamisen sekä toistamisen. Lopuksi testataan järjestelmällä kuvattujen videoiden toimivuutta eri alustoilla.

### **4.1 Laitteisto**

Volumetristen videoiden kuvausjärjestelmän kehittämistyö aloitettiin kartoittamalla se, millaista laitteistoa järjestelmää varten tarvitaan. Hankintoja tehdessä tutustuttiin markkinoilla oleviin vaihtoehtoihin ja vertailtiin niiden ominaisuuksia. Järjestelmän käyttöön hankittavia laitteita valitessa kiinnitettiin erityisesti huomiota käytävissä olevaan budjettiin sekä siihen, miten hyvin markkinoilla olevat kuvausohjelmistot tukivat kyseisiä laitteita.

#### 4.1.1 Syvyyskamerat

Järjestelmän käyttöön hankittavia kameroita vertaillaessa otettiin tarkempaan tarkasteluun kolme eri syvyyskameravalmistajaa: Intel, Microsoft ja Orbbec.

Intelillä on oma RealSense tuoteperhe, johon kuuluu useita eri teknologioita hyödyntäviä kameroita. Intelillä on myös oma avoimen lähdekoodin ohjelmistokehityspaketti, joka tukee useita eri alustoja sekä ohjelmointikieliä. Ohjelmistokehityspaketti on saatavilla Windows, Linux sekä Android käyttöjärjestelmille.

Microsoft valmistaa Kinect syvyyskameroita, jonka viimeisin versio Azure Kinect on vielä kehitysvaiheessa, mutta siitä on saatavilla kehityspaketti. Kirjoitushetkellä tämä paketti on kuitenkin saatavilla vain Yhdysvalloissa, Kiinassa, Saksassa, Japanissa ja Isossa-Britanniassa. Kinectille on myös saatavilla avoimen lähdekoodin ohjelmistokehityspaketti sekä Linux, että Windows käyttöjärjestelmille. Ohjelmistokehityspaketista löytyy tuki useille eri ohjelmointikielille kuten C# ja C++.

Orbbecilla on Astra tuoteperhe, johon kuuluu useita eri syvyyskameramalleja. Orbbecilla on myös oma ohjelmistokehityspaketti, joka on saatavilla Windows, Linux ja Android käyttöjärjestelmille.

Volumeriseen videokuvaukseen parhaiten soveltuvaksi vaihtoehdoksi osoittautui Microsoftin Azure Kinect kamera, jossa oli muihin kameroiden verrattuna paras värikuvan ja syvyysdatan laatu. Se oli myös olemassa olevien volumetristen videoiden kuvaus ohjelmistojen suositteloima kamera. Azure Kinect ei kuitenkaan ollut vielä kirjoitushetkellä saatavissa Suomessa, joten päädyttiin valitsemaan Intelin RealSense D415 syvyyskamera. Valintaan vaikutti eniten se, että kamera oli yhteensopiva olemassa olevien kuvausohjelmien kanssa sekä kameroiden hyvä yhteensopivuus eri alustojen, ohjelmointikielien ja käyttöjärjestelmien kanssa.

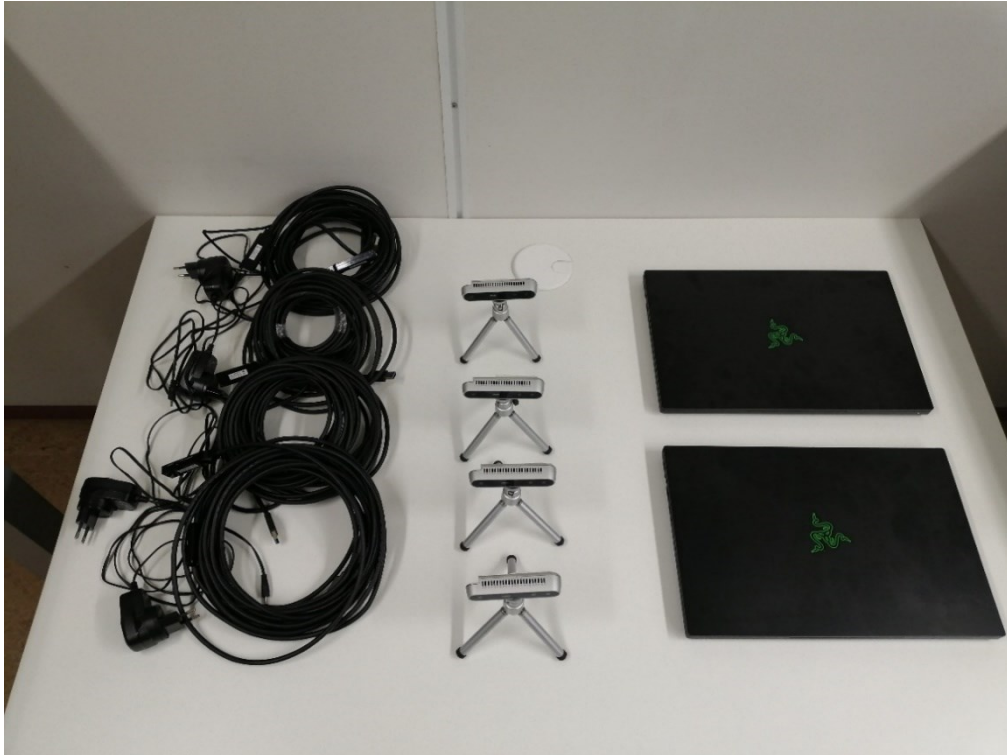
Kameroita tilattiin neljä kappaletta, koska tämä oli Volumetric Capture kuvaussovelluksen kehittäjä EF EVE:n suosittelema kameramäärä kuvattavan kohteen täydelliseen 360-asteen kuvaamiseen.

#### 4.1.2 Tietokoneet

Syvyyskamerat tuottavat värikuvan lisäksi syvyysdataa, joten kameroista tuleva datamäärä on huomattavasti suurempi kuin mitä saadaan perinteisestä videokamerasta. Tästä syystä on datan tallennusta ja käsittelyä varten suositeltavaa käyttää tehokasta tietokonetta. Esimerkiksi Volumetric Capture kuvaussovelluksen kehittäjä EF EVE suosittelee sovelluksen ajamista tietokoneessa, jossa on vähintään 3.0 GHZ kellotaajuudella oleva Intelin i9-sarjan prosessori tai vastaava, vähintään 16 gigatavua muistia, SSD-levy sekä NVIDIA GeForce GTX1080-sarjan näytönohjain tai vastaava. Koska kuvausjärjestelmästä oli tarkoitus kehittää mahdollisimman liikuteltava, ja koska EF EVE ei tue kuin kahden kameran käyttämistä yhdellä tietokoneella, päädyttiin hankkimaan kaksi kappaletta kannettavia Razer Blade Base tietokoneita.

#### 4.1.3 Oheislaitteet

Tietokoneiden ja kameroiden lisäksi hankittiin kuvausjärjestelmää varten erilaisia oheislaitteita. Intel RealSense D415 syvyyskameroiden mukana tullut USB-johto ei lyhyen pituutensa vuoksi soveltunut volumetriseen videokuvaukseen, joten kuvausjärjestelmää varten hankittiin jokaiselle kameralle aktiivinen kymmenen metriä pitkä USB-jatkokaapeli. Kameroiden mukana tuli myös vain pienet pöytäjalustat, mutta kameroista löytyi kuitenkin standardi 1/4-20 UNC kierteellä oleva kamerajalustan kiinnityspiste, joten kameroita varten tilattiin korkeammat lattialle asetettavat kolmijalkaiset kamerajalustat. Kuvausjärjestelmää varten tarvittiin myös valoja ja jatkojohtoja, mutta näitä löytyi jo valmiiksi toimeksiantajalta.



Kuvio 1. Kuvausjärjestelmää varten hankittua laitteistoa

## 4.2 Olemassa olevat ohjelmistot

Laitehankitojen jälkeen tutustuttiin markkinoilta löytyviin valmiisiin volumetriseen videokuvaukseen soveltuviin kuvausohjelmiin ja -ohjelmistoihin. Koska kuvausjärjestelmästä oli tarkoitus tehdä edullinen ja liikuteltava niin keskityttiin etsimään mahdollisimman edullisia kuvausohjelmia. Soveltuvia ohjelmia ei ollut juurikaan tarjolla, mutta markkinoilta löydettiin kaksi ohjelmistoa, jotka otettiin tarkempaan vertailuun: Scatterin kehittämä Depthkit sekä EF EVE:n kehittämä Volumetric Capture.

Ohjelmistoja tutkittaessa tarkasteltiin seuraavia ominaisuuksia: hinta, kuvatun materiaalin laatu, -jälkikäsittely, tuetut tiedostomuodot, yhteensopivuus eri alustojen ja ohjelmistojen kanssa sekä kuvatun materiaalin toistaminen.



### 4.2.1 Depthkit

Depthkit on suunnattu erityisesti videotuotannon käyttöön ja sillä voidaan yhdistää perinteisestä videokameralla kuvattu videokuva ja syvyyskameralla kuvattu syvyysdata korkearesoluutioiseksi volumetriseksi videoksi. Sovellus on kirjoitushetkellä vielä kehityksessä, mutta sovelluksesta on saatavilla beetaversio. Tällä hetkellä Depthkitillä ei voi kuvata kuin yhtä syvyyskameraa käyttäen, mutta useamman syvyyskameran yhtäaikaisen käytön tuki on kehitteillä. Kirjoitushetkellä sovellus on saatavilla vain Windows 10 käyttöjärjestelmälle mutta Mac OS X versio on kehitteillä.

#### **Hinta**

Scatter tarjoaa Depthkit-sovellustaan kuukausimaksullisena palveluna josta on saatavilla eri tasoisia lisenssejä eri käyttötarkoituksiin. Depthkitistä on myös saatavilla ilmainen kokeiluversio jolla pystyy testaamaan sovelluksen ominaisuuksia. Ilmaisessa versiossa tallentaminen on rajoitettu 30 sekuntiin ja ohjelmasta pystyy viemään ulos maksimissaan viiden sekunnin mittaisen videon. Ainoa kameran asetus jota ilmaisversiossa on mahdollista säätää, on värikuvan resoluutio.

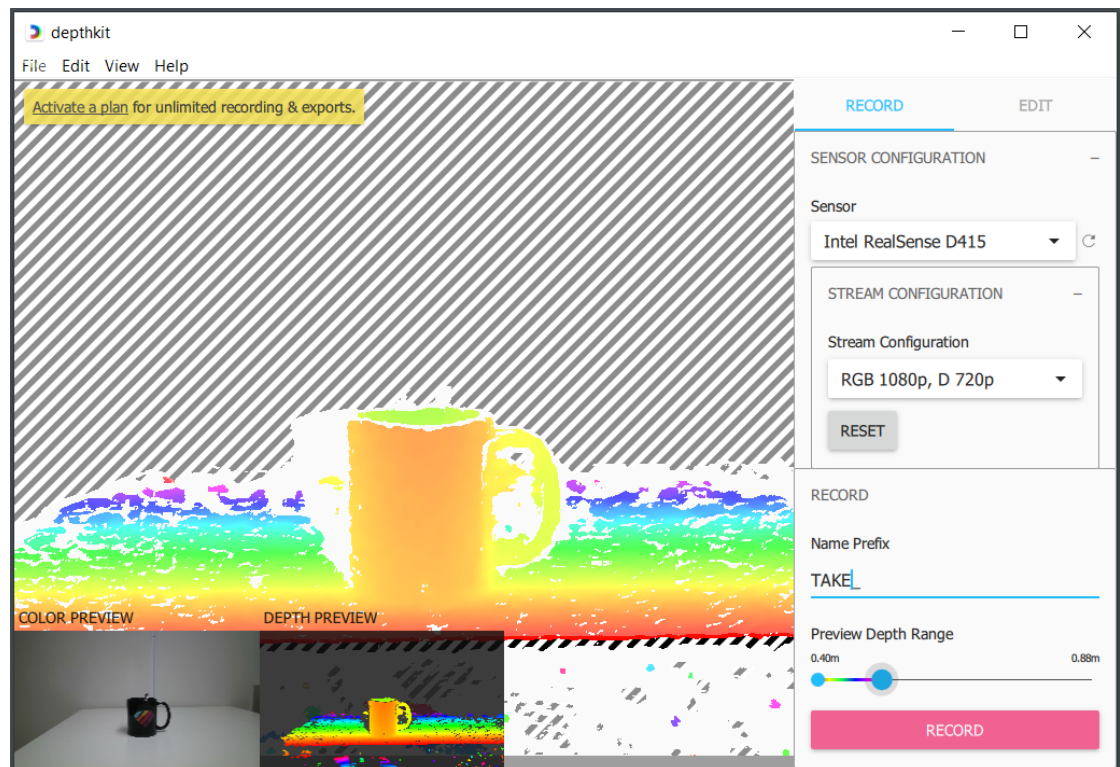
Halvimmalla maksullisella Depthkit Record lisenssillä on mahdollista kuvata vain käyttäen Intelin RealSense kameroita tai Microsoftin Kinect for Windows V2 kameraa. Tämä lisenssi maksaa kirjoitushetkellä 9,99 \$/kuukausi. Mikäli haluaa kuvata Depthkitin suosittelemalla Azure Kinect kameralla, niin tarvitsee Depthkit Pro lisenssin joka maksaa kirjoitushetkellä 39,99 \$/kuukausi. Mikäli haluaa yhdistää videokameran ja Azure Kinect syvyyskameran, niin siihen tarvitaan Depthkit Cinema lisenssin joka maksaa kirjoitushetkellä 399 \$/kuukausi.

Yllämainittujen lisenssitason lisäksi on Depthkit sovellukseen tulossa myös Depthkit Studio lisenssi, johon Scatter lupaa mahdollisuuden kuvata jopa kymmenellä kameralla yhtäaikaisesti. Kirjoitushetkellä ei tämän lisenssin muista ominaisuuksista tai hinnasta ollut kuitenkaan vielä lisätietoja saatavilla.

Koska kirjoitushetkellä Depthkitistä ei ollut saatavilla versioita, joka tukisi usealla syvyyskameralla kuvaamista, päätettiin testata vain Depthkitin ilmaisversiota. Tämän vuoksi maksullisista versioista löytyviä ominaisuuksia ei voitu testata.

## Kuvaaminen

Depthkitillä kuvaaminen osoittautui erittäin helpoksi. Käyttäjän tarvitsi vain kytkeä kamera tietokoneeseen ja käynnistää kuvaussovellus joka tunnisti kamerasensorin automaattisesti. Tämän jälkeen on mahdollista säätää kamerasensorin värivälit ja resoluutiota sekä määrittämään sen miten kaukana tai lähellä olevat asiat näkyvät tallenteessa (ks. kuvio 2). Etäisyyden määrittely tallentaa kuitenkin myös kauempana olevat kohteet, mutta ei näytä niitä esikatselussa. Ainakaan ilmaisversiossa ei pystytty määrittämään nauhoituksen pituutta ennen kuvauksen aloittamista tai asettamaan ajastinta kuvauksen aloittamiselle.

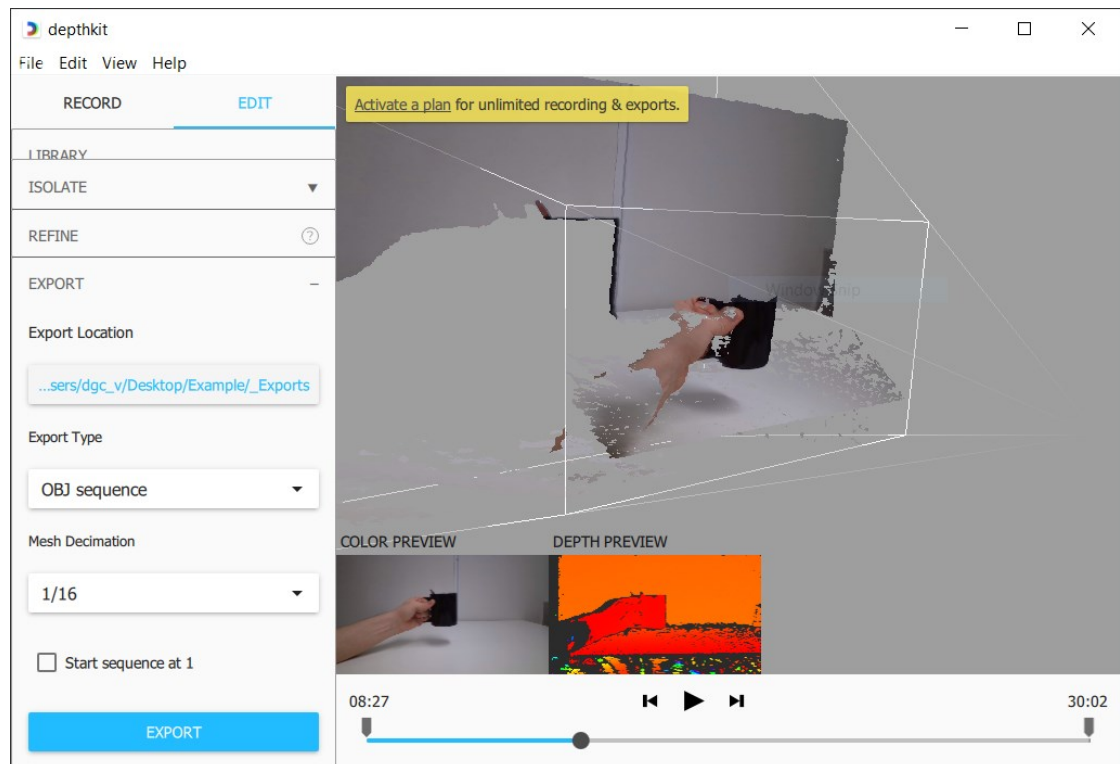


Kuvio 2. Tallennusetäisyyden säätäminen Depthkitissä

## Jälkikäsitteily

Kun video on saatu kuvattua pystyy Depthkitissä määrittelemään erilaisia jälkikäsitteilyasetuksia. Jälkikäsitteilyvaiheessa voidaan säätää uudelleen sitä, kuinka kaukana olevat kohteet tulevat mukaan lopulliseen ulosvietävään tuotokseen. Tämän lisäksi syvyysdataa on mahdollista tarkentaa ja muokata käyttämällä väridatasta generoitua mustavalkoista maskivideota. Ohjelma ei kuitenkaan generoi maskivideota automaattisesti vaan käyttäjän on luotava maski itse käyttäen erillistä videoeditointiohjelmaa, kuten Adobe After Effectsiä tai vastaavaa.

Jos kuvatun videon tallentaa sarjana kolmiulotteisia objekteja, on ohjelmassa mahdollista vähentää kolmiulotteisen mallin geometriamäärää käyttäen ohjelmasta löytyvää desimointityökalua (ks. kuvio 3). Desimoinnin määrää ei kuitenkaan voinut valita vapaasti, vaan valittavana oli ainoastaan viisi ennaltamääritettyä arvoa.



Kuvio 3. Näkymä Depthkitin editointi-ikkunasta

## Ulosvienti

Depthkitissä kuvatun datan voi viedä ulos kolmessa erilaisessa muodossa: Sarjana kolmiulotteisia malleja, videona tai kuvasarjana. Kolmiulotteisena mallina tallentaessa jokaisesta ulos vietävästä kuvasta tallentuu obj-tiedostomuodossa oleva kolmiulotteinen malli, png-tiedostomuodossa oleva tekstuuri sekä mtl-tiedostomuodossa oleva materiaalitiedosto.

Videona tallentaessa yhdistää Depthkit syvyysdatan ja väridatan yhdeksi videotiedostoksi (ks. kuvio 4). Videon lisäksi tallentaa Depthkit myös png-tiedostomuodossa olevan kuvatiedoston sekä txt-tiedoston, jossa on JSON-muodossa olevaa dataa, joita tarvitaan videon toistamiseen kolmiulotteisena Depthkitin tukemissa sovelluksissa. Kuvasarjana tallentaminen toimii muuten kuten videona tallentaminen, mutta videon sijaan tallentuu jokainen videon kehys omana png-tiedostona.



Kuvio 4. Syvyys- ja väridata yhdistettynä yhdeksi videoksi

### **Kuvatun materiaalin toistaminen**

Depthkit tarjoaa kaksi erilaista ratkaisua sillä tuotetun volumetrisen videon toistamiseen: WebGL-sovellus sekä Unity-lisäosa. WebGL-sovellus mahdollistaa Depthkitillä kuvatun kolmiulotteisen videon lisäämisen internet-sivustolle. Sovellus vaatii toimiakseen Three.js-kirjaston.

Unity-lisäosalla on mahdollista lisätä Depthkitillä kuvattuja videoita Unity-projekteihin. Kirjoitushetkellä tuetut Unity-versiot ovat 2017.4 LTS, 2018 ja 2019. Versiossa 2017.4 LTS on tosin ongelmia värien tarkkuudessa, jos projektissa käytetään lineaarista väriavaruutta. Depthkitin Unity-lisäosan tukemat julkaisualustat ovat kirjoitushetkellä PC, Mac, iOS ja Android.

#### **4.2.2 EF EVE Volumetric Capture**

EF EVE tarjoaa valmiin ohjelmistoratkaisun volumetriseen videotuotantoon. Heidän Volumetric Capture tuoteperhe sisältää kuvaussovelluksen, editointisovelluksen sekä verkossa olevan alustan kuvatun materiaalin toistamiseen. Toisin kuin Depthkitillä, on EF EVE:n kuvaussovelluksella mahdollista nauhoittaa volumetrista videota käyttäen useampaa kuin yhtä syvyyskameraa. Kirjoitushetkellä EF EVE:n sovellukset ovat saatavilla vain Windows 10 -käyttöjärjestelmälle. EF EVE:n kuvaussovelluksen tukemat syvyyskamerat ovat Microsoftin Azure Kinect ja Kinect V2 sekä Intelin RealSense D415 ja D435.

### **Hinta**

EF EVE:n sovellukset ovat Depthkitin tapaan kuukausimaksullisia, mutta huomattavasti kalliimpia. Kirjoitushetkellä ohjelmistosta ei myöskään ollut saatavilla ilmaista kokeiluversiota. EF EVE:n julkaisualustaa voi kuitenkin käyttää ilmaiseksi rajoitetuin ominaisuuksin. Valmiiden lisenssitasojen lisäksi EF EVE tarjoaa asiakkaille heidän tarpeidensa mukaan räätälöityjä ratkaisuja.

EF EVE:n kuvaussovelluksesta on saatavilla kaksi eri tasoista lisenssiä. Halvempi lisenssitaso on nimeltään Pro ja se mahdollistaa kuvaamisen maksimissaan kahdella samaan tietokoneeseen liitetyllä kameralla. Kirjoitushetkellä lisenssin hinta on 39 \$

kuukaudessa. Jos haluaa kuvata suuremmalla määrällä kameroita niin tarvitsee Business tason lisenssin jonka hinta kirjoitushetkellä on 89 \$ kuukaudessa. Business tason lisenssi mahdollistaa myös kameroiden sijainnin automaattisen kalibroinnin. Yksi Business tason lisenssi tosin mahdollistaa vain kuvaamisen kahdella kameralla per tietokone, joten jos halutaan kuvata useammalla kameralla niin tarvitaan myös useampi lisenssi ja useampi tietokone.

EF EVE:n editointisovelluksesta on saatavilla kolme eri lisenssitasoja. Halvin lisenssitaso on nimeltään PRO ja sen hinta kirjoitushetkellä on 149 \$ kuukaudessa. Lisenssi mahdollistaa vain maksimissaan kahdella kameralla kuvatun materiaalin editoinnin. Neljällä kameralla kuvatun materiaalin editointiin tarvitsee hankkia Premium tason lisenssi, jonka hinta kirjoitushetkellä on 349 \$ kuukaudessa. Kolmas lisenssitaso on nimeltään Business ja sen hinta kirjoitushetkellä on 749 \$ kuukaudessa ja se mahdollistaa maksimissaan kymmenellä kameralla kuvatun materiaalin editoinnin.

Koska projektia varten oli hankittu neljä kameraa, niin hankittiin testausta varten yhdeksi kuukaudeksi kaksi kappaletta Business tason lisenssiä kuvaussovelluksesta, sekä yksi Premium tason lisenssi editointisovelluksesta.

### **Kuvaaminen**

EF EVE:n kuvaussovelluksella kuvaaminen oli tehty melko helpoksi. Premium tason lisenssi mahdollisti kahden tietokoneen yhtäaikaisen ohjaamisen yhdeltä palvelimeksi määritellyltä tietokoneelta käsin. Premium tason lisenssissä kameroiden sijainti ja asento piti myös olla mahdollista kalibroida ArUco kalibrointimerkkien avulla, mutta tätä ominaisuutta ei saatu toimimaan kuvauksessa käytetyillä Intel RealSense D415 kameroilla. Ongelmaa yritettiin selvittää yhdessä ohjelmistovalmistajan kanssa, mutta kalibrointia ei kuitenkaan saatu toimimaan. Tämä tarkoitti sitä, että kameroiden asento ja sijainti jouduttiin määrittelemään käsin käyttäen ohjelmasta löytyviä säätimiä tai vaihtoehtoisesti jälkikäsitteilyn yhteydessä editointisovelluksessa. Ohjelmassa oli myös mahdollista rajata kuvausalueen rajat, jolloin rajojen ulkopuolelle jäävää materiaalia ei tallenneta. Myös äänen tallentaminen videokuvan yhteydessä oli sovelluksessa mahdollista käyttäen

tietokoneeseen liitettyä mikrofonia. Sovelluksessa pystyi myös säätämään erilaisia nauhoitusasetuksia, kuten resoluutiota, kuvataajuutta sekä määrittelemään erilaisia suotimia, joilla pystyi siistimään kameran syvyysdatassa olevia mittausvirheitä. Kuvauksen pystyi aloittamaan manuaalisesti tai käyttäen ennaltamääriteltä kolmen, viiden tai kymmenen sekuntin ajastinta. Kuvauksen pystyi määrittelemään automaattisesti joko viiden, kymmenen tai kahdenkymmenen sekunnin pituiseksi tai jättämään pituuden ennaltamäärittelemättömäksi, jolloin kuvaus oli pysäytettävä manuaalisesti. Useamalla tietokoneella kuvatessa oli Business tason lisenssissä myös mahdollista lähettää kuvattu materiaali suoraan verkon yli palvelimeksi määritellylle tietokoneelle.

Sovelluksella kuvatessa huomattiin, että sovellus nauhoittaa huomattavasti isompia datamääriä kuin Depthkit. Depthkitillä nauhoittaessa 1280 x 720 resoluutiolla kolmekymmentä kuvaa sekunnissa tuli dataa noin 400 megatavua, kun taas samalla resoluutiolla ja yhdellä kameralla nauhoittaessa EF EVE:n sovelluksessa tuli dataa yli 2 gigatavua. Sovellus tallentaa tiedostot myös heidän omaan tiedostomuotoon, joten kuvattun datan käsittelemistä varten on pakko käyttää EF EVE:n omaa editointisovellusta.

### **Jälkikäsitely**

Kuvauksen jälkeen kuvattu materiaali vietiin EF EVE:n editointisovellukseen, jossa pystyttiin yhdistämään usealla tietokoneella ja kameralla kuvattu materiaali yhdeksi kokonaisuudeksi. Sovelluksessa pystyi myös manuaalisesti säätämään kameroiden sijaintia ja asentoa toisiinsa nähden. Sovelluksessa oli useita eri asetuksia, joilla pystyi vaikuttamaan ulosvietävän materiaalin laatuun ja datamäärään. Toiminnot oli jaettu omiin välilehtiinsä sen mukaan mihin asioihin kyseisen välilehden toiminnot vaikuttavat. Monien työkalujen toiminnallisuutta ei kuitenkaan oltu selitetty ohjelmassa, eikä myöskään EF EVE:n omassa dokumentaatioissa, vaan ainut tapa selvittää joidenkin työkalujen vaikutus oli kokeilla työkalua eri asetuksilla ja katsoa mitä tapahtuu. Testauksien aikana huomasimme, että joillakin työkaluilla ei tuntunut olevan huomattavaa vaikutusta lopputulokseen, vaan vaikutti siltä kuin ne eivät olisi tehneet mitään ja ohjelma vaikutti muutenkin hyvin keskeneräiseltä. Eri asetusten ja työkalujen vaikutusta lopputulokseen oli mahdollista esikatsella ennen ohjelmasta

ulosvientiä käsittelemällä yksittäinen kehys kuvatusta materiaalista. Esikatselun avulla pystyttiin arvioimaan ulosvietävän materiaalin käsittelyyn menevä aika, tiedostokoko sekä se olisiko lopputulos laadultaan halutunkaltainen.

### **Ulosvienti**

Editointisovelluksesta pystyi ulosviemään kuvatun materiaalin kolmessa eri muodossa: obj, ply tai gltf. Gltf-tiedostomuoto oli tarkoitettu käytettäväksi vietäessä materiaalia EF EVE:n omalle julkaisualustalle. Gltf-tiedoston avaamista testattiin myös muissa gltf-tiedostoja tukevissa ohjelmissa, mutta tiedostoa ei saatu avattua näissä ohjelmissa. Tehdyissä testeissä obj-tiedostomuodossa ulosvienti osoittautui toimivan parhaiten silloin kun materiaali vietiin käytettäväksi muissa sovelluksissa, kuten 3D-mallinnusohjelmissa tai pelimoottoreissa. Obj-tiedostomuodossa ulosvietäessä jokaisesta valitusta kehyksestä tallentuu obj-tiedostomuodossa oleva 3D-malli, jpeg-tiedostomuodossa oleva tekstuuritiedosto sekä mtl-tiedostomuodossa oleva materiaalitiedosto.

Testien aikana huomattiin, että editointivaiheessa käytetyistä työkaluista ja asetuksista riippuen saattoi yksittäisen kehyksen ulosvientiin mennä useita, joskus jopa kymmeniä, minuutteja. Tämä tarkoitti sitä, että minuutin mittaisen nauhoitteen ulosvienti saattaa kestää useita kymmeniä tai jopa satoja tunteja.

### **Kuvatun materiaalin toistaminen**

Oman julkaisualustan lisäksi EF EVE:llä on oma lisäosa videoiden toistamiseen Unity- sekä Unreal-pelimoottoreissa. EF EVE:n omalla julkaisualustaa oli mahdollista testata ilmaiseksi, mutta tiedostokoko on rajoitettu ilmaisversiossa viiteenkymmeneen megatavuun. Alustaa yritettiin testata ilmaislisenssillä, mutta videoiden lähettäminen alustalle ei onnistunut. Alustassa vaikutti olevan muutenkin ongelmia sillä testien aikana suurimman osan ajasta alustalla olevia videoita ei pystynyt katselemaan.

Koska EF EVE:n oma julkaisualusta osoittautui epäluotettavaksi, hinnakkaaksi sekä projektin tarpeisiin soveltumattomaksi, niin tutustuttiin myös EF EVE:n Unity-lisäosaan. Unreal-lisäosaan ei tutustuttu tarkemmin, koska Unreal-pelimoottorin käytöstä ei ollut projektitiimillä kokemusta.



EF EVE:n Unity-lisäosa oli suunniteltu selkeästi korkealaatuisten volumetristen videoiden toistamiseen ja se oli suunniteltu käytettäväksi Unityn korkeatarkkuuksisissa HDRP-projekteissa. Koska tässä opinnäytetyöprojektissa oli tarkoitus saada järjestelmällä kuvatuista volumetrisista videoista mahdollisimman monikäyttöisiä, niin EF EVE:n Unity-lisäosa ei soveltunut tämän projektin käyttöön, mutta parhaiten se vaikuttaisi soveltuvan sellaisiin projekteihin, joissa tärkeintä on maksimaalinen kuvanlaatu.

### 4.3 Oman ohjelmiston kehittäminen

Kun EF EVE:n ja Scatterin tarjoamat volumetrisen videokuvauksen sovellukset oli testattu, niin tultiin siihen lopputulokseen, että kumpikaan niistä ei oikeastaan soveltunut projektin tarpeisiin. Tästä syystä päätettiin kehittää demo omasta volumetrisen videokuvauksen sovelluksesta. Kehitystysympäristöksi valittiin Unity, koska siitä oli projektitiimillä aikaisempaa kokemusta, sekä se soveltui hyvin kolmiulotteisen datan käsittelyyn sekä esittämiseen. Inteliltä löytyi myös Unity-lisäosa, jolla RealSense kameroita pystyi käyttämään Unity-pelimoottorissa.

#### 4.3.1 Kuvausalueen kalibrointi ja kameroiden synkronointi

Ensimmäinen asia, jota lähdettiin ratkaisemaan oli kameroiden sijainnin ja asennon määrittelemisen suhteessa kuvausalueeseen. Koska kuvausjärjestelmästä oli tarkoitus tehdä liikuteltava, pyrittiin kuvausalueen kalibroinnista tekemään mahdollisimman nopea ja yksinkertainen käyttöä.

Kameroiden sijainnin ja asennon määrittelemisen toteutettiin OpenCV-kirjastosta löytyvän ArUco-moduulin avulla. ArUco-moduuli mahdollistaa ArUco-merkkien tunnistamisen kameralla otetuista kuvista, sekä niiden sijainnin ja asennon määrittelemisen suhteessa kameraan. Merkin sijainnin määrittelemisen ArUco-moduulin avulla ei kuitenkaan osoittautunut tarpeeksi tarkaksi, mutta sillä pystyttiin määrittelemään tieto siitä, millä pikseleillä mikäkin ArUco-merkin kulma sijaitsee kameralta tulevasta väridatassa. Yhdistämällä tämä tieto kameroista saatuun tietoon

siitä, missä jokainen väridatan pikseli sijaitsee kolmiulotteisessa avaruudessa, pystyttiin ArUco-merkin sijainti ja asento määrittelemään tarkemmin.

Isoin ongelma joka huomattiin kuvatessa usealla kameralla samaa kohdetta yhtäaikaaisesti oli varmistaa se, että kamerat ottavat kuvia tasan samanaikaisesti. Intel RealSense kameroissa on mahdollisuus kameroiden synkronoimiseen erillisen synkronointikaapelin avulla, mutta Intel ei itse valmista kyseistä kaapelia vaan se pitää tehdä itse. Tämän projektin puitteissa ei kuitenkaan lähdetty tekemään omaa kaapelia vaan pyrittiin minimoimaan kameroiden kuvausajkojen välisiä eroja ohjelmallisesti ohjaamalla kameroita verkon ylitse käyttäen Unityn Multiplayer HLAPI:a, joka mahdollistaa komentojen lähettämisen tietokoneelta toiselle. Tästä huolimatta kamerat eivät aloita kuvaamista tasan samaan aikaan, vaan eri kameroiden välillä on eroja kuvanottoajassa. Tämä oli huomattavaa varsinkin suuremmilla resoluutioilla ja kuvataajuksilla kuvatessa. Testiemme perusteilla kuvatessa 640 x 360 pikselin resoluutiolla sekä 15 kuvaa sekunnissa kuvataajuudella saatiin käytössä olevalla laitteistolla paras mahdollinen kuvanlaatu ilman suuria eroja kameroiden välisissä kuvanottoajoissa.

Toinen testien aikana huomattu potentiaalisesti ongelmia aiheuttava asia oli kameroiden käyttö pitkillä USB-johdolla. Tavallisia USB-johdoilla testatessa kamerat eivät aina lähteneet päälle tai saattoivat sammua kesken kuvauksen. Koska kamerat ottavat virtansa suoraan USB-portin kautta ja lähettävät suuria määriä dataa samaa porttia käyttäen, niin pitkillä etäisyyksillä ilmeni tarve käyttää aktiivisia USB-jatkojohtoja.

#### 4.3.2 Kameroista tulevan datan tallentaminen

RealSense Unity-lisäosalla pystyi tallentamaan kameroista tulevaa dataa rosbag-tiedostomuodossa. Tämä ei kuitenkaan soveltunut projektin käyttötarkoitukseen, sillä jo pelkästään yhdestä kamerasta saatu datamäärä oli erittäin suuri. Tästä syystä kamerasta saatu data tallennettiin itsemääritettyyn tiedostomuotoon. Kameroista saatavan datan lisäksi tiedostoon tallennettiin kameran sijainti ja asento kolmiulotteisessa avaruudessa, sekä tieto kameran käyttämästä kuvataajuudesta.

Itse kameroista saatava pistepilvidata pilkottiin siten, että jokaisen kehyksen jokaisesta pisteestä tallennettiin sijainti jokaisella akselilla sekä väriarvo jokaiselta värikanavalta. Koska kamerasta saatava pisteen sijainnin arvo tulee 32-bittisenä liukulukuna ja väridata tulee 8-bittisinä kokonaislukuina niin yhtä pistettä kohti tulee dataa 120 bittiä. Tämä saatiin pienennettyä 72:een bittiin tallentamalla sijaintiarvot 16-bittisinä. Tämä teoriassa heikentää pistepilven pisteiden sijainnin tarkkuutta varsinkin pidemmällä etäisyyksillä, mutta koska kameroiden tarkkuus pitemmillä etäisyyksillä on muutenkin heikko, niin 120- ja 72-bittisiä tallenteita verratessa ei huomattu ihmissilmällä havaittavia eroja pistepilvien laadussa.

#### 4.3.3 Äänen tallentaminen

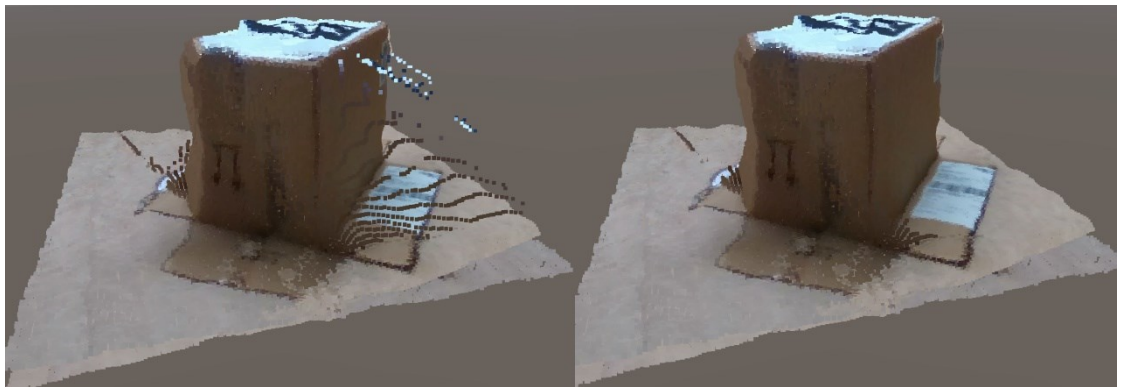
Äänen tallentaminen toteutettiin Unityn sisäänrakennetulla äänennauhoitustoiminnolla. Käyttöliittymään rakennettiin alavetovalikko, josta käyttäjä pystyy valitsemaan haluamansa äänitulolaitteen sekä halutun näytteenottotaajuuden. Mikäli käyttäjä on valinnut äänitulolaitteen niin äänen tallennus aloitetaan yhtäaikaaisesti videon tallennuksen kanssa, jotta ääni ja video tallentuvat samassa tahdissa. Tallennuksen aikana äänidata tallennetaan tietokoneen muistissa sijaitsevaan puskuriin ja kun käyttäjä lopettaa tallennuksen, niin äänidata tallentuu pakkaamattomana binääritiedostoon, johon tallennetaan kaikki ääninäytteet, näytteenottotaajuus sekä tieto äänikanavien määrästä.

#### 4.3.4 Kameroilla kuvatun datan siivoaminen ja yhdistäminen

Kun kuvaaminen on saatu päätökseen, siirretään kaikki kuvattu materiaali yhdelle tietokoneelle, jossa se avataan kuvaussovelluksen editointinäkyvässä. Editointinäkyvässä on mahdollista toistaa kaikilla kameroilla kuvattua materiaalia sekä nauhoitettua ääntä yhtenä kokonaisuutena. Kuvatun materiaalin siivoamista varten kehitettiin muutama erilainen työkalu: Rajausalue, reunojen siistiminen sekä pistepilven vokselisointi.

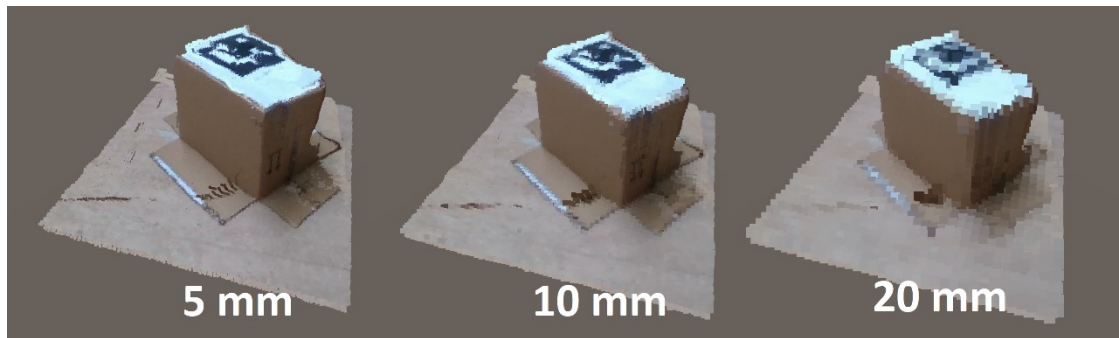
Rajausalueella pystytään rajaamaan pistepilvi siten, että vain kuvattava kohde jää rajausalueen sisäpuolelle. Tällöin rajausalueen ulkopuolelle jäävät pisteet poistetaan loppullisesta volumetrisesta videosta.

Reunojen siistimisellä pyritään vähentämään kuvattavan kohteen reuna-alueille tallentuneita virheellisiä pisteitä (ks. kuvio 5). Siistimisalgoritmi pyrkii etsimään jokaisen kameran kuvaamasta datasta kuvattavan kohteen reuna-alueet ja poistamaan ne pisteet, joiden etäisyys viereiseen pisteeseen ylittää ohjelmassa määritellyn arvon. Käyttäjä pystyy itse määrittelemään etäisyysarvon sekä sen, kuinka monta kertaa siistimisalgoritmi ajetaan.



Kuvio 5. Kuvattu kohde ennen reunojen siistimistä ja siistimisen jälkeen

Vokselisointialgoritmi jakaa rajausalueen pienempiin, käyttäjän määrittelemän kokosiin kuutioihin ja laskee keskiarvon jokaisen kuution sisällä olevan pisteen sijainnista sekä väristä. Tällä pystytään poistamaan pistepilvessä olevia päällekkäisiä pisteitä sekä vähentämään jäljellejäävien pisteiden määrää (ks. kuvio 6). Tämä on etenkin hyödyllistä silloin, jos volumetrista videota halutaan toistaa esimerkiksi mobiililaitteissa, joissa on rajallinen suorituskyky.

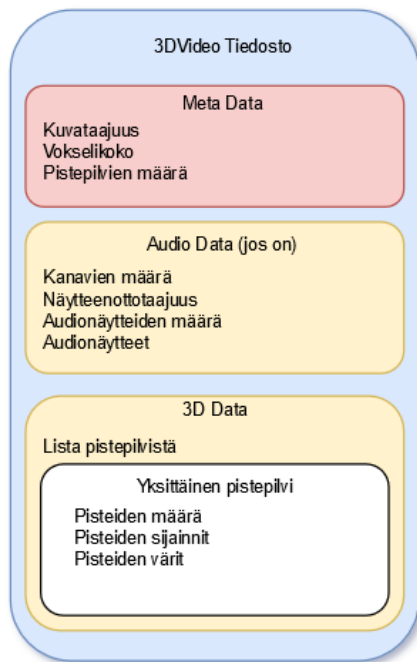


Kuvio 6. Vokselikoon vaikutus pistepilven laatuun

Ennen lopullisen videon ulosvientiä, pystyy käyttäjä esikatselamaan työkalujen asetusten vaikutusta painamalla esikatselupainiketta, jolloin ohjelma rakentaa esikatselupistepilven valitusta videon kohdasta. Tällä tavoin käyttäjä voi nähdä valitsemiensa työkalujen asetusten vaikutuksen nopeasti ilman että käyttäjän tarvitsee käsitellä koko videota kerralla.

Kun käyttäjä on tyytyväinen valitsemiinsa asetuksiin, voi hän valita miltä väliltä kuvatausta materiaalista hän haluaa luoda volumetrisen videon ja painaa ulosvientipainiketta luodakseen videon. Videon luontivaiheessa ohjelma käyttää hyödyksi Unityn Job Systemin tarjoamaa monisäikeisyysominaisuutta käsittelemällä jokaisen pistepilven omassa prosessorin säikeessä. Tämä mahdollistaa useiden pistepilvien käsittelyn yhtäaikaisesti sellaisissa tietokoneissa, joiden prosessorissa on useita säikeitä. Unityn Job System jättää yhden säikeen aina pääohjelman suorittamiseen, joten projektin käytössä olleilla tietokoneilla, joiden prosessorissa oli kaksitoista säiettä, pystyttiin käsittelemään yhtäaikaiseksi yksitoista pistepilveä.

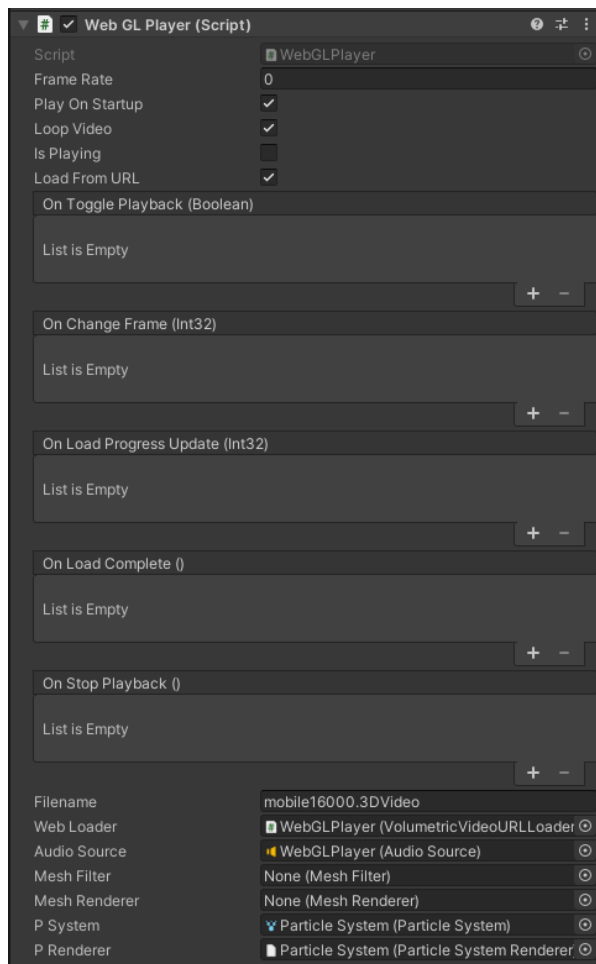
Pistepilvien lisäksi ohjelma tallentaa tiedostoon tiedon kuvataajuudesta, käytetystä vokselin koosta sekä videon pituudesta. Mikäli kuvauksen yhteydessä on tallennettu ääntä, niin ohjelma pakkaa tiedostoon myös tarvittavan äänidatan. Lopputuloksena syntyy yksi tiedosto, joka sisältää kaiken tarvittavan informaation tiedostoon tallennetun volumetrisen videon toistamiseen (ks. kuvio 7).



Kuvio 7. 3DVideo-tiedoston rakenne

#### 4.3.5 Tallennetun tiedoston toistaminen

Volumetrisen videon toistamista varten rakennettiin oma Unity-lisäosa, jolla videon pystyy liittämään teoriassa mihin tahansa Unity-pelimoottorilla tehtyyn projektiin. Volumetrisen videon toistamista varten on lisäosassa videosoitinkomponentti joka kehitettiin sellaiseksi, että siinä on valmiit metodit videon toistamiseen liittyviä toimintoja varten ja sovelluksen kehittäjä pystyy itse ohjaamaan videota näitä metodeita hyväksikäyttäen haluamallaan tavalla (ks. Kuvio 8). Videosoitin tukee tiedoston lataamista tietokoneen tiedostojärjestelmästä, sekä verkon ylitse URL-osoitteen perusteella.



Kuvio 8. Näkymä videosoitinkomponentista Unity-pelimoottorissa

Volumetrisen videon toistamista varten eri alustoilla kehitettiin komponentista kaksi eri versiota: työpöytäversio sekä WebGL-versio. Tavallisia työpöytäsovelluksena ajettavia projekteja varten tehdyssä versiossa volumetrinen video esitetään pistepilvenä käyttäen näytönohjaimella suoritettavaa shader-ohjelmaa, jossa jokaisen pistepilvessä olevan pisteen kohdalle luodaan neliö, jonka koko, sijainti ja väri määräytyy tiedostossa olevan datan perusteella. Nämä neliöt ovat aina suunnattuna kohti kameraa.

WebGL-sovelluksissa ei pystytty käyttämään työpöytäversiossa käytettyä shader-ohjelmaa, koska havaittiin että WebGL ei tukenut kolmiulotteisten objektien luontia shader-ohjelmassa. Tämä ratkaistiin käyttämällä pistepilven esittämiseen Unity-pelimoottorista löytyvää Shuriken-partikkeliefektijärjestelmää. Näytönohjaimen

sijaan partikkeliefektijärjestelmä suoritetaan prosessorilla, mikä ei ole yhtä tehokas esitystapa kuin shader-ohjelman käyttäminen, mutta se oli optimaalisin tapa mikä keksittiin pistepilvien esittämiseen WebGL-alustalla.

Äänen toistaminen tapahtuu molemmissa versioissa Unity-pelimoottorin omalla Audio Source –komponentilla. Tämä mahdollistaa äänen muokkaamisen projektissa olevaan ympäristöön soveltuvaksi, esimerkiksi lisäämällä äänen kaikua tai säätämään sen, kuinka kauas ääni kuuluu ympäristössä.

#### 4.3.6 Volumetrisen videon toimivuus eri alustoilla

Videoiden toistamista testattiin sekä tavallisissa työpöytäsovelluksissa että WebGL-sovelluksessa. Työpöytäsovellusta testattiin toistamalla volumetrinen videota, jossa pistepilvessä oli noin 50 000 pistettä. Testaus tapahtui Windows 10 käyttöjärjestelmällä projektin käytössä olevilla pelikäyttöön suunnitellulla Razer Blade 15 Base kannettavilla tietokoneilla sekä 2017 vuoden peruskäyttöön suunnitellulla HP:n kannettavalla tietokoneella. Molemmissa tapauksissa videon toistaminen kuormitti tietokonetta niin vähäisesti, että sovelluksen ruudunpäivitysnopeus ei tippunut alle näytön natiivin ruudunpäivitysnopeuden.

Videon toistamista WebGL-sovelluksessa testattiin samoilla kannettavilla tietokoneilla ja samalla videolla, joita käytettiin työpöytäsovelluksenkin testaamiseen. Näiden lisäksi testattiin sovellusta myös Huawei Honor 8 ja Honor 8X älypuhelimilla. Tietokoneilla testatessa käytettiin testaamiseen seuraavia selaimia: Microsoft Edge 87, Mozilla Firefox 83, Google Chrome 86 sekä Opera GX 71. Älypuhelimilla testattiin videon toistamista seuraavilla selaimilla: Mozilla Firefox 83, DuckDuckGo 5.70, Brave 1.17, Opera 60 sekä Google Chrome 86.

Kannettavilla tietokoneilla testatessa videon toistaminen onnistui ongelmitta molemmilla tietokoneilla kaikilla testatuilla selaimilla lukuunottamatta Firefoxia, joka ei suostunut toistamaan yli 267 megatavun kokoisia tiedostoja, koska se oli selaimessa IndexedDB:n suurin sallittu tiedostokoko. Tämän rajoituksen pystyi



kiertämään käyttämällä Firefoxin yksityistä selausta, koska yksityisen selauksen olessa käytössä Firefox ei tallenna verkon yli ladattuja tiedostoja IndexedDB:een.

Älypuhelimilla testatessa oli monilla selaimilla ongelmia videon toistamisessa. Ongelmat vaikuttivat testien perusteella johtuvan siitä, että videoiden toistaminen käyttää liian paljon muistia. Luotettavimmin videot saatiin toimimaan Firefoxilla, mutta video onnistuttiin toistamaan myös DuckDuckGolla. Muilla selaimilla ei testattua videota saatu toimimaan.

Pidemmissä yli minuutin mittaisissa nauhoitteissa huomattiin että ääni ja videokuva eivät enää pysyneet synkronoituna vaan ääni alkoi toistua videokuva hitaammin. Tämä havaittiin sekä työpöytäversiossa, että WebGL-versiossa.

## 5 Tulokset

Tässä kappaleessa käydään läpi kehitystutkimuksen aikana saadut tulokset ja kuinka hyvin ne vastasivat toimeksiantajan asettamia vaatimuksia.

### 5.1 Kuvausjärjestelmän helppokäyttöisyys

Kuvauslaitteiston käytöstä saatiin tehtyä helppoa sillä kuvaamiseen käytetyt kamerat tarvitsee vain asettaa jaluistoilleen ja liittää tietokoneisiin USB-johdoilla. Kuvaaminen kehitystutkimuksen aikana kehitetyllä kuvaussovelluksella taas ei ole ihan niin helppoa, kuin mitä testatuilla valmiilla kuvaussovelluksilla olisi ollut, koska sovelluksesta tehtiin vain proof of concept –prototyypisovellus. Tästä syystä sovelluksen käyttöliittymää ei viimeistelty ja sovellusta joutuu käyttämään Unity-pelimoottorin editorin kautta. Kuvaamista helpottamaan kehitettiin kuitenkin ominaisuus jonka avulla kuvaaminen onnistuu tarvittaessa yksin myös silloin, kun kuvataan käyttäen useampaa tietokonetta, koska eri tietokoneilla olevat sovellukset pystyvät lähettämään käskyjä toisilleen verkon yli. Kameroiden sijainnin ja asennon kalibroinnin tarkkuus ohjelmaan luodun kalibroituvuuskalun avulla ei myöskään ollut

täydellinen, vaan kameroiden sijaintia ja asentoa joutui säätämään manuaalisesti parhaan mahdollisen lopputuloksen aikaansaamiseksi. Kuvauksen yhteydessä on sovelluksessa mahdollista nauhoittaa myös ääntä käyttäen erillistä tietokoneeseen liitettyä mikrofonia.

Kuvatun materiaalin yhdistäminen ja jälkikäsitteily saatiin tehtyä melkko helpoksi, mutta sitä myös joutuu käyttämään Unity-pelimoottorin editorista käsin, koska käyttöliittymää ei viimeistelty. Lopullisen videon ulosvienti sovelluksesta saatiin tehtyä tehokkaaksi, joten myös pidempien volumetristen videoiden ulosvienti onnistuu nopeasti. Lopputuloksena syntyy yksi tiedosto johon on pakattu sekä volumetrinen video sekä ääniraita mikäli ääntä nauhoitettiin kuvauksen yhteydessä.

## 5.2 Toistaminen

Järjestelmällä kuvattujen videoiden toistamista varten kehitettiin Unity-lisäosa, joka mahdollistaa kuvattujen volumetristen videoiden toistamisen missä tahansa Unity-projektissa. Tämä tekee videoiden lisäämisen erilaisiin projekteihin helppoa. Lisäosan lisäksi tehtiin proof-of-concept –videosoitinsovellus, josta tehtiin työpöytäversio sekä verkkosivustolle upotettava WebGL-versio (ks. kuvio 9). Nämä mahdollistavat järjestelmällä kuvattujen volumetristen videoiden toistamisen suoraan sovelluksesta. Etenkin WebGL-versio on kätevä silloin, kun halutaan vain lisätä volumetrinen video verkkosivustolle, mutta projektin aikana tehtyjen testien perusteella todettiin videoiden toimivan huonosti mobiililaitteiden selaimissa.



Kuvio 9. Näkymä videosoitinsovelluksesta

### 5.3 Liikuteltavuus ja hinta

Järjestelmästä pyrittiin tekemään mahdollisimman liikuteltava, joten laitteistovalinnat tehtiin sen perusteella, että kuvausjärjestelmä olisi helppo siirtää paikasta toiseen. Suurin liikuteltavuutta rajoittava tekijä järjestelmässä on se, että kameroiden käyttäminen aktiivisilla USB-jatkojohdoilla vaatii erillisen virtalähteen. Ilman virtalähdettä kamerat eivät toimineet luotettavasti silloin kun ne olivat kytkettynä tietokoneisiin kyseisillä jatkojohdoilla. Tämän takia järjestelmää ei pysty käyttämään luotettavasti sellaisissa kuvausympäristöissä, joissa verkkovirtaa ei ole saatavilla. Teoriassa tämä olisi mahdollista ratkaista käyttämällä esimerkiksi kahdentoista voltin vapaa-ajan akkuja sekä virtainvertteriä verkkovirran sijaan. Aiheesta keskusteltiin toimeksiantajan kanssa, mutta tätä ei kuitenkaan lähdetty testaamaan käytännössä. Tietokoneiden käyttöä niiden omien akkujen varassa testattiin myös, mutta tällä ei huomattu olevan vaikutusta kuvaukseen tai järjestelmän toimivuuteen.

Järjestelmän rakentamisen tarkkaa euromääräistä hintaa ei laskettu, mutta hinta ei kuitenkaan ylittänyt projektille annettua budjettia. Koska kuvausohjelmisto päädyttiin kehittämään itse, ei sen käyttämisestä aiheudu jatkossa myöskään sellaisia kuukausittaisia kuluja kuin mitä olemassa olevia kuvaussovelluksia käyttämällä olisi tullut.

## 6 Johtopäätökset

Tässä luvussa vastataan opinnäytetyön tutkimuskysymyksiin.

**Mitä on volumetrinen videokuvaus?**

Volumetrinen videokuvaukseen on kohteen videokuvaamista laitteistolla, jolla perinteisen väridatan lisäksi saadaan tietoa jokaisen väridatassa olevan pisteen sijainnista kameran suhteutettuna. Tämän informaation avulla pystytään muodostamaan pistepilvi, joka sisältää jokaisen kuvattun pikselin sijainnin kolmiulotteisessa avaruudessa. Jotta kohteesta saadaan kuvattua täydellinen kolmiulotteinen video, niin tarvitaan useampi kamera, jotta kohteesta saadaan kuvattua dataa jokaisesta suunnasta. Useammalla kameralla kuvatessa täytyy myös olla tiedossa kameran sijainti ja asento suhteessa kuvattavaan kohteeseen, jotta eri kameroista tuleva data pystytään yhdistämään yhdeksi isoksi pistepilveksi. Yhdistämisvaiheessa on dataa mahdollista myös jälkikäsitellä esimerkiksi siivoamalla pistepilvestä pois tarpeeton data tai poistamalla pisteet, joiden sijaintiarvot ovat virheellisiä. Kun kuvattu data on saatu yhdistettyä yhdeksi pistepilveksi, voidaan sen perusteella rakentaa pistepilvestä kolmiulotteinen malli jokaisesta kuvattun videon kehyksestä tai vaihtoehtoisesti voidaan kuvattu materiaali toistaa myös pistepilvenä käyttäen esimerkiksi pistepilvien toistamiseen soveltuvia shader-ohjelmia.

### **Mitä laitteita vaaditaan volumetriseen videokuvaukseen soveltuvan kuvausjärjestelmän rakentamiseen?**

Vähimmäisvaatimukset volumetriseen videokuvaukseen ovat kamera, tietokone ja ohjelmisto. Volumetristä videota voidaan teoriassa tallentaa millä tahansa kameralla, jolla pystyy perinteisen väridatan lisäksi tallentamaan kuvatussa väridatassa olevien pikseleiden sijainti kolmiulotteisessa avaruudessa, mutta mikäli ei haluta kehittää omaa kuvaussovellusta niin kameravalintaan vaikuttaa se, mitä kameroita järjestelmän käyttöön valittu kuvaussovellus tukee. Usealla kameralla kuvatessa täytyy myös ottaa huomioon kameroiden sekä kuvausohjelmistojen laitteistovaatimukset.

### **Miten volumetristä videota kuvataan?**

Volumetrisen videon kuvaamiseen tarvitaan laitteiston lisäksi kuvaukseen soveltuva ohjelmisto. Kirjoitushetkellä kuvaussovelluksia on markkinoilla vielä hyvin vähän, mutta lupaavimmilta vaikuttavat EF EVE:n Volumetric Capture ja Scatterin Depthkit. Nämä sovellukset tukevat kirjoitushetkellä Intelin RealSense D415 ja D435 kameroita sekä Microsoftin Azure Kinect kameroita.

Ennen kohteen kuvaamista täytyy määritellä kameroiden sijainti ja asento suhteessa kuvattavaan kohteeseen. Kun tämä on määritelty, niin voidaan aloittaa kohteen kuvaaminen. Kun kohde on saatu kuvattua, täytyy eri kameroilla kuvattu data yhdistää yhdeksi kokonaisuudeksi. Tämän yhteydessä voidaan kuvatusta materiaalista siivota pois tarpeeton tai virheellinen data. Materiaalille voidaan käytetystä ohjelmistosta riippuen suorittaa myös muita jälkikäsittelytoimia.

Materiaalin yhdistämisen jälkeen yhdistetty ja käsitelty materiaali viedään ohjelmasta ulos joko videomuodossa, pistepilvenä tai sarjana kolmiulotteisia malleja. Eri ohjelmistot mahdollistavat materiaalin ulosviennin eri tiedostomuodoissa, joten käytävissä olevat tiedostomuodot vaihtelevat käytetyn ohjelmiston mukaan.

### **Miten volumetristä videota toistetaan?**

Volumetriselle videolle ei ole vielä kirjoitushetkellä olemassa olevia standardoituja tiedostomuotoja tai videokoodekkeja, joten järjestelmällä kuvatun materiaalin toistaminen on riippuvainen käytetyn ohjelmiston tarjoamista toistomahdollisuuksista. MPEG-työryhmällä on kuitenkin kehitteillä MPEG-I standardi, joka pitää sisällään muun muassa määritelmät volumetrisen videon tiedostomuodolle, pakkaamiselle ja toistamiselle. Jos kuvausjärjestelmään valittu ohjelmisto ei tarjoa halutunlaista toistamistapaa, niin on sellainen tehtävä itse.

## **7 Pohdinta**

### **7.1 Tulokset ja jatkokehitys**

Tämän opinnäytetyön tavoitteena oli luoda demo volumetriseen videokuvaukseen soveltuvasta kuvausjärjestelmästä. Järjestelmässä oli alun perin tarkoitus käyttää olemassa olevia kuvausohjelmistoja, mutta kehitystyön aikana havaittiin, että olemassa olevilla kuvaussovelluksilla ei saatu halutunlaista tulosta aikaiseksi, joten päätettiin järjestelmää varten kehittää proof of concept –prototyyppi omasta kuvaussovelluksesta.

Kehitystyö aloitettiin tutustumalla volumetrisen videokuvauksen teoriaan ja aiheesta tehtyihin tutkimuksiin. Tämän tuloksena syntyi opinnäytetyön teoriaosuus ja siitä saatiin luotua myös tietopohja, jota hyödynnettiin itse kuvausjärjestelmän kehitystyössä. Aiheesta ei löydetty juurikaan aikaisempaa suomenkielistä tutkimusta ja tämä opinnäytetyö on todennäköisesti yksi ensimmäisistä volumetrisesta videokuvauksesta tehdyistä suomalaisista tutkimuksista.

Kuvausjärjestelmää varten hankittu laitteisto soveltui kehitettävän kuvausjärjestelmän tarpeisiin hyvin. Ainoastaan kameroiden tuottaman syvyysdatan laatu osoittautui oletettua huonommaksi. Syvyysdatan laatu heikkeni huomattavasti kuvattaessa kohteita, jotka olivat yli kahden metrin etäisyydellä kamerasta.

Olemassa olevat kuvaussovellukset osoittautuivat myös tutkimuksen aikana tehdyissä testeissä oletettua huonommiksi ja keskeneräisimmiksi, mutta etenkin Scatterin kehittämä Depthkit-sovellukseen kehitteillä oleva useamman kameran tuki vaikutti lupaavalta. Tästä syystä päädyttiin kehittämään demo omasta kuvaussovelluksesta. Kuvaussovelluksen kehittäminen veikin suurimman osan opinnäytetyön tekemiseen käytetystä ajasta. Koska ohjelmisto kehitettiin itse, niin pystyttiin varmistamaan, että se vastaa toimeksiantajan tarpeita mahdollisimman hyvin eikä sen käytöstä myöskään koidu ylimääräisiä kuukausittaisia kuluja. Ohjelmiston kehitys osoittautui kuitenkin oletettua vaikeammaksi, mutta siitä saatiin tuotettua kuitenkin toimiva demoversio, joka piti sisällään perustoiminnallisuudet järjestelmällä kuvatun volumetrisen videon kuvaamiseen, editoimiseen ja toistamiseen.

Vaikka kehitetyllä ohjelmistolla pystytäänkin kuvaamaan volumetrisia videoita, niin oli ohjelmistossa vielä paljon asioita, jotka vaatisivat jatkokehittämistä. Muun muassa kameroiden sijainnin ja asennon automaattisen määrittelyn tarkkuutta pitäisi vielä parantaa, ettei kameroiden sijaintia ja asentoa tarvitsisi säätää käsin.

Kameroiden käyttäminen suuremmilla resoluutioilla aiheutti myös epävakautta kameroiden kuvataajuudessa. Tämä olisi ehkä ollut mahdollista ratkaista käyttämällä erillistä synkronointikaapelia kameroiden kuvataajuuden synkronoimiseksi, mutta

tätä ei kuitenkaan testattu koska järjestelmässä käytetyille Intel RealSense D415 –kameroille ei ole saatavilla valmista synkronointikaapelia vaan sellainen olisi pitänyt rakentaa itse. Tutkimisen arvoinen jatkokehitysaihe olisi myös järjestelmässä käytettävien kameroiden vaihtaminen Microsoftin Azure Kinect –kameroihin. Niissä kameroiden välinen synkronointi on toteutettu käyttäen tavallista 3.5 mm audiokaapelia ja ainakin internetissä löytyvien vertailuiden perusteella niissä on myös parempi syvyysdatan laatu järjestelmässä käytettyihin kameroihin verrattuna.

Pidempiä yli minuutin mittaisia videoita kuvatessa huomattiin äänen alkavan toistua videota hitaammin videon loppua kohden. Tämän ratkaiseminen tarvitsisi myös jatkokehitystä. Kuvatussa materiaalissa ilmeni myös paljon mittavirheitä syvyysinformaatioissa ja vaikka materiaalin siivoamiseen kehitettiin työkaluja, niin paremman lopputuloksen saamiseksi näitä olisi hyvä myös kehittää eteenpäin.

Kuvausjärjestelmän liikuteltavuutta olisi myös mahdollistaa parantaa kehittämällä akkukäyttöinen virtalähde järjestelmän verkkovirtaa tarvitseville komponenteille. Tämä mahdollistaisi kuvaamisen myös paikoissa, joissa verkkovirtaa ei ole saatavilla.

## 7.2 Tulosten luotettavuus

Teoreettisen viitekehyksen luotettavuuteen vaikuttaa pääasiassa aihe-alueesta tehtyjen tutkimusten vähäinen määrä sekä aihe-alueen uutuus. Teoreettista viitekehystä kirjoittaessa pyrittiin käyttämään kuitenkin pääasiassa tieteellisiä lähteitä niiltä osin kun se oli mahdollista ja lähteinä pyrittiin käyttämään mahdollisimman uusia lähteitä ja teoriaosuutta voidaan pitää pääosin luotettavana.

Itse järjestelmän kehittämistutkimus pyrittiin tekemään mahdollisimman luotettavaksi, mutta koska tuloksena syntyi valmiin tuotteen sijaan proof of concept –tyylinen demo, niin tulosten luotettavuus ei ole välttämättä kaikilta osin paras mahdollinen. Isoin tulosten luotettavuuteen vaikuttava tekijä oli kehitetyn järjestelmän verrattaen vähäinen testaus.

Videoiden toistamista olisi voinut testata useammilla laitteilla, selaimilla ja käyttöjärjestelmillä, jotta olisi saatu parempi selvyys niiden toimivuudesta eri alustoilla. Tätä ei kuitenkaan tehty, koska videoiden toistosovellukset olivat vielä kokeellisia ja mikäli niitä lähdetään jatkokehittämään, niin videotiedostot ja toistosovellukset tulevat todennäköisesti vielä muuttumaan. Tästä syystä päätettiin sovellusta testata vain projektissa saatavilla olevilla laitteilla ja ohjelmilla.

Järjestelmän käytettävyyttä olisi myös ollut hyvä testauttaa projektin ulkopuolisilla ihmisillä, että olisi saatu parempaa tietoa järjestelmän käytettävyydestä. Tätä ei kuitenkaan lähdetty tekemään, koska kuvausjärjestelmä oli vielä hyvin keskeneräinen ja kuvausohjelman käyttöliittymässä ei esimerkiksi ollut kaikkia toimintoja vielä, joten ohjelmaa joutui käyttämään Unityn editorin kautta.



## Lähteet

Alexiadis D., Chatzitofis, A., Daras, P., Louizis, G., Zarpalas, D., Zioulis, N. & Zoidi, O. 2017. An Integrated Platform for Live 3D Human Reconstruction and Motion Capturing. IEEE Transactions on Circuits and Systems for Video Technology, 27, 4, 798–813. Viitattu 7.12.2020. <https://janet.finna.fi>, IEEE Xplore Digital Library.

Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C. & Taubin, G. 1999. The Ball-Pivoting Algorithm for Surface Reconstruction. IEE Transactions on Visualization and Computer Graphics, 5, 4, 349–359. Viitattu 7.12.2020. <https://janet.finna.fi>, IEEE Xplore Digital Library.

Bister, T. 2019. Tietojenkäsittelyn opinnäytetyö: viittoja ja karttoja tutkimisen ja kehittämisen teille. Jyväskylä: Jyväskylän ammattikorkeakoulu. Jyväskylän ammattikorkeakoulun julkaisuja 272.

Bolitho, M., Hoppe, H. & Kazhdan, M. 2006. Poisson Surface Reconstruction. Viitattu 7.12.2020. [https://people.engr.tamu.edu/schaefer/teaching/689\\_Fall2006/poisson-recon.pdf](https://people.engr.tamu.edu/schaefer/teaching/689_Fall2006/poisson-recon.pdf).

Boyce, J., Peng, W.-H., Stockhammer, T. & Wien, M. 2019. Standardization Status of Immersive Video Coding. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 9, 1, 5–17. Viitattu 7.12.2020. <https://janet.finna.fi>, IEEE Xplore Digital Library.

Daras, P., Doumanoglou, A., Karakottas, A., Papachristou, A., Sterzentsenko, V., Zarpalas, D. & Zioulis, N. 2019. A low-cost, flexible and portable volumetric capturing system. Viitattu 7.12.2020. <https://arxiv.org/pdf/1909.01207.pdf>.

Daras, P., Papachristou, A., Zarpalas, D. & Zioulis, N. 2018. Markerless Structure-based Multi-sensor Calibration for Free Viewpoint Video Capture. Viitattu 7.12.2020. <https://www.iti.gr/iti/files/document/publications/S05-Markerless%20Structure-based%20Calibration.pdf>.

Depthkit.js. 2018. Depthkit.js:n GitHub repositorio. Viitattu 7.12.2020. <https://github.com/ScatterCo/Depthkit.js>.

Dorodnicov, S., Grunnet-Jepsen, A., Khuong, T., Mulla, O., Sweetser, J. & Tong D. 2020. Intel® RealSense™ Self-Calibration for D400 Series Depth Cameras. Intel Corporation. Viitattu 7.12.2020. <https://dev.intelrealsense.com/docs/self-calibration-for-depth-cameras>.

Export Formats. 2019. Artikkelin Depthkitin dokumentaationsivustolta. Viitattu 7.12.2020. <https://docs.depthkit.tv/docs/depthkit-early-access>.

Girskaitė, L. 2020. Equipment list. Artikkelin EF EVE:n tukisivustolla. Viitattu 7.12.2020. <https://ef-eve.com/help/knowledge-base/equipment-list/>.

- Girskaitė, L. 2020. Volumetric Video Exporting. Artikkelin EF EVE:n tukisivustolla. Viitattu 7.12.2020. <https://ef-eve.com/help/knowledge-base/step-3-volumetric-video-exporting/>.
- Gopalakrishnan, V., Han, B., Pair, J. & Qian, F. 2019. Toward Practical Volumetric Video Streaming on Commodity Smartphones. Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications, 135–140. Viitattu 7.12.2020. [https://www-users.cs.umn.edu/~fengqian/paper/vvs\\_hotmobile19.pdf](https://www-users.cs.umn.edu/~fengqian/paper/vvs_hotmobile19.pdf).
- Grzegorzek, C., Koch, R., Kolb, A. & Theobalt, C. 2013. Time-of-Flight and Depth Imaging: Sensors, Algorithms, and Applications. Berliini: Springer.
- Hsieh, F., Leang, J., Lin, K., Schipf, D., Tsui, C. & Wang, W. 2014. Using a Time of Flight Method for Underwater 3-Dimensional Depth Measurements and Point Cloud Imaging. Viitattu 7.12.2020. <https://janet.finna.fi>, IEEE Xplore Digital Library.
- Kolb, A., Lefloch, D. & Sarbolandi, H. 2015. Kinect Range Sensing: Structured-Light versus Time-of-Flight Kinect. Institute for Vision and Graphics, University of Siegen, Germany. Viitattu 7.12.2020. <https://arxiv.org/pdf/1505.05459.pdf>.
- Kolb, A., & Lindner, M. 2007. Calibration of the intensity-related distance error of the PMD TOF-camera. Institute for Vision and Graphics, University of Siegen, Germany. Viitattu 2020. [https://www.researchgate.net/publication/241478403\\_Calibration\\_of\\_the\\_intensity-related\\_distance\\_error\\_of\\_the\\_PMD\\_TOF-camera](https://www.researchgate.net/publication/241478403_Calibration_of_the_intensity-related_distance_error_of_the_PMD_TOF-camera).
- Porter, A., Porter T. 2018. Photogrammetry and Volumetric Capture. Intel Corporation. Viitattu 7.12.2020. <https://software.intel.com/sites/default/files/managed/94/3d/photogrammetry-and-volumetric-capture.pdf>.
- The first Volumetric Capture Studio in Finland to be built in CIVIT. 2020. Uutisartikkeli CIVIT:n verkkosivuilla. Viitattu 13.3.2020. <https://civit.fi/the-first-volumetric-video-capture-studio-in-finland/>.
- Volumetric Video Market - Global Forecast to 2025. 2020. Volumetrin videoiden markkinatutkimuksen yhteenveto. Viitattu 7.12.2020. <https://www.marketsandmarkets.com/Market-Reports/volumetric-video-market-259585041.html>.