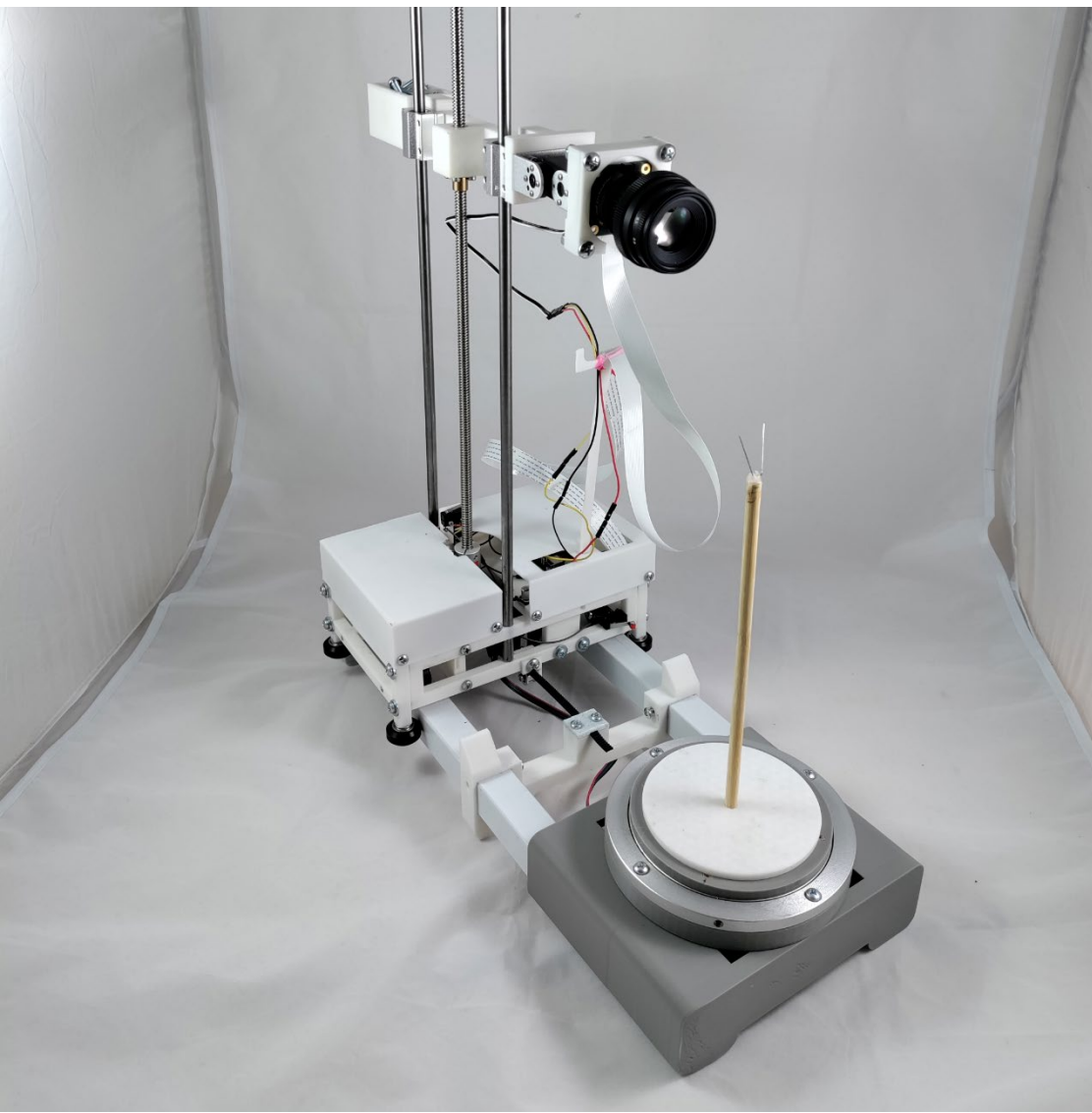


Joni Mustakangas

Fotogrammetriastudion automatisointi



Tradenomi (AMK)
Tietojenkäsittelyn koulutus
Syksy 2021



KAMK • University
of Applied Sciences

Tiivistelmä

Tekijä: Mustakangas Joni

Työn nimi: Fotogrammetriastudion automatisointi

Tutkintonimike: Tradenomi, tietojenkäsittely, peliala

Asiasanat: Fotogrammetria, Fotogrammetriastudio, pelituotanto, automaatio

Fotogrammetria on tieteenala, jossa mitataan etäisyyksiä tarkastelemalla valokuvia. Moderneja tietokoneohjelmia käyttäen valokuvista saadaan luotua rekonstruktio kuvatusta tilasta tai kohteesta, jota voidaan käyttää useisiin eri tarkoituksiin, kuten kartoittamiseen, maanmittaukseen, lääketieteeseen, historian tallioimiseen ja viihdeteollisuuteen, kuten videopelit.

Tämän opinnäytetyön tarkoituksena oli tutustua fotogrammetriaan ja toteuttaa mahdollisimman pitkälle toteutettu automatisoitu fotogrammetriastudio. Luotettavasti toimiva automaatio fotogrammetriastudiossa vähentäisi manuaalista työtä ja lisäisi työskentelytehokkuutta. Monissa tapauksissa automaatiolla pystytään täysin korvaamaan manuaalisesti tehtävä 3D-mallinnustyö varsinkin tapauksissa, missä tavoitteena on toteuttaa realistista grafiikkaa.

Opinnäyte sisältää pinnallisesti monia eri aihealueita, kuten CAD-suunnittelua, automaatiota, korttitietokoneita, 3D-grafiikkaa ja tietenkin fotogrammetriaa. Opinnäytetyössä käydään läpi, mitä fotogrammetria on, kuinka fotogrammetria eroaa muista 3D-skannaustekniikoista ja miten fotogrammetriatekniikoita voidaan hyödyntää pelituotannossa.

Suunnittelun pohjana mallia katsottiin kuluttajille saatavilla olevista 3D-tulostimista ja toteutus rakennettiin Raspberry Pi - korttitietokoneen ympärille. Suunnittelutyössä käytettiin FreeCAD-ohjelmistoa ja osat suunniteltiin siten, että ne voitiin tulostaa Creality Ender V2 3D – tulostinta käyttäen.

Lopputuotoksena toteutettiin puoliautomatisoitu fotogrammetriastudio, jota pystytään käyttämään esineiden digitalisoinnissa ja hyödyntämään niitä pelituotannossa. Fotogrammetriastudio tuottaa laadukasta lähdemateriaalia jälleenrakennusohjelmille ja lopputuloksena saatiin useita laadukkaita 3D-malleja, jotka laadullisesti riittäisivät myös mainostuotantoon.

Abstract

Author: Mustakangas Joni

Title of the Publication: Photogrammetry Studio Automation

Degree Title: Bachelor of Business Administration, Business Information Technology

Keywords: photogrammetry, photogrammetry studio, game production, automation

Photogrammetry is a discipline in which distances are measured by looking at photographs. Using modern computer software these images can be used to create a digital reconstruction of the photographed object or scene. Nowadays photogrammetry has been used in mapping, surveying, medicine, history, and in the entertainment industry such as video games.

The purpose of this thesis was to get acquainted with photogrammetry and to implement an automated photogrammetry studio as ready to use as possible. A reliable automated photogrammetry studio would reduce manual work and increase work efficiency. In many cases, automation could completely replace manual 3D-modeling work, especially in cases where the goal is to implement realistic graphics.

The thesis superficially covers many different subject areas, such as CAD design, automation, single-board computers, 3D graphics, and photogrammetry. The thesis goes through what photogrammetry is, how photogrammetry differs from other 3D scanning techniques, and how photogrammetry techniques can be utilized in game production.

As a design basis, consumer 3D printers were viewed, and the implementation was built around the Raspberry Pi single-board computer. FreeCAD software was used for the design work, and the parts were designed so that they could be printed using a Creality Enver V2 3D printer.

The end product was a semi-automated photogrammetry studio that can be used to digitize objects and utilize them in-game production. The photogrammetry studio produces high-quality source material for reconstruction programs, and the result was several high-quality 3D models that would also be of sufficient quality for advertising production.

Sisällysluettelo

1	Johdanto	1
2	3D-Skannaus	2
2.1	Aikaeromenetelmä	2
2.2	Strukturoitu valomenetelmä	3
2.3	Stereoskooppinen kuvaus	4
3	Fotogrammetria	5
3.1	Fotogrammetrian soveltaminen pelialalla	5
3.2	Ohjelmistot	7
3.3	Fotogrammetriavalokuvaaminen	8
4	Fotogrammetriastudio	11
5	Fotogrammetriastudion suunnittelu	12
5.1	Studion ja automaation suunnittelu	14
5.2	Kehitysalustan valitseminen	16
5.3	Kameran valitseminen	17
5.4	Raspberry Pi - kameran kytkeminen	18
5.5	Askelmoottorin hallinta	21
5.6	Servomoottorin hallinta	24
5.7	Prototyypin rakentaminen	28
6	Fotogrammetriastudion toteutus	30
6.1	CAD-suunnittelu	31
6.2	Komponenttien kytkentä	35
6.3	3D-Tulostaminen	36
6.4	Automaation ohjelmointi	37
7	Tulokset	40
8	Pohdinta	45
9	Yhteenveto	46
10	Lähteet	47

Käsiteluettelo

3D-skannaaminen	Prosessi, jossa digitalisoidaan esine tai asia digitaalisesti esitettävään formaattiin.
Digitalisointi	Prosessi, jossa fyysinen materiaali, esine tai tieto digitalisoidaan siihen tarkoitettua välineistöä käyttäen.
Fotogrammetria	Tieteenala, jossa mitataan etäisyyksiä vertailemalla valokuvia.
Fotogrammetriastudio	Tila, jossa pystytään ottamaan valokuvia fotogrammetria käyttöä varten.
Tekstuuri	Valokuva, jota käytetään antamaan pintakuviointi / väri 3D-malleille.
Käänteinen suunnittelu	Suunnittelua, jossa lähtökohtana käytetään valmiista tuotetta.
Pistepilvi	Pistepilvellä tarkoitetaan kolmiulotteista dataa. Jokaisella pisteellä pistepilvessä on x, y ja z sijaintiarvo. Joissakin tiedostomuodoissa piste saattaa sisältää myös väriarvon.
GPIO	General Purpose Input / Output. Tarkoitetaan sellaisia pinnejä korttitietokoneessa tai mikrokontrollerissa, joita voidaan hallita ohjelmallisesti.
ToF	Time of Flight, 3D-skannaustekniikka, jossa mitataan valon lentoaikaa.
STL	STL-tiedosto on yleinen 3D-tiedostomuoto, joka sisältää polygonimallidataa.
RAW	RAW-tiedosto on yksi monista raakakuvan muodoista, eli häviötön tiedostomuoto

1 Johdanto

Tämän opinnäytetyön tavoitteena oli luoda automatisoitu fotogrammetriastudio ja käsitellä sen tuottama data lopputuotteeksi pelimoottoriin. Aiheesta kiinnostuneille tämä opinnäytetyö kertoo fotogrammetriateknologiasta ja sen käytöstä pelituotannossa. Tarkoitukseni oli oppia opinnäytetyön aikana lisää fotogrammetriasta, automaatiosta, datankäsittelystä sekä haastaa omaa osaamispohjaani monipuolisesti.

Kiinnostus fotogrammetriaan heräsi kesällä 2015, kun tutustuin Microsoft Kinect - laitteeseen. Microsoft Kinect - laitteet pystyvät kaappaamaan pistepilven näkemästään ympäristöstä. Tätä pistepilveä pystyy hyödyntämään pelinkehityksessä tilojen sekä esineiden visualisoinnissa.

Microsoftin tarjoamat ohjelmistot olivat silloin vaikeita käyttää, eikä oma taitotasoni silloin riittänyt kehittämään niitä pidemmälle. Kinect-laitteiden käytön myötä minulle kumminkin syntyi kiinnostus kaapata sekä digitalisoida esineitä. Minusta oli hyvin kiehtovaa nähdä, kuinka todellinen maailma kääntyi digitaaliseksi ja kuinka paljon erilaisia käyttötarkoituksia on digitalisoiduille tuotteille.

Tulevaisuudessa fotogrammetrialla tulee olemaan entistä merkittävämpi rooli elokuvien ja pelien tuotannossa ja se näkyy tällä hetkellä fotogrammetrian käytön yleistymisenä pelialalla. Ongelmatonta sen käyttäminen ei kumminkaan ole. Tekniikkana se vaatii laitehankintoja, osaavia tekijöitä, sekä paljon laskentatehoa. [1.]

Opinnäytetyössä tutustuttiin fotogrammetrian puutteisiin sekä ongelmakohtiin ja tarkoituksena oli löytää helppo sekä luotettava tapa digitalisoida esineitä käyttäen halpoja ja helposti saatavilla olevia komponentteja.

2 3D-Skannaus

3D-skannaaminen on teknologia ja prosessi, joka mahdollistaa tilojen ja esineiden digitalisoinnin 3D-formaatiin. 3D-Skannausteknologioita on kehitetty useita eri käyttötarkoituksia varten. Perinteisiä käyttötarkoituksia ovat olleet kartoittaminen, maanmittaus, lääketiede ja tuotekehitys. [2.]

3D-Skannerit voidaan jakaa kahteen pääryhmään koskettaviin ja ei-koskettaviin. Koskettavat 3D-skannerit keräävät tietoa koskettamalla digitalisoitavaa kohdetta. Ei-koskettavat 3D-skannerit pystytään jaottelemaan aktiivisiin sekä passiivisiin skannereihin. [3.]

Passiivi- ja aktiiviskannerit eroavat toisistaan siten, että passiiviset skannerit luottavat ympäristön valoon skannaustilanteessa ja eivät itse säteile valoa. Tyypillisiä aktiiviskannaus menetelmiä ovat aikaeromenetelmä, laserkolmiomittaus ja strukturoitu valo. Tyypillisiä passiiviskannaus menetelmiä ovat stereoskooppinen kuvaus, fotogrammetria ja siluettikuvaaminen. [4.]

Fotogrammetria tyypillisesti luokitellaan passiiviseksi skannausmenetelmäksi, mutta mikäli kuvaus tapahtuu studiossa, voidaan se tällöin luokitella aktiiviseksi 3D-skannaamiseksi. Automatisoitu fotogrammetriastudio on siten aktiivinen 3D-skannausmenetelmä, koska studiovalaistuksen tarkoituksena on tuoda esiin kuvattavan kohteen luonnollinen väri ja valaista kuvattava kohde tasaisesti.

2.1 Aikaeromenetelmä

Aikaeromenetelmällä eli ToF mitataan etäisyyksiä hyödyntäen valon lentoaikaa. Tyypillisesti laite koostuu laserista ja sensorista. Aikaeromenetelmälaitteilla saadaan talletettua pistepilvi liikuttamalla skanneria skannattavan kohteen ympäri. [5, s. 3.]

Nykyisin käytetään myös kameroita, jotka hyödyntävät aikaeromenetelmää. Nämä kamerat ovat nostaneet suosiotaan, koska ne kykenevät taltioimaan useita etäisyyksiä kameran näköalueelta. Näiden kameroiden suosio on kasvanut, myös kuluttajalaitteiden keskuudessa. Tunnettu kuluttajalaitte, joka käyttää aikaeromenetelmää on Microsoft Kinect - liiketunnistin. [6.]

ToF menetelmästä on olemassa useita variaatioita, kuten d-ToF(Direct time of Flight) ja i-ToF (indirect Time of Flight). Azure Kinect DK- syvyyskamera käyttää tekniikkaa Amplitude Modulate Continuous Wave (AMCW) Time-of-Flight(ToF). Kamera valaisee moduloitua valoa lähi-infrapuna-

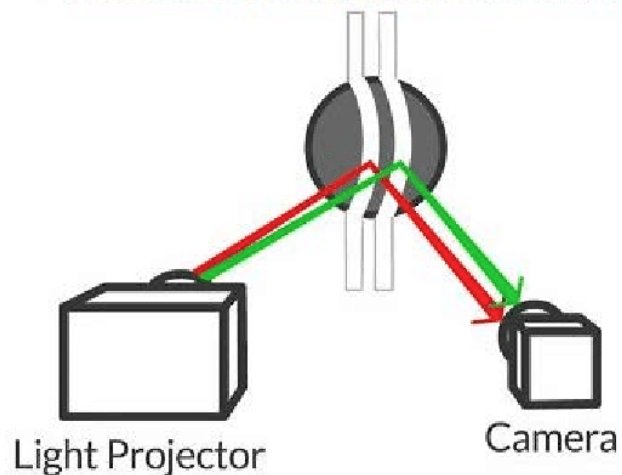
spektrissä tilaan ja laskee tästä, kuinka pitkään valolla kestää kulkea kamerasta tilaan ja takaisin. [6.]

2.2 Strukturoitu valomenetelmä

Strukturoidussa valomenetelmässä digitalisoitavaan kohteeseen projektori heijastaa kuvan, jossa on ennalta tiedossa oleva kuvio. Tyypillisesti kuviossa on mustavalkoisia raitoja. Kamera tarkastelee kuvion vääristymiä ja tämän perusteella pääättelee kuvattavan kohteen muodon. [7, s. 4]

Usein digitalisoitava kohde asetetaan moottorisoidulle alustalle, jotta kuvattava kohde saadaan taltioitua kauttaaltaan. Mikäli kuvattavan kohteen pyörittäminen ei ole mahdollista, voidaan myös skanneria liikuttaa. Strukturoitu valomenetelmä on suosittu käänteisessä suunnittelussa ja historiallisten esineiden digitalisoinnissa. Kuvassa 1. havainnollistetaan strukturoidun valomenetelmän toimintaperiaatetta. [8.]

The object distorts the light lines. This light distortion, or curvature, is captured by the camera and used to calculate object depth, structure and detail.

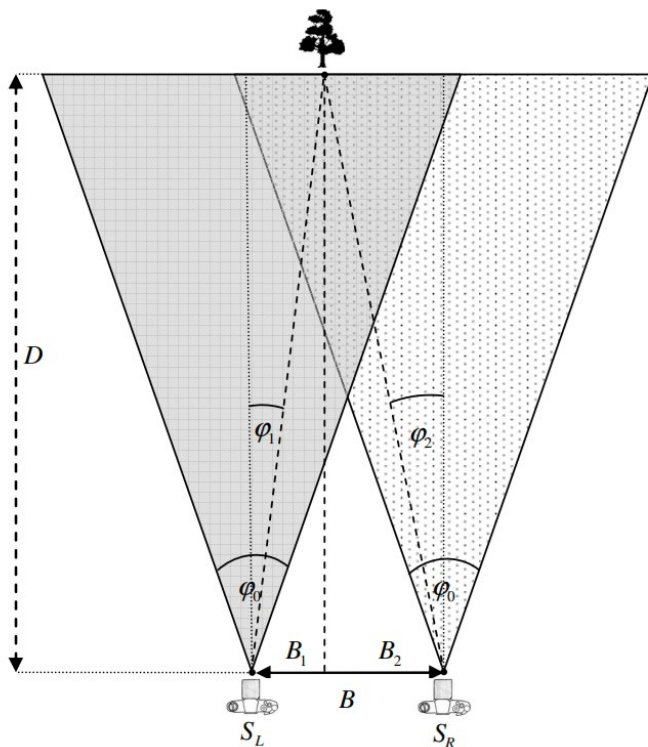


Kuva 1. Kuvassa strukturoidun valomenetelmän toimintaperiaate. [7.]

2.3 Stereoskooppinen kuvaus

Stereoskooppinen kuvaaminen on yksinkertainen ymmärtää, sillä tekniikka toimii samalla periaatteella, miten ihmisen syvyysnäkökin toimii. Stereoskooppisessa kuvaamisessa käytetään kahta kameraa, joiden etäisyys on tiedossa (kuva 2). Kameran ottavat yhtäaikaaisesti samasta kohteesta kuvan, koska molempien kameroiden sijainti ja optiikka on tiedossa, pystytään kuvia vertailemalla laskemaan jokaisen yhtenevän pisteen etäisyys.

Tekniikka muistuttaa fotogrammetriaa siten, että molemmissa käytetään etäisyyksien mittaamisen tietolähteenä valokuvia. Fotogrammetria poikkeaa stereoskooppisesta kuvaamisesta siten, että kameroiden välinen etäisyys on lähtötilanteessa tuntematon. Stereoskooppista valokuvaaamista käytetään stereoskooppisten kuvien sekä videoiden luontiin ja se on monille tuttu teknologia 3D-elokuvista. [9.]



Kuva 2. Osoittaa kuinka etäisyys lasketaan käyttäen stereoskooppista menetelmää. [9.]

3 Fotogrammetria

Fotogrammetria on tiedettä, jossa mitataan etäisyyksiä hyödyntäen tietolähteenä valokuvia tai videota. Valokuvattaessa valokuvaan taltioidaan 3D-tila projisoimalla se tasolle ja tällöin informaatio syvyydestä häviää.

Fotogrammetrian tarkoituksena on kääntää tämä prosessi ja palauttaa menetetty syvyysinformaatio. Fotogrammetria prosessissa otetaan kuvattavasta kohteesta valokuvia tai videota ja ne syötetään 3D-jälleenrakennusohjelmistolle. Jälleenrakennusohjelmisto yrittää etsiä kuvista yhtenevät pisteet ja selvittää pisteiden väliset etäisyydet. Tiheän pistepilven rakentaminen kohtauksesta saadaan linkittämällä kaksi konenäköön perustuvaa tekniikkaa ”Structure-from-motion” (SfM) ja ”Multi View Stereo” (MVS). [10.]

3.1 Fotogrammetrian soveltaminen pelialalla

Fotogrammetriaa käytetään pelialalla 3D-mallien tuottamiseen. Digitalisoinnin kohteena ovat muun muassa eläimet, ihmiset, esineet, tilat sekä luonto. Fotogrammetriaa hyödynnetään pelisovelluksien kehityksessä, koska sillä voidaan potentiaalisesti säästää tuotantokustannuksissa, verrattuna perinteisiin työskentelymenetelmiin. Fotogrammetrialla voidaan myös toistettavasti tuottaa materiaalia tasaisella laadulla.

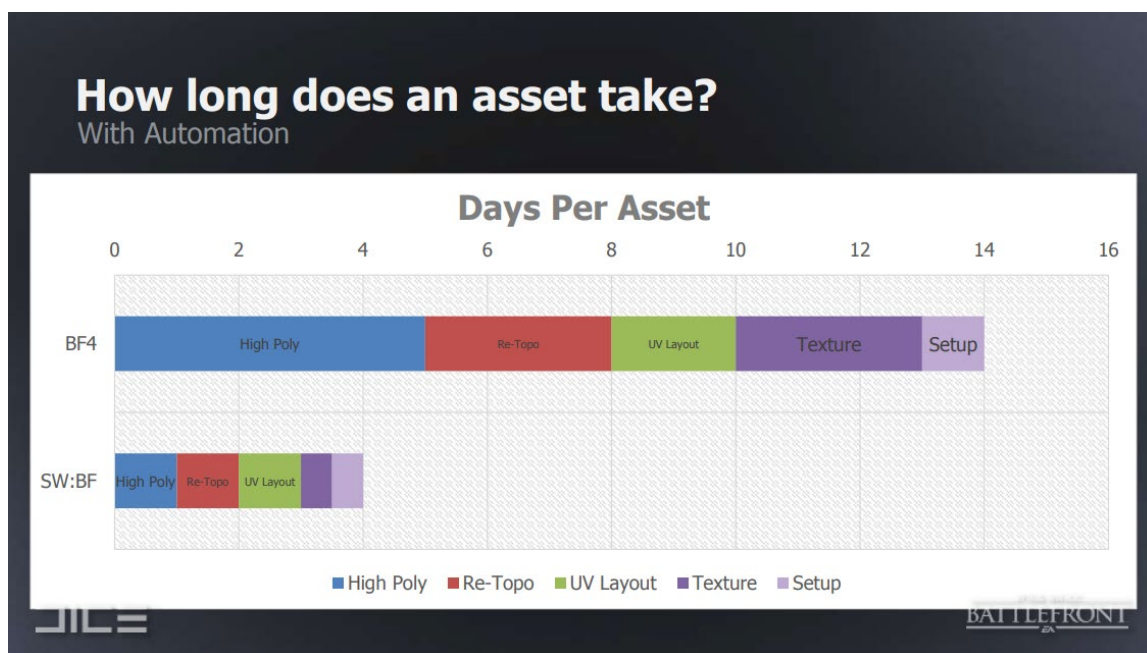
Fotogrammetriaohjelmistot ovat ottaneet viimeisen parin vuosikymmenen aikana merkittäviä kehitysharppauksia. Täten monet pelistudiot ovat lähteneet tutkimaan fotogrammetrian hyötyjä ja mahdollisuuksia.

Unity Technologies on tunnettu pelimoottorikehittäjä. Heidän julkaisemassaan teoksessa ”Photogrammetry Workflow” Fotogrammetrian työnkulku -oppaassa he kertovat, kuinka heidän pelimoottorilleen voidaan tuoda todellisuudesta kaapattuja 3D-malleja. Julkaisussa kerrotaan, kuinka fotogrammetrialla voi perinteiseen mallinnustyöskentelyyn verrattuna säästää kuusi päivää työaika. Tämä on nähtävissä taulukossa 1. [11.]

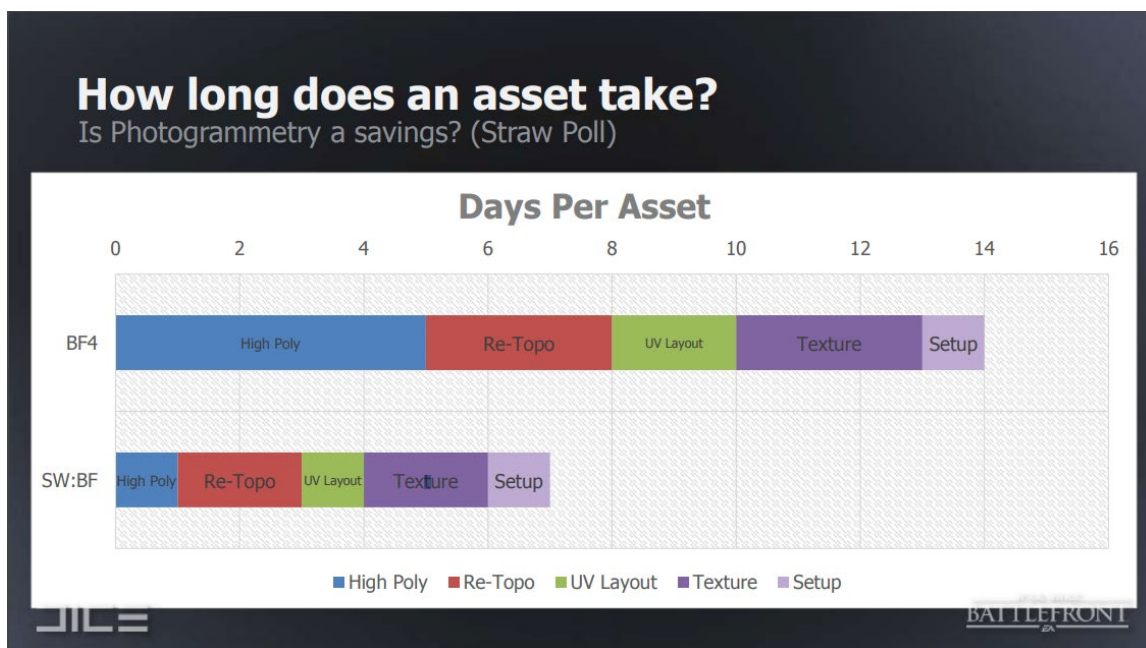
Days	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Classic workflow	High Mesh					Texturing			Retopology	UV + baking	Material	Import IG	LOD	
Photogrammetry workflow	Photos	HM + T	Retopology	UV + baking	Material+ Delighting	Import IG	LOD	Time saved						

Taulukko 1 . Fotogrammetria säästää Unity Technologiesin julkaisun mukaan kuusi päivää työaikaa. [11.]

Vastaavia ajansäästöjä esittelivät myös Hamilton Andrew ja Brown Ken heidän GDC:n konferenssi-puheessaan. He esittelivät peliteknologiakehittäjän DICE:n sisäistä tutkimusta fotogrammetrian käytöstä pelienkehityksessä ja vertasivat sitä perinteisiin menetelmiin. Tutkimustulokset osoittivat selkeitä ajansäästöjä (kuva 3). Heidän mukaansa fotogrammetrian hyödyntämisessä ja automaatioprosessissa on kuitenkin vielä paljon kehitettävää. Kuvassa 4 näkyy DICE:n työntekijöiden olettamus mahdollisista ajansäästöistä. [12.]



Kuva 3. DICE:n tutkimuksen lopputulos osoittaa kymmenen työpäivän ajansäästöä. [12.]



Kuva 4. DICE:n työntekijöiden arvio siitä, kuinka paljon fotogrammetria säästää työaika. [12.]

3.2 Ohjelmistot

Viimeisen reilun kymmenen vuoden aikana kuluttajille ja kehittäjille on tullut tarjolle paljon erilaisia ohjelmistoja, joilla voidaan tuottaa ja käsitellä fotogrammetria tekniikalla tuotettua dataa sekä malleja. Taulukkoon 2 on koottu oman kokemuksen ja Unity Technologiesin fotogrammetriatyöskentelyohjeen perusteella lista hyödyllisistä ohjelmista, joihin kannattaa tutustua. [11.]

Ohjelman Nimi	Kuvaus
Reality Capture	Fotogrammetrian käsittelyohjelma
Meshroom	Fotogrammetrian käsittelyohjelma
Metashape	Fotogrammetrian käsittelyohjelma
Xnormal	Normaalitekstuurien luomistyökalu
Blender	3D-mallinnus työkalu
Knald	Tekstuurien luomistyökalu

Substance designer	Tekstuurien luomistyökalu
VLC media player	Videontoisto-ohjelma, jolla voidaan muuntaa video yksittäiseksi kuviksi.
Unity Light Removal Tool	Etsii ja yrittää poistaa suoravalaistuksen tekstuurista.
ImageMagick	Komentokehoteohjelma, jolla voidaan käsitellä valokuvia.
MeshLab	Ohjelmalla voidaan käsitellä 3D-tiedostoja ja niiden dataa.
Gimp	Käyttöliittymällä varustettu kuvien käsittelyohjelma

Taulukko 2. Taulukossa fotogrammetriatyöskentelyssä käytettäviä ohjelmia. [11.]

3.3 Fotogrammetriavalokuvaaminen

Valokuvaamiseen riittää hyvällä kameralla varustettu älypuhelin, mutta itse valokuvaaminen vaatii harjoittelua. Kuvaamisessa käytettävä tekniikka on riippuvainen kuvattavan kohteen tyypistä. Valokuvaamistekniikat voidaan jakaa kolmeen pääryhmään taso-, kohde- ja tilakuvaaminen.

Tasokuvaamisen kohteena tyypillisesti on maasto, seinä- tai lattiamateriaali. Kohdekuvaamisessa kohteena on usein esine, patsas tai ihminen. Tilakuvaamisella tarkoitetaan sisätilojen, ulkotilojen tai niiden yhdistelmien valokuvaamista. [11, 13.] Kuvassa 5 ja 6 visualisoituna eri tekniikoita.



Kuva 5. Punaiset viivat esittävät kuvattavaa kohdetta ja vihreät kameroiden sijaintia, sekä kuvaussuuntaa. 1 Tasokuvaaminen, 2 Kohdekuvaaminen, 3 Sisätilakuvaaminen ja 4 Tilakuvaaminen.



Kuva 6. Agisoftin kuvaamisohjeesta kuva, jossa näkyy tasokuvaaminen, kohdekuvaaminen sekä tilakuvaaminen. [13.]

Taulukkoon 3 koottiin kokemuksen sekä Agisoftin ja Unityn ohjeiden perusteella listan, joka helpottaa valokuvien ottamista ja vähentää epäonnistumisen riskiä. Ohjeistukset taulukossa 3 [11, 13.]

1	Ota paljon kuvia. (50–200 kpl)	Kuvien määrä on riippuvainen kuvattavan kohteen tyypistä ja koosta. Kuviin on jäätävä riittävästi päällekkäistä informaatiota. Yritä varmistaa, ettei kuvasarjaan synny katvealueita.
2	Käytä mittamerkkiä	Laita kuvattavaan kohteeseen kaksi merkkiä, joiden etäisyys on tiedossa. Näitä merkkejä käyttäen 3D-malli saadaan skaalattua oikean kokoiseksi. Vaihtoehtoisesti merkinä voi toimia mitta tai esine, jonka mitat ovat tiedossa.
3	Pilvinen sää	Kauttaaltaan pilvinen sää tuo esiin kuvattavan kohteen luonnollisen värin. Lisäksi pilvet hajottavat auringon valon siten, ettei kuvattavaan kohteeseen synny tummia ja teräviä varjoja.
4	Minimoi kuvista kohina.	Käytä kuvattaessa mahdollisimman pientä ISO-asetusta kohinan minimoimiseksi.
5	Tallenna kuvat käyttäen RAW tiedostoformaattia, mikäli mahdollista.	RAW-tiedostoformaatin käyttö takaa, että kuviin jää mahdollisimman paljon informaatiota. Mikäli RAW-tiedostoformaatin käyttö ei ole mahdollista, tarkista tukeeko kamera, jotain muuta häviämätöntä raakatiedostoa.
6	Käytä korkearesoluutioista kameraa, vähintään viisi megapikseliä.	Korkeampi resoluutioinen kuva ei välttämättä takaa hyvää lopputulosta, mutta mitä suurempi resoluutioinen kuva, sitä todennäköisemmin se sisältää hyödyllistä dataa.
7	Minimoi linssin vääristys.	Käytä normaaliobjektiivia kuvatessa. Vältä laajakuvaobjektiivin käyttöä. Käytä 35–50 mm polttovälillä olevaa objektiivia

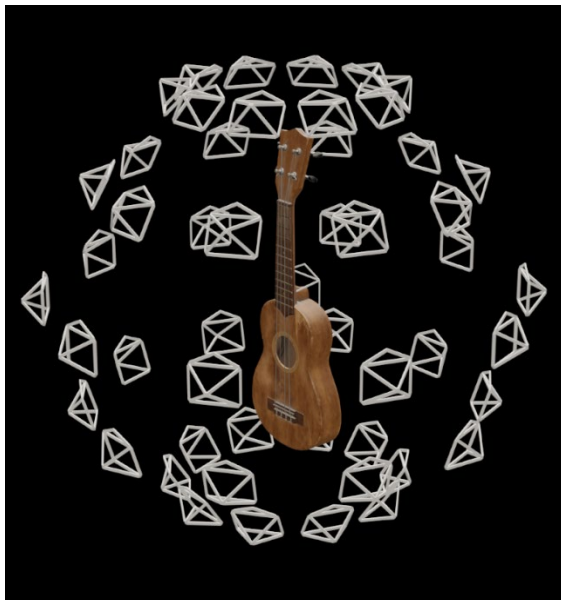
Taulukko 3. Ohjeita fotogrammetriakuvaamiseen.

4 Fotogrammetriastudio

Fotogrammetriastudio on tila tai huone, jossa voidaan suorittaa digitalisoitavan kohteen kuvaaminen mahdollisimman optimissa oloissa ja tarvittavalla laitteistolla. Tyypillisesti fotogrammetriastudioissa kuvattavana kohteena on esineitä, eläimiä ja ihmisiä. Kaupalliseen käyttöön tarkoitettuisia fotogrammetriastudioissa käytetään 50–100 järjestelmäkameraa kohteen kuvaamiseen. [14, 15.]

Ideaalit kuvaolosuhteet ovat käytännössä mahdotonta toteuttaa fyysisessä maailmassa, mutta pystymme sellaisen kuvittelemaan ja simuloimaan digitaalisessa ympäristössä. Hypoteettinen ihannetilanne valokuvaamiseen olisi sellainen, jossa kuvattavan kohteen pystyisi vapaasti kiertämään ympäri, sekä tarkastelemaan sitä yläpuolelta, että alapuolelta. Kuvattavan kohteen tulisi olla jokaisesta kuvakulmasta tasaisesti valaistu ilman, että siihen muodostuisi tummia varjoja. Taustalla ei saisi olla tarkkoja yksityiskohtia eikä se saisi muodostaa heijastuksia kuvattavaan kohteeseen, taikka vääristää kuvattavan kohteen luonnollista väriä.

Valaistuksen tulisi olla pehmeä ja puhtaanvalkoinen. Puhtaanvalkoinen väri mahdollistaa kuvattavan kohteen luonnollisen värintoiston. Pehmeä valaistus mahdollistaa materiaalipinnan luonnollisen varjostumisen. Kuvassa 7 esitetään ihanteelliset olosuhteet kuvaamiselle.



Kuva 7. Visualisointi ideaalisesta kuvaus ympäristöstä.

5 Fotogrammetriastudion suunnittelu

Fotogrammetriastudion perimmäinen tarkoitus on mahdollistaa kuvausympäristö, joka tuottaa toistettavia ja laadukkaita lopputuloksia. Ennen varsinaisen suunnittelun aloittamista haluttiin tutustua koko fotogrammetriatyöskentelyn työnkulkuun ja työskentelyssä käytettäviin ohjelmistoihin. Halusin varmistuksen, että automatisaatio on ylipäättänsä mahdollista toteuttaa.

Suunnittelutyö aloitettiin hankkimalla valotelttä, jonka mukana tuli myös studiovalot, jotka ovat nähtävissä kuvassa 8. Kaapattavan esineen pyörittämiseen käytettiin pyöritettävää tarjotinta. Kuvamiseen käytettiin kolmijalkaa ja Nikon D3200 järjestelmäkameraa.



Kuva 8. Todentamiseen käytetty testikuvaustilanne.

Fotogrammetrian kuvaaminen käsin vaatii harjoittelua ja ensimmäinen sekä toinen kuvaamisyritys epäonnistui. Epäonnistumisien syytä oli liian vähäinen kuvien määrä ja kuvasarjaan jääneet katvealueet. Kuvattaessa on muistettava ottaa mieluummin liian monta kuvaa ja vähentää kuvien määrää tarvittaessa.

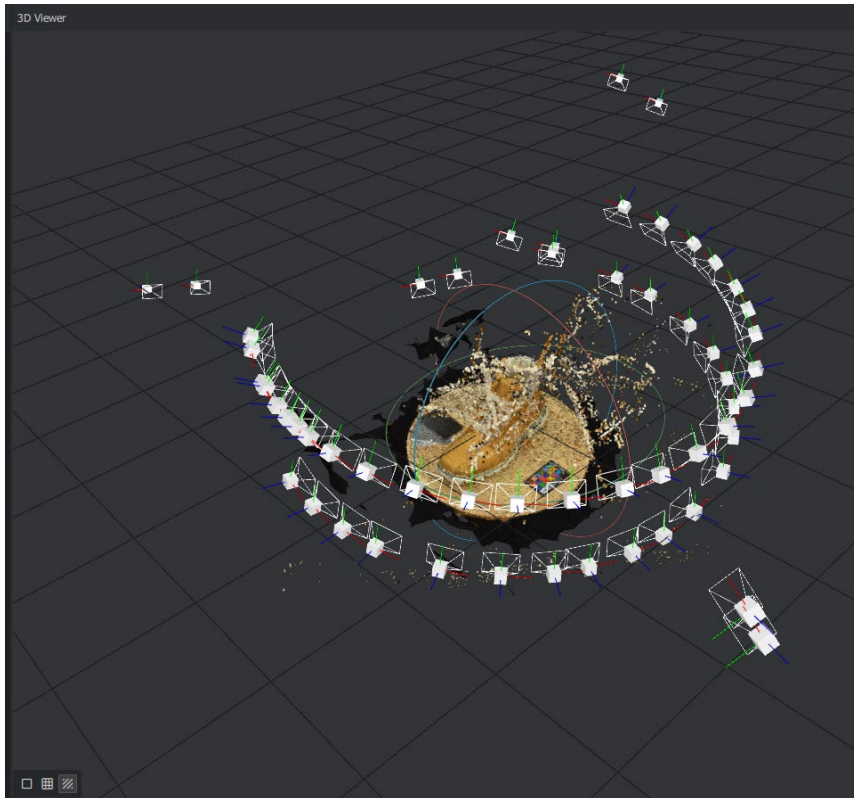
Onnistuneessa kuvasarjassa oli 68 kuvaa. Testikuvaussarjaan jäi parantamisen varaa, mutta tulokset olivat riittävän hyviä projektin jatkamisen kannalta. Kuvasarjaa kehoitettiin Agisoftin Metashapessa sekä avoimeen lähdekoodiin perustuvassa AliceVisionin MeshRoom-ohjelmassa.

Metashape epäonnistui muutaman kuvan kohdalla löytämään ominaisuuksien vastaavuuksia ja siitä syystä se epäonnistui laskemaan muutaman kameran sijainnin, mutta tästä huolimatta lopputulos oli yllättävän hyvä. Kuvassa 9 on nähtävillä digitalisoinnin lopputulos.



Kuva 9. Kuvakaappaus Agisoft Metashape -ohjelmasta. Kuvassa ensimmäinen onnistunut digitalisointi.

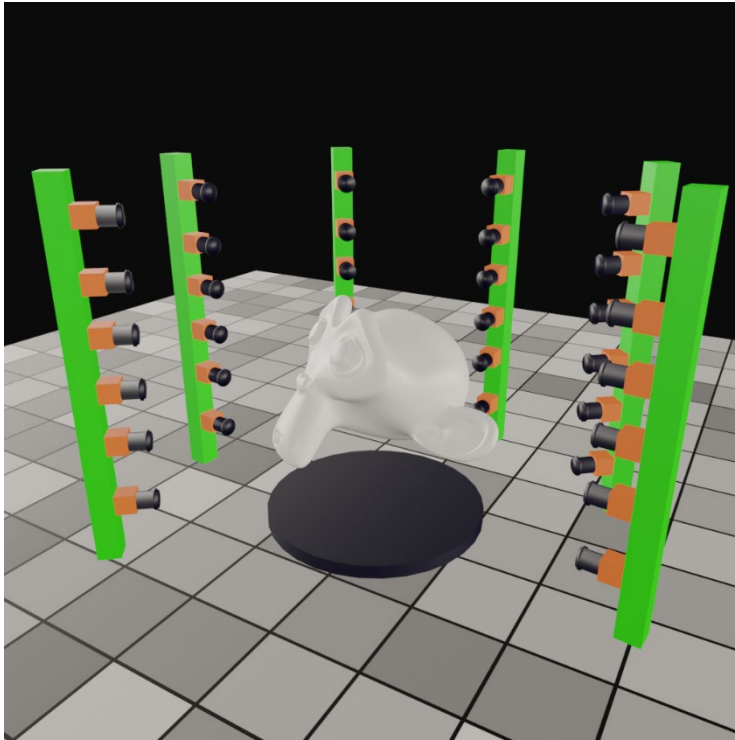
Samaa kuvasarjaa kehoitettiin myös Meshroomissa. Meshroom oli näistä kahdesta ohjelmasta selkeästi herkempi kuvien laadun suhteen, kuten kuva 10 osoittaa. Molemmat ohjelmat onnistuivat tuottamaan riittävän hyvän lopputuloksen ja todentamaan, että fotogrammetriastudion automatisointi on mahdollista. Molempien ohjelmien lopputulokseen olin tyytyväinen ja näiden tuloksien pohjalta oli turvallista edetä automaation suunnitteluun.



Kuva 10. Kuvakaappaus AliceVisionin Meshroom-ohjelmasta. Kuvassa näkyy, kuinka Meshroom suoriutui tässä tapauksessa heikommin verrattuna Metashapeen.

5.1 Studion ja automaation suunnittelu

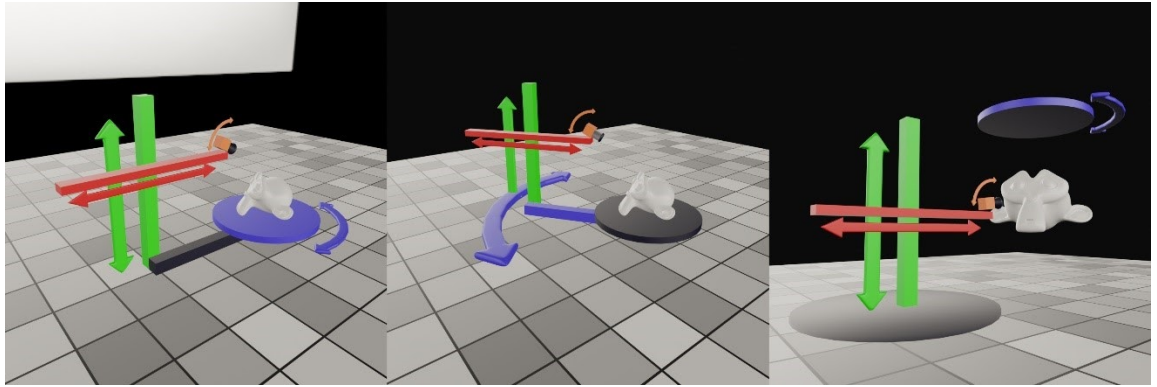
Automaation toteutukselle maksimibudjetiksi asetettiin noin 1000 € ja studion oli mahdollista pieneen huoneeseen. Kaupallisissa ratkaisuissa käytetään useita kameroita, joilla ympäröidään kuvattava kohde. Tämä on nähtävissä kuvassa 11. Usean kameran käyttäminen mahdollistaa liikkuvien kohteiden taltioimisen, kun kaikki kamerat laukaistaan yhtenäkkaisesti.



Kuva 11. Visualisointi tyypillisestä fotogrammetriastudiosta. Kuvattava kohde on ympäröity kameroilla ja valoilla. [14, 15.]

Studion alustava visualisointi toteutettiin 3D-mallinnusohjelma Blenderiä käyttäen. Suunnittelun lähtökohdaksi oli eliminoida kustannuksia mahdollisimman paljon ilman merkittävää laadun menetystä lopputuotteesta. Helpoin tapa kustannuksien säästämiseen oli vähentää kameroiden määrää. Suunnittelun lähtökohdaksi otettiin projektin toteuttaminen yhdellä kameralla. Tämä päätös poissulki liikkuvien kohteiden kuvaamisen.

Blenderillä toteutettiin kolme eri suunnitelmaa, jotka mahdollisesti mahtuisivat valoteltan sisälle. Jokaisessa luonnoksessa oli neljä liikkuvaa osaa. Jokainen luonnos vaatisi kolme kappaletta askelmootoreita ja yhden servon kamerasiirrettävien varten. Laitteisto pyörittäisi kameraa kuvattavan kohteen ympäri tai kamera pysyisi paikallaan ja kuvattava kohde pyöristä alustalla. Kamerasiirrettävien pysty- ja vakaasuuntainen liike saataisiin aikaiseksi käyttäen askelmootoreita ja ohjausruvia. Kuvassa 12 nähtävillä esimerkkejä laitteiston mahdollisista liikeratavariaatioista.



Kuva 12. Visualisointi mahdollisista liikerataratkaisuista.

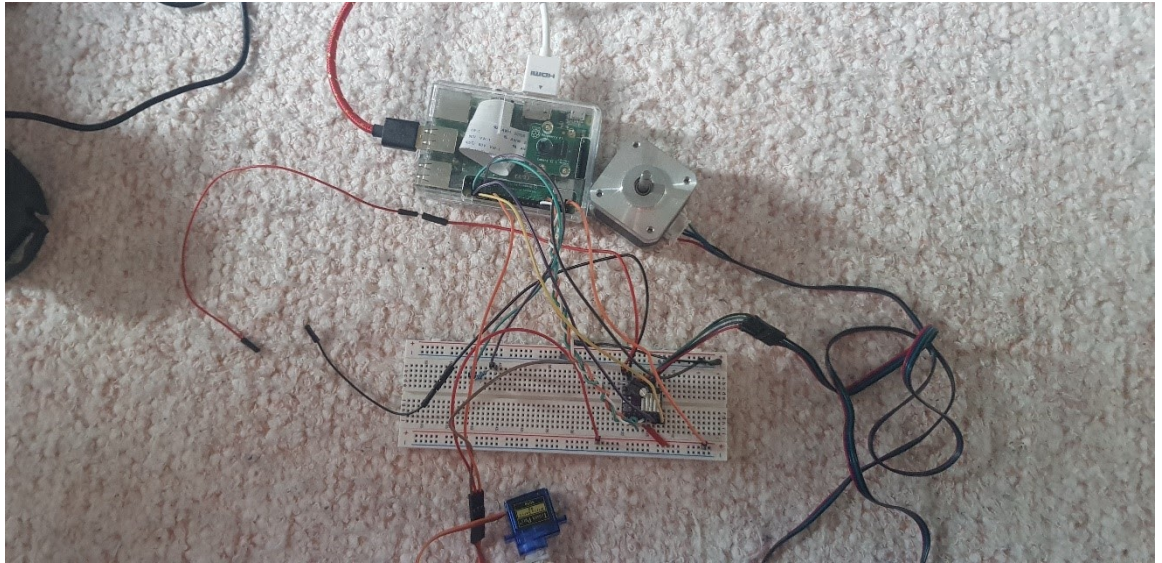
5.2 Kehitysalustan valitseminen

Mahdolliset kehitysalustat voidaan jakaa kahteen pääryhmään, mikrokontrolleihin ja korttitietokoneisiin. Molemmista kategorioista löytyy useita eri valmistajia, ja mallit vaihtelevat ominaisuuksiltaan, teholtaan ja kooltaan.

Alustan valitsemisessa täytyy ottaa huomioon alustaan kytkettävät komponentit, eli moottorit, kytkimet ja kamera. Kuinka alustalle kommunikoidaan: Wi-Fi, ethernet, vai Bluetooth ja onko tarvetta analogiselle signaalille, kuten lämpöanturille. [16 s. 15, 17 s. 17.] Alustan valinnassa hyödynsin Mikrobitti-lehden lehtiartikkeleita ”Luottokortin kokoiset monitaiturit” ja ”Kehitysalustoja joka lähtöön” sekä omaa aikaisempaa kokemustani ja tietämystäni.

Päädyin valitsemaan prototyypin alustaksi Raspberry Pi 3 – korttitietokoneen mikrokontrollerin sijasta, koska korttitietokoneella voi suorittaa useampaa yhtäaikaista ohjelmaa ja siinä on enemmän laskentatehoa datankäsittelyä ja tiedonsiirtoa varten. Lisäksi Raspberry Pi:llä on suuri käyttäjäkunta, jonka ansiosta Raspberry Pi:lle yhteisö on tuottanut paljon valmista opiskelumateriaalia sekä kolmannen osapuolen tuotteita kuten kameroita.

Mikrokontrollerin hyödyt olisivat energiatehokkuus, yksinkertaisempi ohjelmointi ja ohjelmiston suorittamisen tarkkuus. Projektin kannalta oli hyödyllisempää, että alusta on mahdollisimman monipuolinen ja ei itsessään sulje toteutustapoja pois. Siksi päädyin valitsemaan Raspberry Pi:n korttitietokoneen, joka on nähtävissä kuvassa 13. [16, 17.]



Kuva 13. Kuvassa kamera, servo ja askelmoottori kytkettynä Raspberry Pi -korttitietokoneeseen.

5.3 Kameran valitseminen

Korttitietokone mahdollistaa eniten vaihtoehtoja kameran valintaan ja valitsin kameran käyttöönoton helppouden kautta. Vaihtoehtoisia ratkaisuja on käytännössä kolme. Parhaan kuvan saisi modernilla peilillisellä tai peilittömällä järjestelmäkameralla. Järjestelmäkameran käyttäminen takaisi parhaan lopputuloksen visuaalisesti, mutta automaation kannalta se saattaisi olla ongelmallinen.

Järjestelmäkameroista löytyy etähallintamahdollisuuksia joko langattomasti tai USB-väylää pitkin. Mallista riippuen nämä etähallintaominaisuudet rajoittuvat tiedoston siirtoon, kameran asetusten säätöön ja kameran laukaisuun. Useimmissa kameroissa kuvien siirtäminen kameran ulkoiselta muistilta on mahdollista USB-kaapelin kautta. Lisäksi markkinoille on tullut Wi-Fi-sd kortteja, jotka voisivat ratkaista tiedonsiirtoon liittyvät ongelmat.

Etähallintaan liittyvät ongelmat olisivat olleet ratkaistavissa. Vaikeammaksi ongelmaksi muodostui järjestelmäkameran paino ja koko. Järjestelmäkamera olisi vaatinut kalliimpia ja kookkaampia komponentteja kuten servoja ja moottoreita. Lisäksi suuremmat komponentit tarkoittaisivat isompaa prototyyppiä. Tämä olisi ylittänyt aiemmin määritellyn budjetin ja studion maksimitilan.

Toinen vaihtoehto olisi käyttää älypuhelimien kameraa. Nykyisten puhelimen kamerat ovat tarkkoja ja helppokäyttöisiä verrattuna harrastelijatason järjestelmäkameroihin. Automatisointikaan

ei tuottaisi haasteita kommunikaation osalta, mutta puhelimen oma käyttöliittymä ja käyttöön liittyvät asiat toisivat omia haasteita.

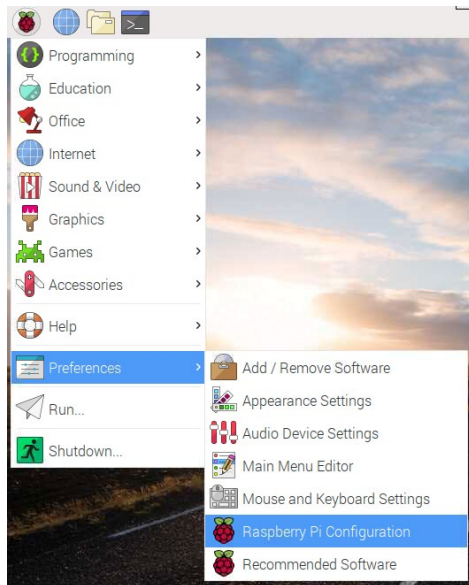
Älypuhelimien käyttö kamerana olisi ollut myös helpompaa, jos alustan olisi toteuttanut mikrokontrollerilla. Edullisessakin puhelimesta on älykäs kamera, joka ottaa laadukkaita kuvia. Kun älypuhelimien kamera takaa laadukkaan kuvan ja laskennallisen suorituskyvyn, voisi alustan liiketarkkuutta kasvattaa käyttämällä mikrokontrolleria. Käytännössä puhelimen ja mikrokontrollerin yhteistyö toimisi siten, että puhelin hoitaisi kuvien ottamisen ja lähettämisen prosessointia varten sekä kertoisi mikrokontrollerille suoritettavat toimenpiteet. Mikrokontrolleri kertoisi puhelimelle takaisin, milloin se on suorittanut kyseiset komennot.

Raspberry Pi - korttitietokoneeseen myydään yhteensopivia kameroita, joita päädyin käyttämään niiden implementoinnin helppouden takia. Näiden kameroiden huono puoli on, että niissä ei ole lainkaan automaattitarkennusta. Tämän puutteen voi korjata helposti laitteiston liikuttamisella x-akselin suuntaisesti.

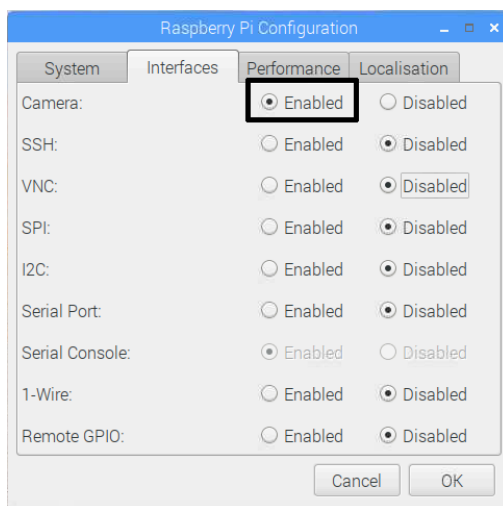
5.4 Raspberry Pi - kameran kytkeminen

Virallisen Raspberry Pi - kameran käyttäminen tarvitsee Raspberry Pi - korttitietokoneen ja sille yhteensopivan kameramoduulin. Kameramoduuli liitetään Raspberry Pi - korttitietokoneeseen käyttäen litteää nauhakaapelia. [18.]

Raspberry Pi - käyttöjärjestelmästä on käyttäjän asetettava kameran käyttöasetus päälle. Työpöydällisessä versiossa kameran saa kytkettyä päälle käyttämällä Raspberry Pi Configuration työkalua. Työkalu löytyy käyttöjärjestelmän päävalikosta: **Preferences->Raspberry Pi Configuration** (kuva 14 ja 15). Asetuksen vaihdon jälkeen on Raspberry Pi käynnistettävä uudelleen.

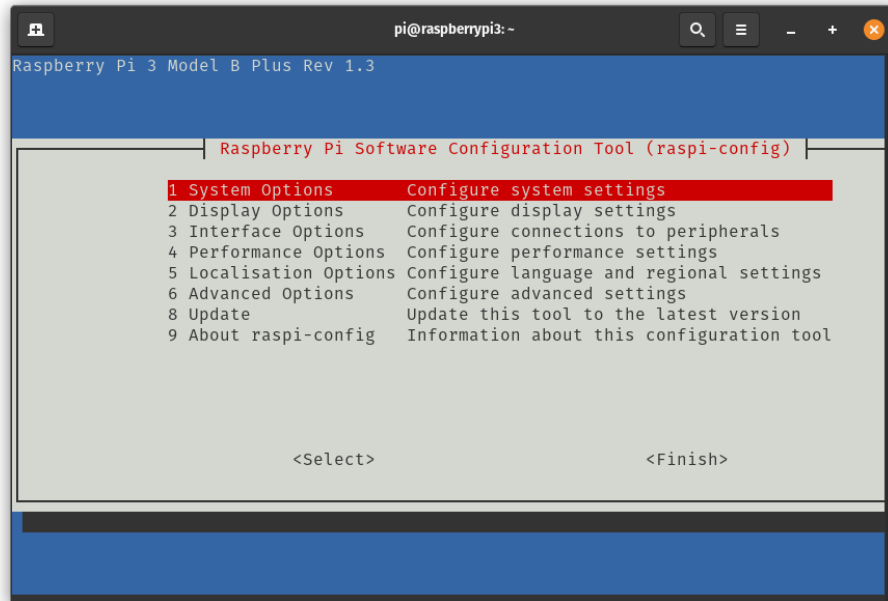


Kuva 14. Konfiguraatiotyökalu sijainti Raspberry Pi - käyttöjärjestelmässä. [18 s. 2.]



Kuva 15. Kamera-asetuksen sijainti konfiguraatiotyökalussa. [18 s. 2.]

Mikäli käyttöjärjestelmässä ei ole graafista käyttöliittymää, voi asetuksen vaihtaa raspi-config -konsoli ohjelmalla. Konsoli ohjelman saa päälle komennolla ”sudo raspi-config”. Raspi-config -ohjelma tarvitsee pääkäyttäjän oikeudet käynnistyäkseen [19]. Asetuksen vaihtaminen tapahtuu navigoimalla käyttäen tabulaattoria, nuolinäppäimiä, enteriä ja esc näppäintä käyttäen. **Asetuksen sijainti on: Interface Options -> P1 Camera -> Yes.** Kuvassa 16 näkyy raspi-config -ohjelman päävalikko. [19.]



Kuva 16. Näkymä raspi-config-päävalikosta.

Raspberry Pi - kameran toimivuus kannattaa testata ennen ohjelmoinnin aloittamista. Kameran toimivuuden pystyy testaamisen käyttäen raspistill-ohjelmaa ja sen voi suorittaa komennolla "raspistill -o Desktop/image.jpg".

Kameran hallitseminen Pythonilla on hyvin yksinkertaista. Käyttöjärjestelmälle tulee kuitenkin asentaa python-picamera ja python3-picamera, jotta python pystyy käyttämään kameraa. [20, s. 28.]

Asentamiseen on kaikista järkevintä käyttää käyttöjärjestelmän omaa pakettienhallintajärjestelmää. Tämä tapahtuu syöttämällä komentokehotteelle komennon "sudo apt-get install python-picamera python3-picamera". Asentaminen pakettienhallintajärjestelmää käyttäen helpottaa picameran päivittämistä ja poistamista, mikäli poistaminen on tarpeellista. Lisäksi se mahdollistaa picameran käyttämisen kaikille järjestelmän käyttäjille. [21.]

Virallisessa Raspberry Pi - kameraoppaasta löytyy kuusi esimerkkiskriptiä. [20, s. 32–33] Myös python-picameran dokumentaatiosta löytyy osio "Basic Recipes". Siellä on esillä laadukkaita skriptiesimerkkejä kameran käytöstä. Koodiesimerkistä 1 nähtävillä, kuinka picameraa käyttäen otetaan kuva. [20, 21.]

```

from time import sleep
from picamera import PiCamera

camera = PiCamera()
camera.resolution = (1024, 768)
camera.start_preview()
# Camera warm-up time
sleep(2)
camera.capture('foo.jpg')

```

Koodiesimerkki 1. Kuvan ottaminen picameralla.

5.5 Askelmoottorin hallinta

Askelmoottorit ovat harjattomia tasavirtamoottoreita, joita voidaan käskyttää askel kerrallaan ilman takaisinkytkentää. Moottoreiden yksittäisten askeleiden määrä kierroksessa vaihtelee moottorin mallista riippuen. Askelmoottorit tarvitsevat askelmoottoriajurin, jotta ne voidaan kytkeä Raspberry Pi - tietokoneeseen ja jotta niitä voidaan hallita ohjelmallisesti. [22.]

Askelmoottoreita myydään NEMA (National Electrical Manufacturers Association) määrittämän kokostandardin mukaan. NEMA määrittää askelmoottorit NEMA-kokoluokkiin, niiden etuseinämän halkaisijan mukaan (Taulukko 4). NEMA-standardi ei huomioi moottorin syvyyttä. [23.]

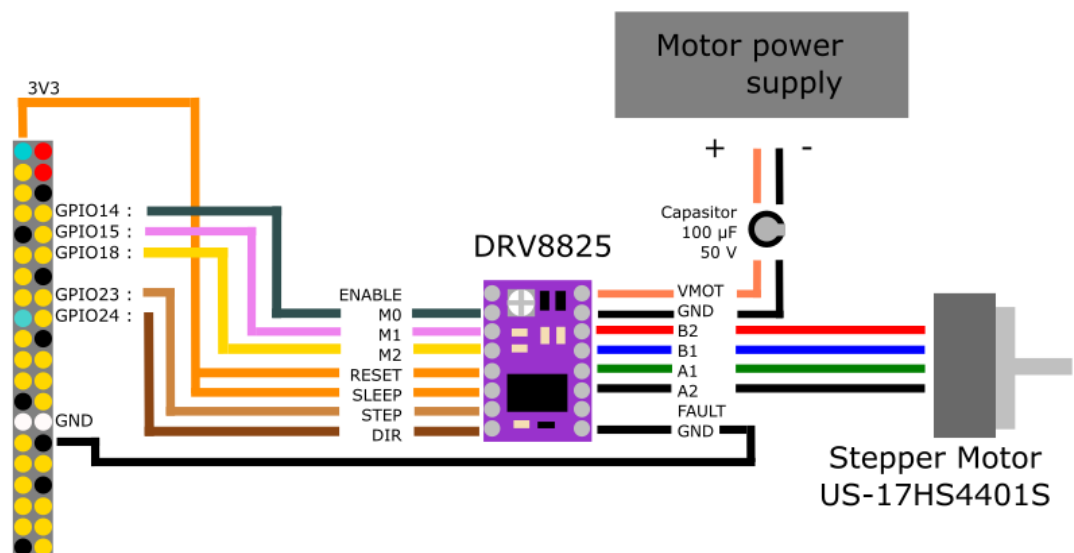
Askelmoottoriksi valittiin Nema17 kokoluokan askelmoottorin US-17HS4401, koska se oli yleisesti käytetty moottori 3D-tulostimissa ja fotogrammetriastudioni tulee olemaan vastaavan kokoinen fyysisesti, että myös vastaavanlainen liikkumiskuormaltaan.

Frame Size	Diameter (mm)	Typical Torque Range for a Stepper Motor (Nm)	Typical Speed Range for a Stepper Motor (RPM)
NEMA 8	20	0.01 - 0.04	0-1000
NEMA 11	28	0.06 - 0.12	0-1000
NEMA 14	35	0.05 - 0.5	0-1000
NEMA 16	39	0.1 - 0.25	0-1000
NEMA 17	43	0.2 - 1	0-1000
NEMA 23	57	0.5 - 3	0-1000
NEMA 24	60	1.2 - 4.6	0-1000
NEMA 34	86	3 - 12	0-1000
NEMA 42	102	12 - 20	0-1000

Taulukko 4. Taulukossa ZikoDriven arvioita askelmoottoreiden väännöstä niiden NEMA-kokoluokittain. [23.]

DRV8825-askelmoottoriajurin kytkemisessä on otettava huomioon virtalähteen ja askelmoottorin suorituskyky. [22.] DRV8825-askelmoottoriajuriin pystyy ottamaan vastaa virtalähteeltä 8.2–45 Volttia. DRV8825-askelmoottoriajuri kykenee maksimissaan antamaan askelmoottorille 2.5 Ampeeria virtaa virtalähteen antaessa 24 V ja ympäristön lämpötilan ollessa 25 °C.

Ennen moottorin kytkemistä on muistettava asettaa DRV8825 jännitteensäätöruuvi oikeaan asentoon. Jännitteensäädinasetus mitataan yleismittarilla koskettamalla maajohdolla virtalähteen maata ja positiivisella johdolla ruuvia, joka on nähtävissä kuvassa 17. Jännitteensäätöruuvi on tärkeä asettaa oikeaan asentoon ennen moottorin kytkemistä. Liian korkea asetus voi vaurioittaa moottorinajuria tai moottoria. Oikein asetettu jännitteensäätöruuvi takaa askelmoottorilla parhaan mahdollisen suorituskyvyn. Mikäli moottori ei tarvitse työssään paljoa vääntöä tai paikallaan pitämistä, voi asetuksen laittaa pienemmälle. [22, 24.]



Kuva 17. Askelmoottorin kytkeminen Raspberry Pi - korttitietokoneeseen.

Askelmoottorit liikkuvat nimensä mukaan askel kerrallaan. US-17HS4401S-moottorissa yksittäinen askel on 1.8° astetta, joten yhdessä kierroksessa on kaksisataa askelta ($360 / 1.8 = 200$.) DRV8825 tukee useita mikroaskelmien muotoja, jotka ovat nähtävillä taulukossa 5. [25.]

$$360 / 1.8 = 200.$$

Table 1. Stepping Format

MODE2	MODE1	MODE0	STEP MODE
0	0	0	Full step (2-phase excitation) with 71% current
0	0	1	1/2 step (1-2 phase excitation)
0	1	0	1/4 step (W1-2 phase excitation)
0	1	1	8 microsteps/step
1	0	0	16 microsteps/step
1	0	1	32 microsteps/step
1	1	0	32 microsteps/step
1	1	1	32 microsteps/step

Taulukko 5. DRV8825-askelmoottoriajurin mikroaskelasetukset. [24.]

Rdaggeri:in tutoriaalin ja komponenttivalmistajien ohjeistuksen mukaan tehtiin Python-skripti, jolla testattiin kytkentöjen toimivuudet. Tämän skriptin tekeminen myös syvensi ymmärrystä askelmoottorin toimimisesta ja askelmoottoriajurin hallinnasta.

Koodiesimerkissä 2 nähtävissä kuinka Python-skripti käy läpi kaikki mikroaskel asetukset aloittaen täydestä askeleesta. Moottori pyörähtää kerran ympäri jokaista mikroaskelasetusta kohden, vaihtaen suuntaa jokaisen pyörähdyksen välissä. [22, 24, 25.]

```

from time import sleep
import RPi.GPIO as GPIO

DIR = 24 # Direction GPIO Pin
STEP = 23 # Step GPIO Pin
SPR = 200 # Steps per Revolution (360 / 1.8)

GPIO.setmode(GPIO.BCM)
GPIO.setup(DIR, GPIO.OUT)
GPIO.setup(STEP, GPIO.OUT)

MODE = (14, 15, 18) # Microstep Resolution GPIO Pins
GPIO.setup(MODE, GPIO.OUT)

RESOLUTION = {'Full': (0, 0, 0),
              'Half': (1, 0, 0),
              '1/4': (0, 1, 0),
              '1/8': (1, 1, 0),
              '1/16': (0, 0, 1),
              '1/32': (1, 0, 1)}
GPIO.output(MODE, RESOLUTION['1/32'])

keys = list(RESOLUTION.keys())
stepValues = [1, 2, 4, 8, 16, 32]
for i in range(keys.__len__()):
    GPIO.output(MODE, RESOLUTION[keys[i]])
    step_count = SPR * stepValues[i]
    delay = .0208 / stepValues[i]
    GPIO.output(DIR, i % 2) # Sets direction

    print("Step mode " + keys[i] + " | Step count " +
          str(step_count) + " | Delay " + str(delay) + " | Dir " + str(i % 2))
    for x in range(step_count):
        GPIO.output(STEP, GPIO.HIGH)
        sleep(delay)
        GPIO.output(STEP, GPIO.LOW)
        sleep(delay)
    sleep(.5)

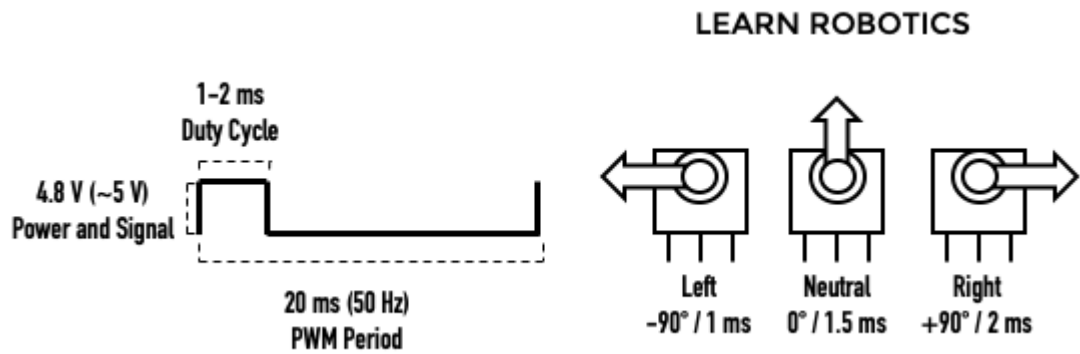
GPIO.cleanup()

```

Koodiesimerkki 2. Rdaggerin ohjeistuksen pohjalta tehty moottoreiden testauskripti.

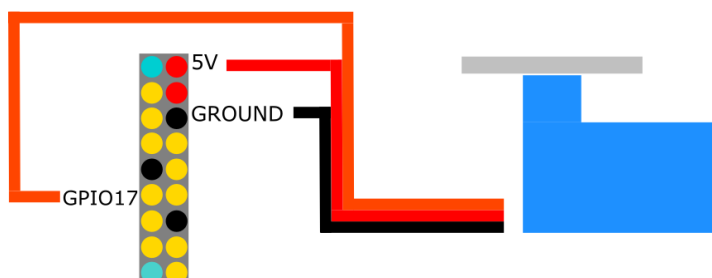
5.6 Servomoottorin hallinta

Servomoottori eroaa askelmoottorista siten, että siinä on takaisinkytkentä, jonka avulla moottori tietää tarkasti asentonsa. Yleensä servomoottorissa on rajoitettu liikerata ja tyypillinen rajoitus on 0 ja 180 asteen väliltä (kuva 18), mutta on myös olemassa servomoottoreita, jotka kykenevät pyörimään jatkuvalla liikkeellä 360 astetta tai niitten liikerata on pienempi kuin 180 astetta. [26.]



Kuva 18. Visualisointi SG90 9g - servon asennoista suhteessa työsignaalin pituuteen. [26.]

Tyypillisissä harrasteservomootoreissa on kolme kytkentäjohtoa. Positiivisen ja negatiivisen johdon lisäksi niistä löytyy signaalijohto (kuva 19). Signaalijohtoa pitkin moottorille kerrotaan tavoiteasento. Tähän käytetään pulssinleveysmodulaatiota, joka paremmin tunnetaan lyhenteenä PWM (Pulse-Width Modulation). [26.]



Kuva 19. Servon kytkeminen Raspberry Pi - korttitietokoneeseen. [27.]

Servon kalibroiminen tapahtuu siten, että tarkastetaan käytettävän servon datalehdestä, minkä pituisen PWM-signaalin se tarvitsee ja kuinka pitkä työsignaali vaaditaan kuhunkin ääriasentoon. SG90-servon datalehdestä [Liite 1] käy ilmi, että moottori on ääriasennossa vasemmalle ~ 1 ms työsignaalilla ja vastaavasti ääriasennossa oikealle ~ 2 ms pituisella työsignaalilla. Tämä työsykli-alue saadaan selville jakolaskulla. Esimerkiksi vasen ääriasento kestää yhden millisekunnin, joten $1 / 20 \times 100 \% = 5 \%$ ja vastaavasti ääriasento oikealle kestää kaksi millisekuntia, joten $2 / 20 \times 100 \% = 10 \%$. [26.]

Raspberry Pi:ssä kaikki GPIO-pinnit tukevat pulssinleveysmodulaatiota ohjelmallisesti, mikäli servon asennon tarkkuudella on merkitystä, käytä laitteistohallinnallista pulssinleveysmodulaatiota.

Raspberry Pi:ssä on neljä GPIO-pinniä, joita pystyy laitteistollisesti hallitsemaan nämä pinnit ovat GPIO12, GPIO13, GPIO18, GPIO19. [28.] Kytin servon Raspberry Pi:hin kuvan 19 mukaisesti.

Aluksi käytin ohjelmaesimerkissäni lukuja 5 ja 10, mutta jostain syystä moottorin liikerata oli vajavainen, joten manuaalisesti korjasin Barnatt Christopherin ohjeen mukaan liikealueen käyttämään lukuja 2–12, jotka ovat nähtävissä koodiesimerkissä 3. [27.]

```

import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(17,GPIO.OUT)
servo = GPIO.PWM(17,50) # Note 11 is pin, 50 = 50Hz pulse
servo.start(0)

def goToAngle(angle : float):
    duty = angle / 18 + 2
    servo.ChangeDutyCycle(duty)
    time.sleep(0.3)
    servo.ChangeDutyCycle(0)
    time.sleep(0.7)
    duty = duty + 1

servo.ChangeDutyCycle(2) # left -90 deg position
time.sleep(1)
servo.ChangeDutyCycle(7) # neutral position
time.sleep(1)
servo.ChangeDutyCycle(12) # right +90 deg position
time.sleep(1)

for x in range(10):
    servo.ChangeDutyCycle(x+2)
    time.sleep(0.3)
    servo.ChangeDutyCycle(0)
    time.sleep(0.7)

# Wait a couple of seconds
time.sleep(1)
print ("Go to 0")
goToAngle(0)
time.sleep(1)
print ("Go to 180")
goToAngle(180)
time.sleep(1)
print ("Go to 90")
goToAngle(90)
time.sleep(1)
print ("Go to 0")
goToAngle(90)
time.sleep(1)

servo.stop()
GPIO.cleanup()

```

Koodiesimerkki 3. Testiskriptillä voidaan selvittää servon toiminnallisuus.

5.7 Prototyypin rakentaminen

Prototyypin tarkoituksena oli varmentaa koko prosessin toimivuus mahdollisimman edullisilla hankinnoilla sekä helpottaa laitteiston suunnittelua. Prototyyppi rakennettiin mahdollisimman edullisista ja välittömästi saatavilla olevista osista. Prototyypin runko rakennettiin lattialistasta ja pahvista (kuva 20). Prototyyppi koostui yhdestä johtoruuvista ja kahdesta askelmoottorista. Prototyypillä pystyin varmistamaan automaatio-ohjelman toimivuuden ja lopputulos on nähtävillä kuvassa 21. Taulukossa 6 prototyypin osalista.



Kuva 20. Kuvassa automatisoitu fotogrammetriastudioprototyyppi.



Kuva 21. Kuvakaappaus prototyypin kuvaamasta lopputuotteesta.

Raspberry Pi 3 B
Raspberry Pi Camera V2
Ikean pyörivä tarjotin
Askelmoottori Nema17 x 2
Johtoruuvi ja laakerit / pidikkeet
Servo moottori SG90
DRV8825 Askelmoottoriajuri x 2
Koekytkenälevy x 2
Hammasyörä 12 hammasta
Hammasyörä 32 hammasta
6 mm hihna

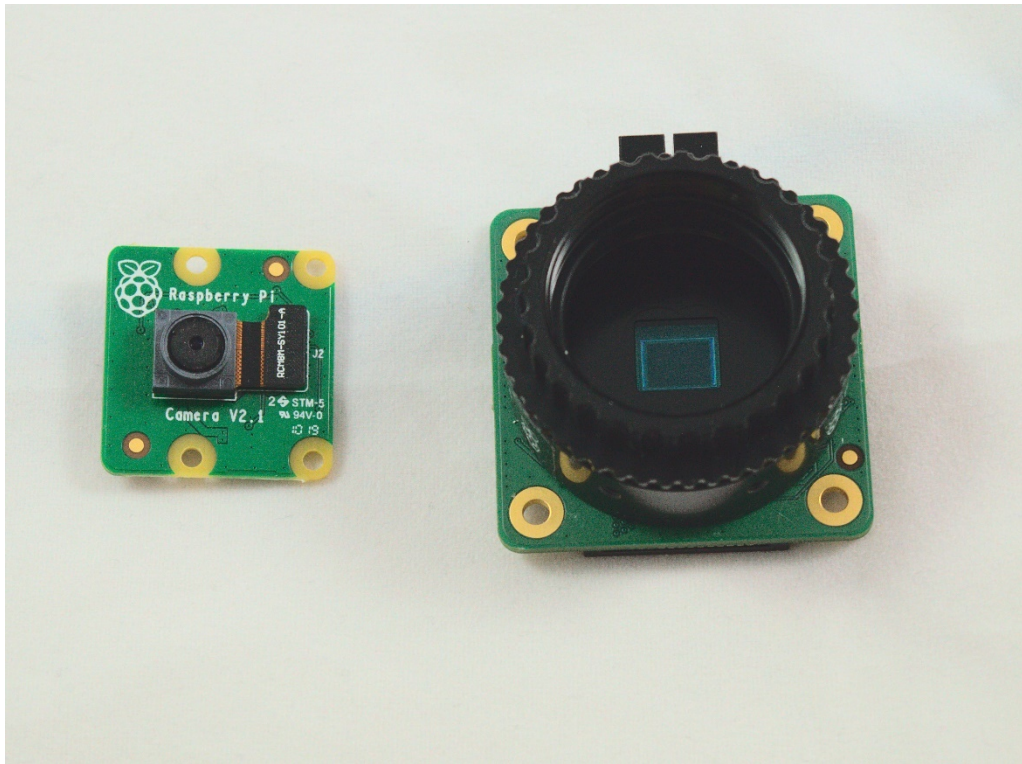
Taulukko 6. Prototyypissä käytetyt osat.

6 Fotogrammetriastudion toteutus

Studion tarkoituksena on mahdollistaa kuvausympäristö, joka tuottaa toistettavia lopputuloksia. Aikaisemmin määritelty budjetti rajoitti studion kokoa, mutta hyvänä puolena pienessä studiossa on, että muutoksien tekeminen on helpompaa ja päivityskustannukset ovat halvempia.

Automatisoidussa fotogrammetriastudion toteutuksessa käytettiin hyödyksi kaikkea aikaisemmin opittua. Prototyyppiä käyttäessä huomattiin, kuinka tärkeää on saada otettua laadukkaita kuvia, joten päätettiin päivittää kameran Raspberry Pi High Quality -kameraan, jossa on huomattavasti edeltäjänsä suurempi sensori ja lisäksi mahdollisuus vaihtaa objektiivia.

Kameran päivittäminen malliin, jossa on vaihdettava objektiivi, toi kamerarakenteeseen huomattavan määrän lisää painoa. Vanhempi sekä uudempi kamera nähtävissä kuvassa 22 ja uuteen kameraan sopivat objektiivit nähtävissä kuvassa 23. Servo vaihdettiin kookkaampaan ja tehokkaampaan malliin, jotta kameran kääntömekanismiin olisi helpompi suunnitella ja toteuttaa. Lisäksi Raspberry Pi 3 päivitettiin uudempaan Raspberry Pi 4 -versioon paremman suorituskyvyn toivossa.



Kuva 22. Vasemmalla kamera V2.1 ja oikealla uudempi High Quality-kamera.



Kuva 23. Kuvassa HQ-kameraan yhteensopivia objektiiveja.

6.1 CAD-suunnittelu

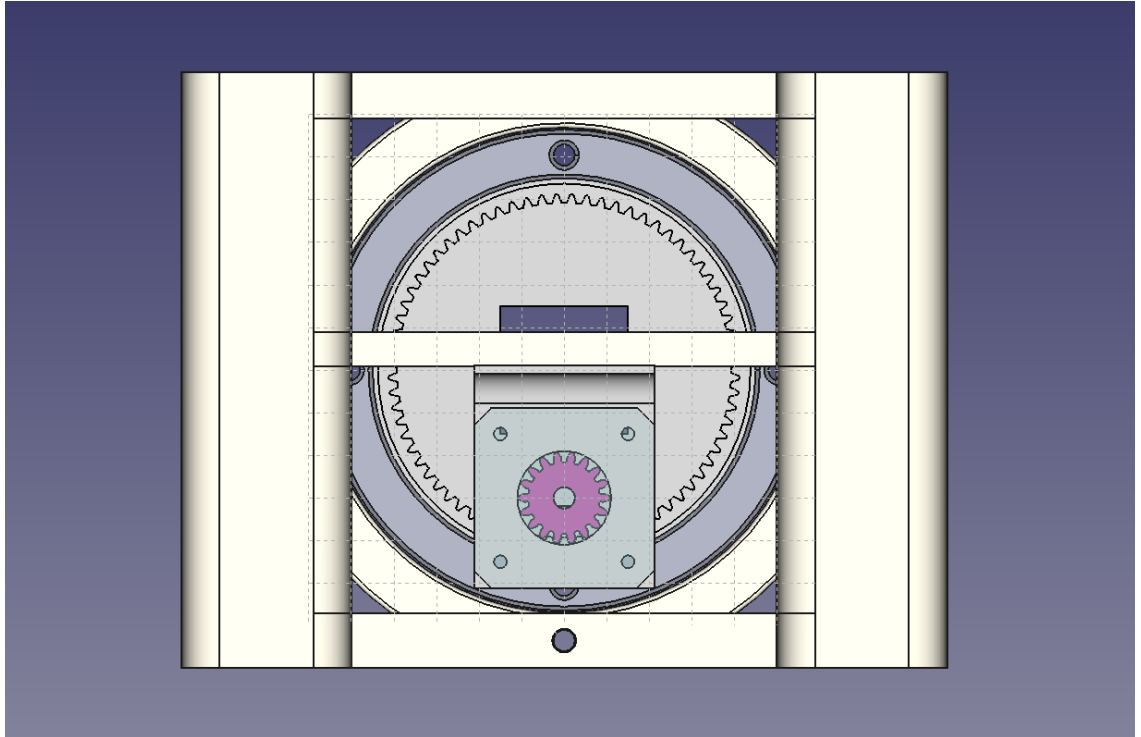
Fotogrammetriastudion CAD-suunnittelu perustui prototyypin rakentamisesta ja suunnittelusta opittuun tietoon. Inspiraationa ja suunnittelupohjana käytettiin kuluttajamarkkinoilla olevia 3D-tulostimia.

Osat suunniteltiin FreeCAD-ohjelmistolla ja ne tulostettiin käyttäen Creality Ender 3 V2 3D-tulostinta. Aikaisempaa kokemusta CAD-suunnittelusta ei ollut, joten tein paljon suunnittelu virheitä. 3D-tulostaminen onneksi mahdollistaa nopean iteroimisen.

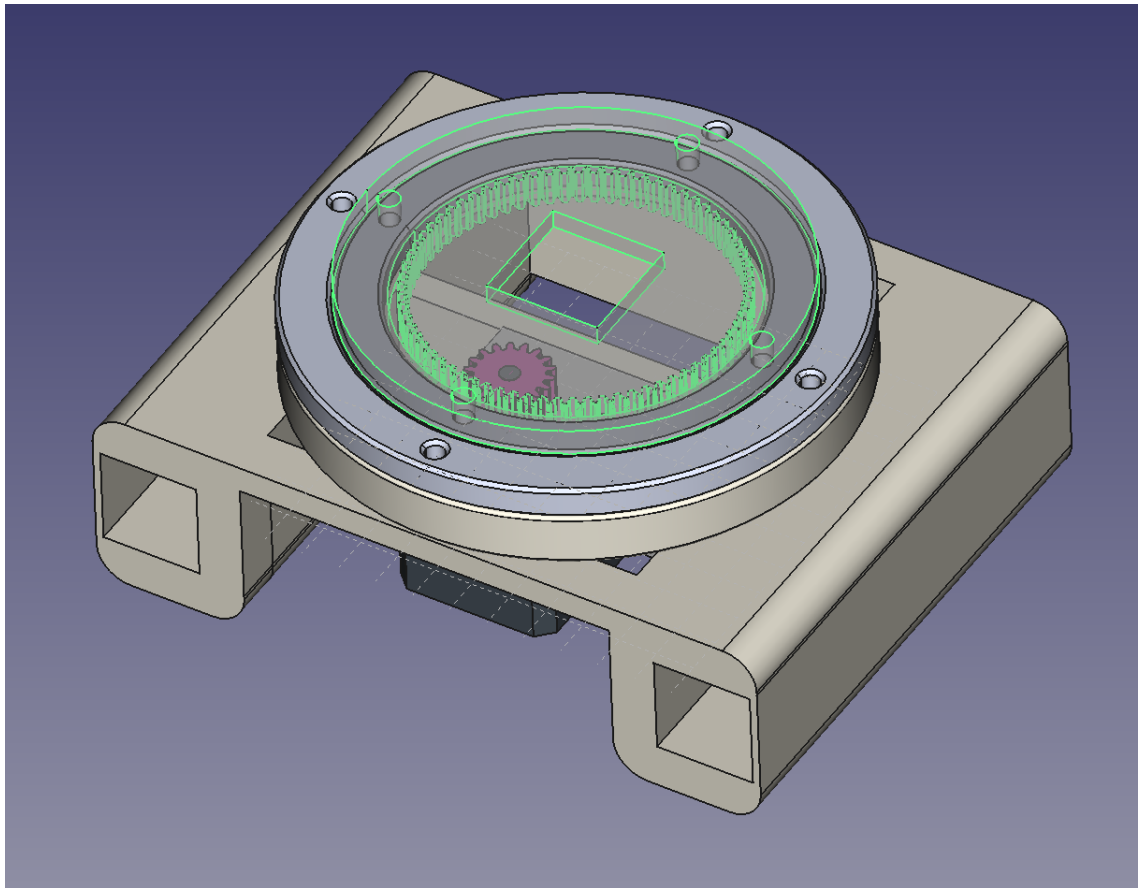
CAD-suunnittelussa täytyi ottaa huomioon käytettävien osien koko sekä niiden kiinnittäminen tulostettaviin osiin. Suunnittelutyö aloitettiin pyörimisalustasta, koska ajateltiin, että siinä on kaikista vähiten suunnittelussa huomioonotettavia muuttujia, sillä siihen kiinnittyy ainoastaan askelmoottori ja laakeri.

Askelmoottorit liikkuvat luonnostaan epätasaisesti ja 17HS4401S-askelmoottorissa on vain 200 askelta. Pyörimisalusta haluttiin liikkuvan mahdollisimman tasaisesti, joten pyörimisalustaan

suunniteltiin hammasrattaisto, jolla saatiin enemmän askelia kierrokselle. Askelmoottoriin suunniteltiin 18-hampainen ratas ja vastarattaaksi suunniteltiin 72-hampainen ratas, joka kiinnittyi pyörimisalustaan (kuva 24 ja 25). Näin ollen välityssuhteeksi muodostui 0.25: 1 tarkoittaen sitä, että kierroksessa on 800 askelta ja 32 mikroaskelta saadaan kokonaisaskelluvuksi 25 600 askelta.



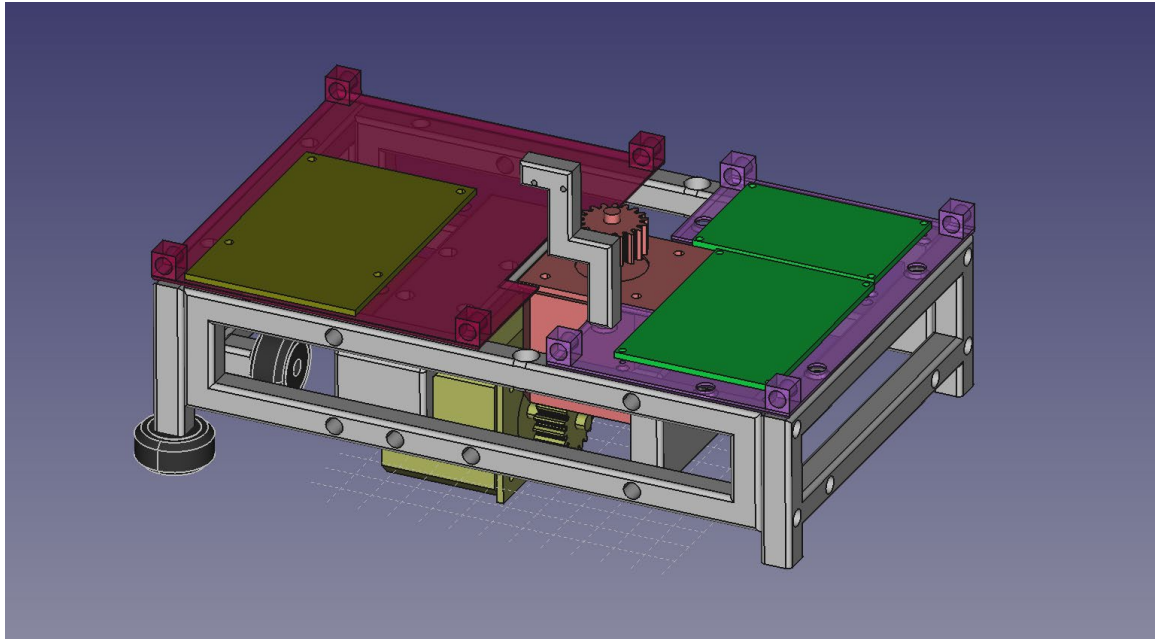
Kuva 24. Kuvassa moottorisoitu pyörimisalusta alhaaltapäin kuvattuna FreeCAD-ohjelmassa.



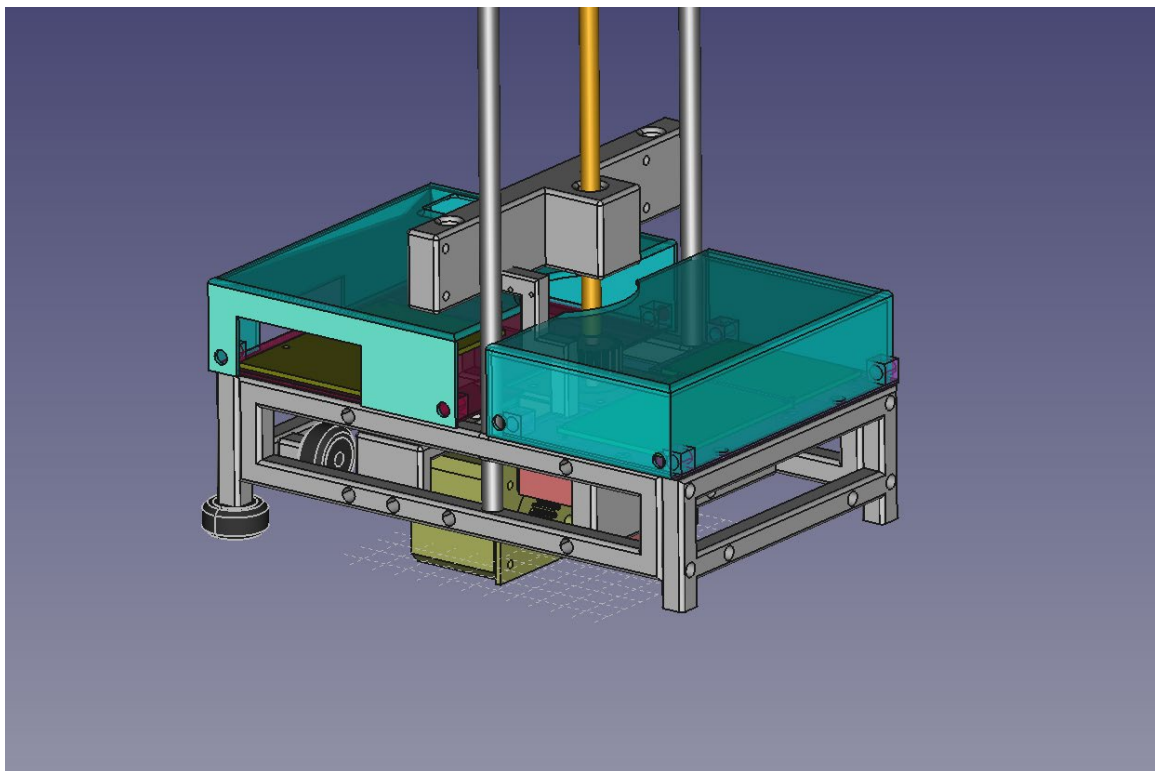
Kuva 25. Kuvassa pyörimisalusta ja isompi hammasratas esitetty osittain läpinäkyvä.

Runkokehikon tehtävä oli kyetä liikuttamaan kameraa. Runko-osa suunniteltiin kulkemaan neliönmuotoisilla putkilla. Rungon painopisteen suunniteltiin olemaan mahdollisimman matalalla, jotta se ei pääsisi kaatumaan kameran ollessa yläasennossa. Lisäturvaksi suunniteltiin liikkumishihna kulkemaan rungon läpi siten, ettei kaatuminen ole mahdollista.

Kuvassa 26 keltainen levy kuvastaa Raspberry Pi:n sijaintia ja vihreät levyt kuvastavat moottoriajuriin kiinnityspaikkoja. Keltainen moottori liikuttaa runkoa eteenpäin / taaksepäin ja punainen moottori ajaa johtoruuvia, joka liikuttaa kameratelinettä ylös / alas. Johtoruuvi ja kamerateline nähtävissä kuvassa 27.



Kuva 26. Runkokehikko, keltainen levy kuvastaa Raspberry Pi:n kiinnityspaikkaa ja vihreät levyt kuvastavat moottoriajuriin kiinnityspaikkoja.

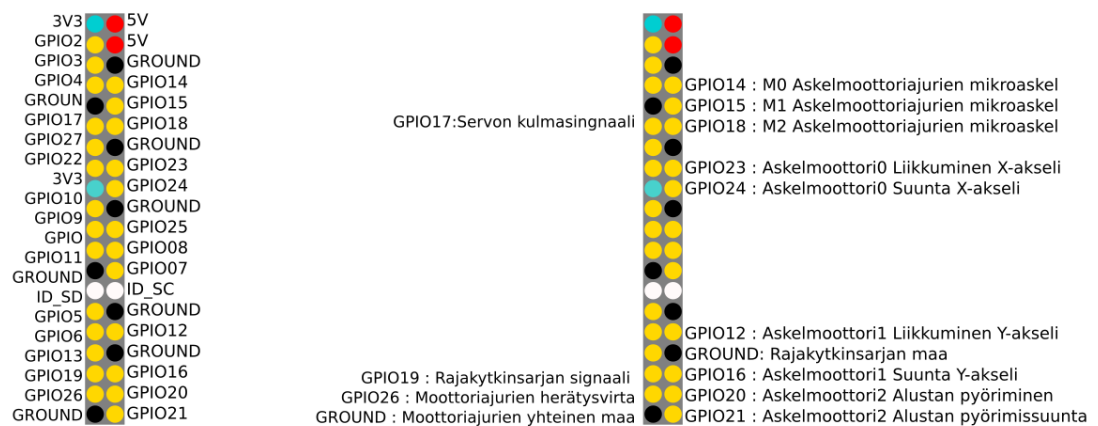


Kuva 27. Kuvakaappaus rungosta ja kameratelineestä.

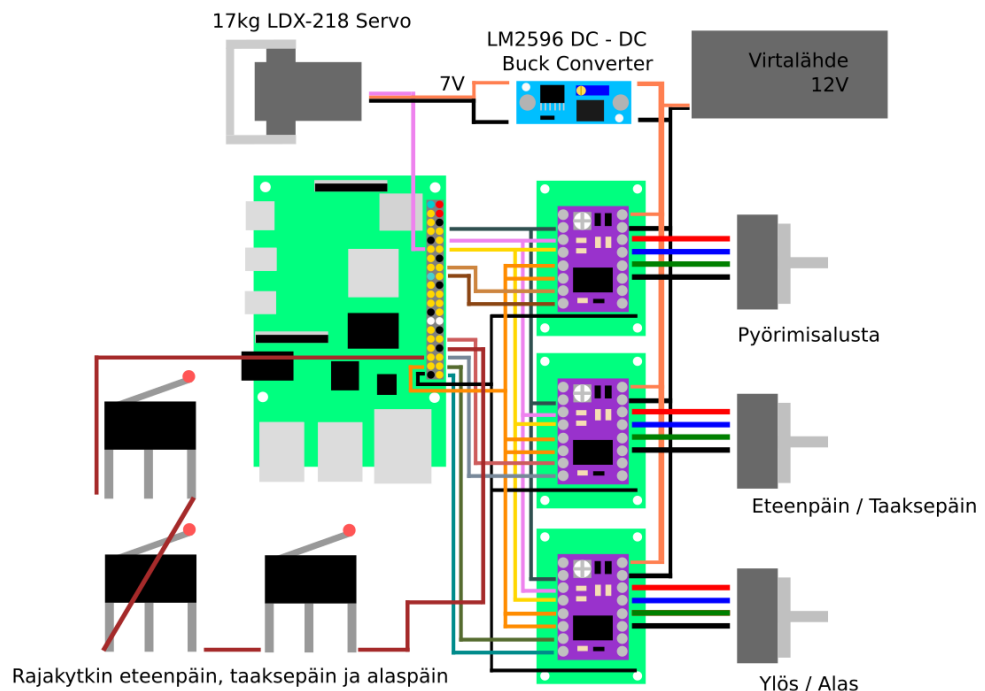
6.2 Komponenttien kytkentä

Lopullisessa toteutuksessa päädyttiin käyttämään suunnitelman mukaisesti kolme askelmoottoria, joiden kytkentäpaikat ja tavat ovat nähtävissä kuvissa 28 ja 29. Suunnitelmasta poiketen kytkentöihin lisättiin rajakytkimet, etteivät liikkuvien osien liikeradat pääse ylittymään ja täten mahdollisesti vaurioittamaan komponentteja tai tulostettuja 3D-osia. Lisäksi uusi servomoottori vaati 7 voltin toimintavirran. Koska Raspberry Pi:ssä ei ole 7 voltin ulostuloa ja käytin 12 voltin virtalähdettä, jouduttiin käyttämään konvertertia, jolla virransyötön pystyy muuttamaan 7 voltin ulostuloksi.

Kytkentöjen tekeminen oli haastavaa, koska liitännät olivat hyvin lähellä toisiaan. Askelmoottoriajurien mikroaskeljohdot ja herätevirta kytkettiin siten, että kaikkia askelmoottoriajureiden asetuksia pystyttiin hallinnoimaan samoista GPIO-pinneistä. Tämä päätös perustui siihen ajatukseen, että vain yhtä askelmoottoria tullaan käyttämään kerrallaan. Virransäästösyistä herätevirta olisi voitaisiin erotella jokaiselle askelmoottoriajurille erikseen, mutta tämä lisäisi johtojen määrää ja se pitäisi ottaa huomioon myös ohjelmoitaessa. Kuvassa 29 nähtävissä kuinka GPIO-pinnit 14, 15, 18 ja 26 ovat kytketty askelmoottoriajureihin.



Kuva 28. Raspberry Pi GPIO-pinnien nimet ja komponenttien kytkentäkohdat.

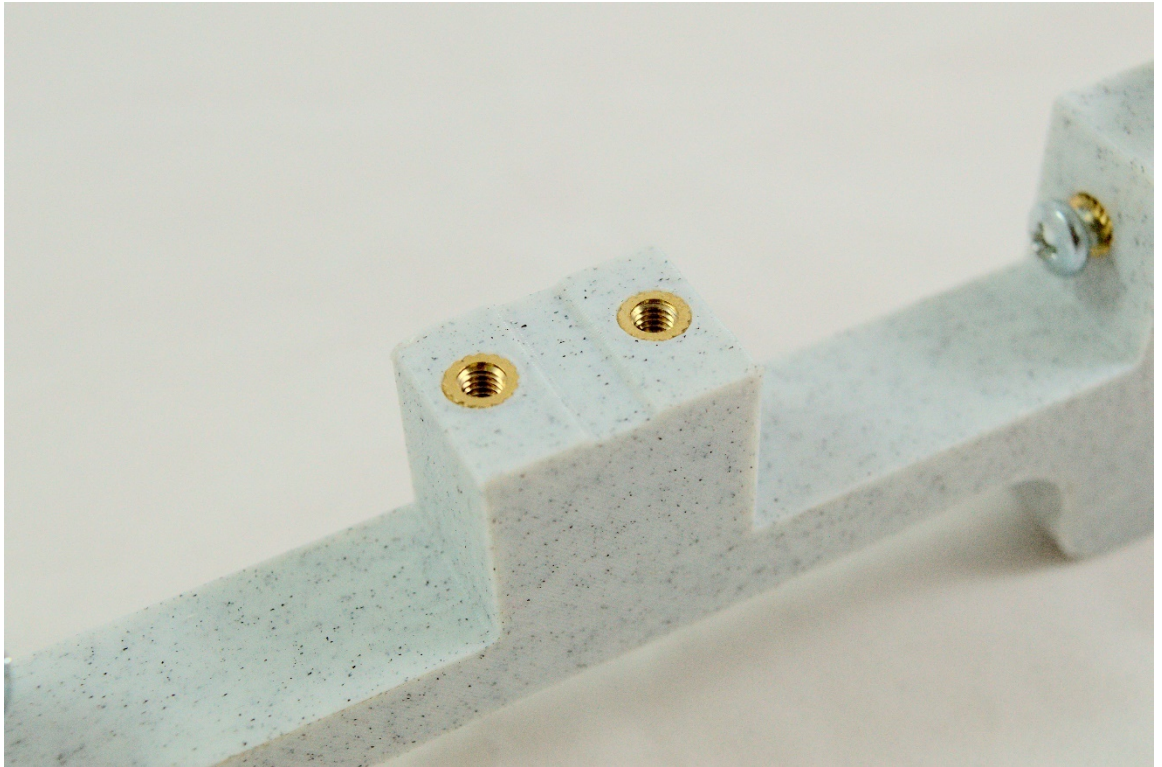


Kuva 29. Komponenttien kytkeminen Raspberry Pi - korttitietokoneeseen.

6.3 3D-Tulostaminen

FreeCAD-ohjelmasta suunnitellut osat vietiin STL-tiedostoformaattia käyttäen CURA-viipalointiohjelmaan. Viipalointiohjelman tarkoituksena on muuttaa STL-tiedostoformaatti 3D-tulostimelle tulostusohjeeksi.

Osia tulostettiin käyttäen Ender 3 V2 3D -tulostinta käyttäen. Materiaalina käytettiin pääosin valkoista PLA- tai PETG-muovia, koska ne ovat muovityyppejä, joita on helppo tulostaa. Molemmilla muovityypeillä on matala sulamispiste, joka helpottaa upotekierteiden upottamista muoviin (kuva 30).



Kuva 30. Lähikuva 3D-tulostetussa osasta. Osiin jätettiin upotekierteille kolot ja ne upotettiin muoviin käyttäen kolvia.

6.4 Automaation ohjelmointi

Automaatio toteutettiin käyttäen Python-ohjelmointikieltä. Automaation ohjelmiston toteutuksessa on useita haasteita. Ensimmäinen haaste on valokuvien säilyttäminen. Fotogrammetriassa kohteesta otetaan useita kuvia ja näiden kuvien säilyttäminen vie paljon tilaa.

Raspberry Pi 4 käyttää massamuistina SD-korttia. SD-kortin käyttö massamuistina lyhentää kortin elinikää, mikä vähentää laitteen luotettavuutta. Valitettavasti virallinen Raspberry Pi 4 tukee huonosti muita massamuistivaihtoehtoja ja niiden käyttö on hankalampaa.

Toinen haaste on fotogrammetriadatan käsittely, eli valokuvien yhtenäisten pisteiden tunnistaminen ja näistä pistepilven laskeminen. Tämä datan käsittely on laskennallisesti raskasta ja käsittelyn keston pituus vaihtelee kuvien laadun ja määrän mukaan. [29.] Teoriassa Raspberry Pi:llä olisi mahdollista toteuttaa koko tuotantoprosessi kuvaamisesta pelivalmiiseen 3D-malliin. Raspberry Pi:n vähäinen laskentateho ja muisti tarkoittaisi liian pitkiä laskenta aikoja, jotta se olisi käytännöllistä. Toteutin fotogrammetriastudion ympäristön siten, että Raspberry Pi huolehtii kuvien

ottamisen ja niiden siirtämisen tiedostopalvelimella ja rekonstruktio-ohjelman ajetaan tehokkaalla pöytäkoneella.

Kolmantena haasteena on ratkaista käyttäjän ja alustan välinen kommunikointi. Yksinkertaisin tapa komentaa alustaa on käyttää ssh-protokollaa. Ratkaisu ei ole hirveän elegantti ja jatkuvien komentosarjojen antaminen on hankalaa kuten näppäimen painallus ja sen pohjassa pitäminen. Vaihtoehtoisesti komentoja voitaisiin antaa puhelimella, joka on kytketty samaan lähiverkkoon Raspberry Pi:n kanssa, mutta tämän toteuttaminen vaatisi paljon ohjelmointityötä. Raspberry Pi - korttitietokoneessa on Bluethooth, joka mahdollistaa langattomien peliohjaimien kytkemisen. Päädyin ratkaisussani käyttämään ssh-protokollaa ja Bluethooth-peliohjainta.

Studion käyttäjä asettaa kuvattavan kohteen moottorisoidulle kuvaalustalle, minkä jälkeen käyttäjä käynnistää automaatio-ohjelman (koodiesimerkki 4). Ennen kuvaamista ohjelma kysyy käyttäjältä kuvattavan kohteen nimen sekä kuinka monta kuvaa otetaan kierroksen aikana. Tämä tieto kerrotaan automaatio-ohjelmalle käyttäen ssh-protokollaa. Automaatio-ohjelma kysyy käyttäjältä kuvaamisessa käytettävät kuvakulmat, jotka hän voi syöttää käyttäen Bluethooth-ohjainta. Tämän jälkeen automaatio-ohjelma ottaa käyttäjän antamien ohjeiden mukaan kuvia, siirtäen kuvat säilytettäväksi tiedostopalvelimelle.

```

import cameraServo
import stepMotors
import fileServer
import userInput
import cameraController
from position import position

positions = list(position)
picturesPerRotation = 0

if(fileServer.fileServeIsOnline):
    objectName = userInput.askObjectName()
    fileServer.createProjectFolder(objectName)
    cameraServo.goToAngle(90)
    stepMotors.findHome()
    wantedPicturesPerRotation = userInput.askPicturesPerRotation()
    pictureInStep = int(round(stepMotors.getRotationStageStepCount()
/ wantedPicturesPerRotation))
    positions = userInput.askPositionsAndCameraAngle()
    for pos in positions:
        cameraServo.goToAngle(pos.cameraAngle)
        stepMotors.goToPosition(pos)
        for i in range(stepMotors.getRotationStageStepCount()):
            stepMotors.rotateRotationStage()
            if i % pictureInStep:
                cameraController.takePicture()
        stepMotors.findHome()
else:
    print("Cant find file server")

```

Koodiesimerkki 4. Studion automatisaatio yksinkertaistettuna.

7 Tulokset

Fotogrammetriastudiolla saatiin otettua useita onnistuneita kuvasarjoja, joista jälleenrakennus ohjelmat onnistuivat laskemaan pelituotantokäyttöön soveltuvia polygonimalleja. Tarkasteltavaksi valittiin kolme erittäin hyvin onnistunutta skannausta.

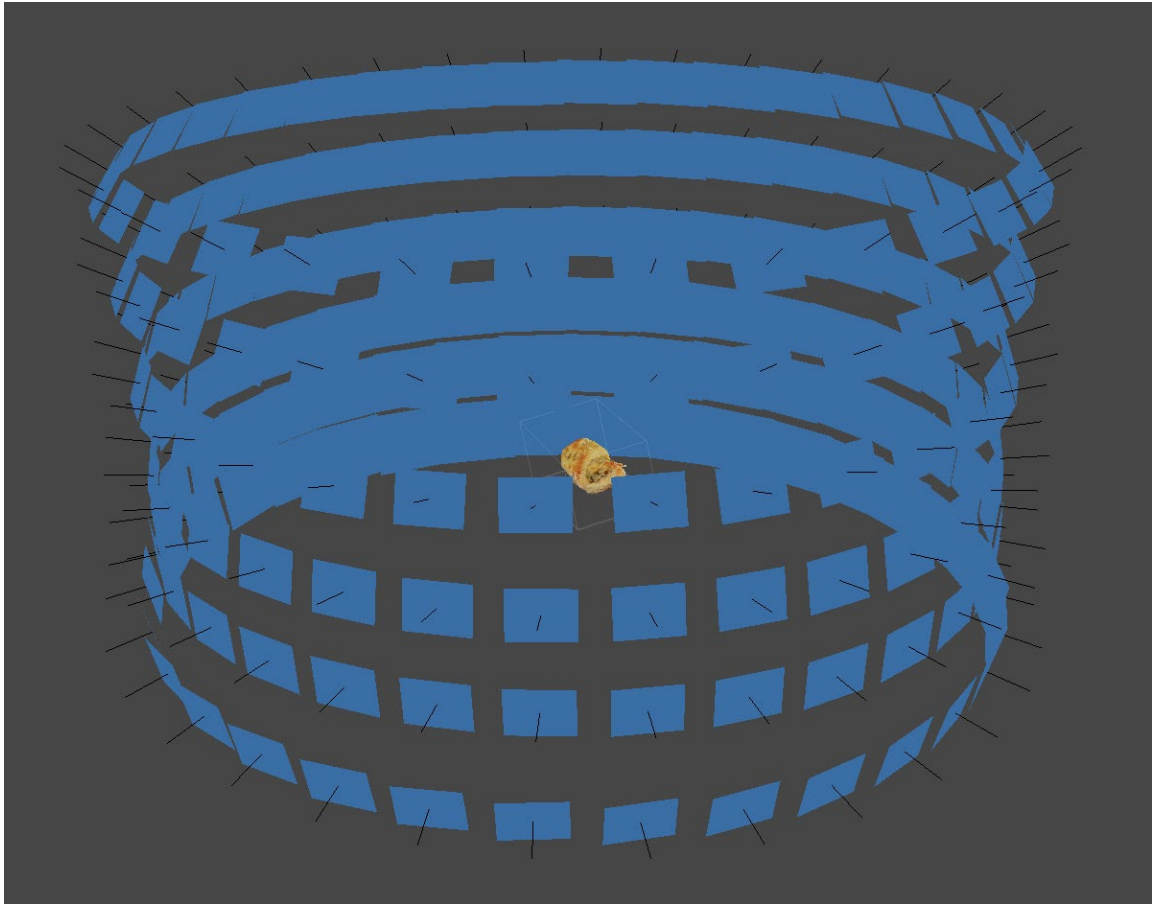
Huomioi kuvista 31 ja 32 kuinka tarkasti jälleenrakennusohjelma sai asetettua kameroiden sijainnit, sekä miten korkealaatuinen tekstuuri molemmissa polygonimalleissa on. Tekstuurit olivat sekä teräviä, että väriltään luonnollisia. Taulukoissa 7 ja 8 on kyseisten kuvaustilanteiden tekniset tiedot Metashape-ohjelmasta. Kuvissa 33 ja 34 nähtävillä kuvaustilanteiden 31 ja 32 lopputuotteet. Lisäksi kuvassa 35 Blenderillä viimeistelty hahmonnus.

Sinihomejuustosarvi
Kameroita 192 kpl kuudessa kerroksessa
Tiheäpistepilvi koostui 1 925 801 pisteestä
3D-mallissa oli 385 160 polygonia

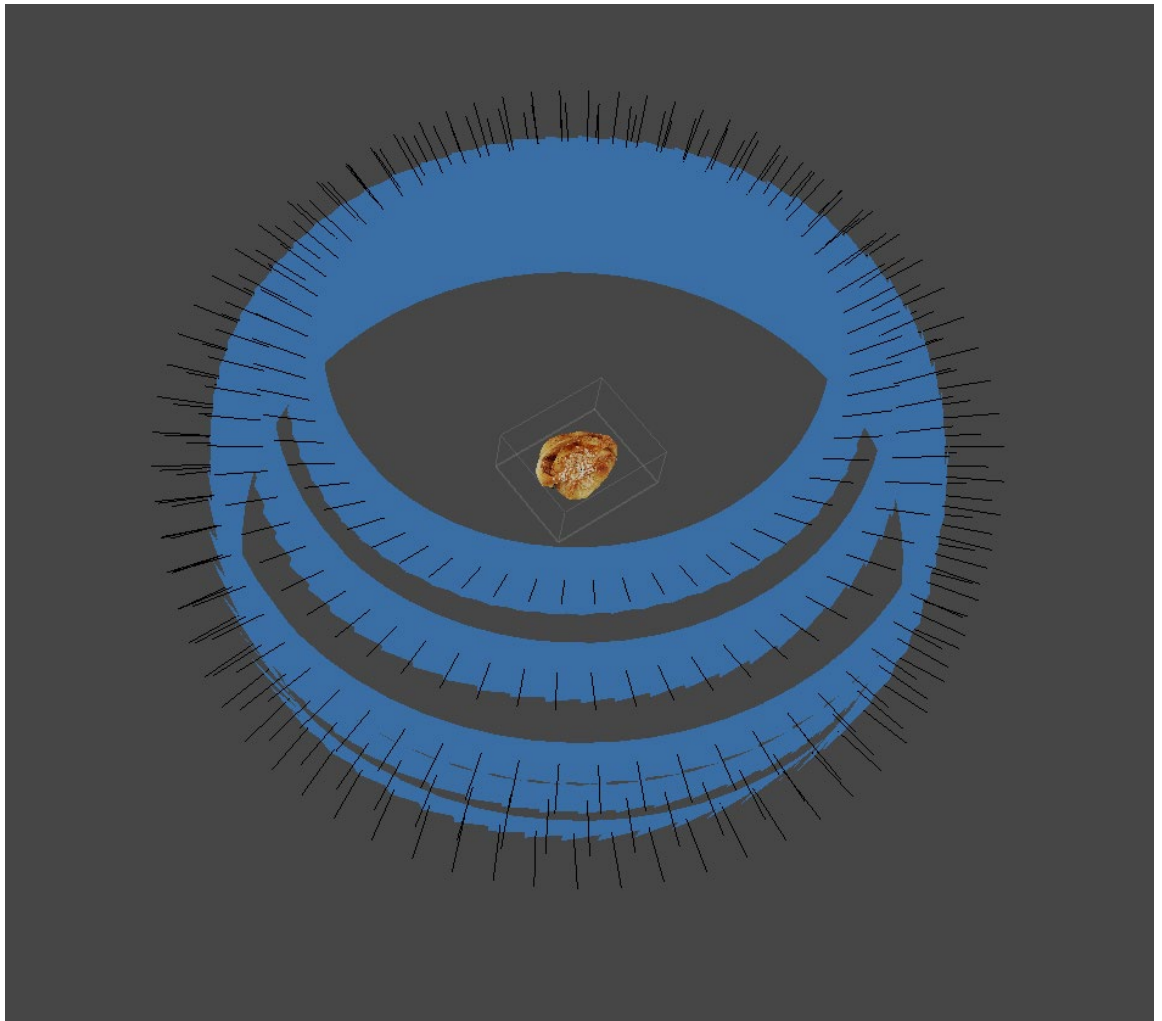
Taulukko 7. Sinihomejuustosarven tekniset tiedot.

Korvapuusti
Kameroita 320 kpl viidessä kerroksessa
Tiheäpistepilvi koostui 21 072 543 pisteestä
3D-mallissa oli 4 214 508 polygonia

Taulukko 8. Korvapuustin tekniset tiedot.



Kuva 31. Kuvassa sinihomejuustosarvi ja kameroiden sijainnit.



Kuva 32. Kuvassa korvapuusti ja kameroiden sijainnit.



Kuva 33. Lähikuva korvapuustin polygonimallista.



Kuva 34. Lähikuva sinihomejuustosarven polygonimallista.



Kuva 35. Blenderillä viimeistelty 3D-malli piirakasta.

8 Pohdinta

Kokonaisuutena olen lopputulokseen tyytyväinen. Opin paljon fotogrammetriasta ja laitteiston suunnitteleminen ja toteuttaminen haastoi osaamistani. Projektin alkuperäistä toteutusideaa ei kumminkaan saavutettu. Alkuperäisenä ideana oli toteuttaa mahdollisemman automatisoitu kokonaisuus, jossa käyttäjän tarvitsee vain asettaa kuvattava kohde ja antaa sille kuvaava nimi. Koska korkeampi automatisaatiotaso vaatisi ohjelmallisesti hallittavan objektiivin tai konenäkökirjaston, jolla pystytään tunnistamaan kuvattavan kohteen tarkkuustilanne ja automaattisesti siirtämään runkokehikko oikeaan sijaintiin.

Opin paljon additiivisesta valmistuksesta, koska opinnäytetyön aikana opettelini 3D-tulostamisen ja CAD-suunnittelun. Alussa tein paljon suunnitteluvirheitä ja virheet näkyivät niin, että tulostin useita versioita samoista osista ja muokkasin joitain osia käsin tulostuksen jälkeen. Osien toteuttaminen 3D-tulostomalla oli kuitenkin hyvä ratkaisu, koska muut valmistustavat olisivat olleet minulle työläämpiä.

Alkuperäiseen opinnäytetyökokonaisuuteen kuului myös jälleenrakennusprosessi sekä siitä syntyvien 3D-mallien optimisointi pelimoottoreille. Tämä olisi ollut mahdollista toteuttaa siten, että työpöytä tietokoneelle kehitettäisiin ohjelma, joka katsoo tiedostopalvelimelta jälleenrakentamistöitä ja ajaa sekä jälleenrakennusohjelmat ja optimisointiohjelman komentokehotetta käyttäen ja valvoen niiden suoriutumista. Tästä on hyvä jatkaa kohti seuraavaa fotogrammetriastudioversiota.

9 Yhteenveto

Lopputuotoksena toteutui puoliautomaticoitu fotogrammetriastudio, jolla saadaan luotettavasti digitalisoitua pieniä esineitä. Valokuvausteltan ja studiovalojen ansiosta Raspberry Pi High quality-kamera onnistuu ottamaan korkealaatuisia kuvia. Näistä kuvista Meshroomilla ja Metashapella ei ole ongelmia tuottaa tiheitä pistepilviä ja laskea niistä korkealaatuisia polygonmalleja sekä väritekstureja. Näitä malleja pystyy käyttämään sellaisenaan pelimoottoreissa, mutta korkean polygon määrän takia ne eivät välttämättä sovellu pelituotanto käyttöön, vaan vaativat polygonien määrän vähentämistä, joko käsin 3D-mallinnus ohjelmalla tai erillisellä ohjelmalla, joka osaa automaattisesti optimoida 3D-mallin pelituotanto käyttöön.

Pienellä budjetilla saatiin aikaiseksi laitteistollisesti toimiva fotogrammetriastudiokokonaisuus, mutta ohjelmistopuolelle jäi vielä kehitettävää. Tämänhetkisessä toteutuksessa käyttäjältä vaaditaan liikaa manuaalista väliintuloa. Ongelmia tuotti Raspberry Pi -kameran objektiivit, joissa ei ole automaattista kohdistamista, vaan kohdistaminen täytyi toteuttaa säätämällä kameran etäisyyttä kuvattavaan kohteeseen. Automaattisen kohdistamisen olisi voinut toteuttaa käyttämällä konenäkökirjastoa kuten OpenCV:tä.

10 Lähteet

- 1 Sergeev A, How will 3D production change in the future?, 80.lv, 26.6.2020, Viitattu: 12.11.2021, Saatavilla: <https://80.lv/articles/how-will-3d-production-change-in-the-future>
- 2 Kivolya N. What are 3D scanners used for, Professional 3D scanning solutions, 19.6.2019, Viitattu: 2.11.2021, Saatavilla: <https://www.artec3d.com/learning-center/what-are-3d-scanners-used-for>
- 3 Curless B, From Range Scans to 3D Models, ACM SIGGRAPH Computer Graphics, 4.11.1999, 33(4):38-41, Viitattu: 15.11.2021, DOI: 10.1145/345370.345399, S2CID: 442358
- 4 Bennett A, An Introduction of 3D-scanning and 3D-scanners, Julkaistu: 12.10.2019, Viitattu: 24.10.2021, Saatavilla: <https://www.revopoint3d.com/an-introduction-of-3d-scanning-and-3d-scanners/>
- 5 Büttgen B, Oggier T, Lehmann M, Kaufmann R, Lustenberger F, CCD/CMOS Lock-In Pixel for Range Imaging: Challenges, Limitations and State-of-the-Art, 2005, Viitattu: 23.11.2021, Saatavilla: <http://www.j-clavis.co.jp/pdf/h-3.pdf>
- 6 Synch T, Meadows P, Brent A, Azure Kinect DK depth camera, Docs.microsoft.com, 2020, Viitattu: 3.11.2021, Saatavilla: <https://docs.microsoft.com/en-us/azure/kinect-dk/depth-camera>
- 7 Lopez E, Felquiras T, Grunert C, Brueckner F, Mirko R, Seidel A, Marquardt A, Leyens C, Beyer E, Evaluation of 3D-printed parts by means of high-performance computer tomography, Journal of Laser Applications, 1.8.2018, Vol. 30, DOI: 10.2351/1.5040644
- 8 Kersten T, Lindstaedt M, Starpsta D, COMPARATIVE GEOMETRICAL ACCURACY INVESTIGATIONS OF HAND-HELD 3D SCANNING SYSTEMS – AN UPDATE, 30.05.2018, Vol. XLII-2, SPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, DOI: 10.5194/isprs-archives-XLII-2-487-2018

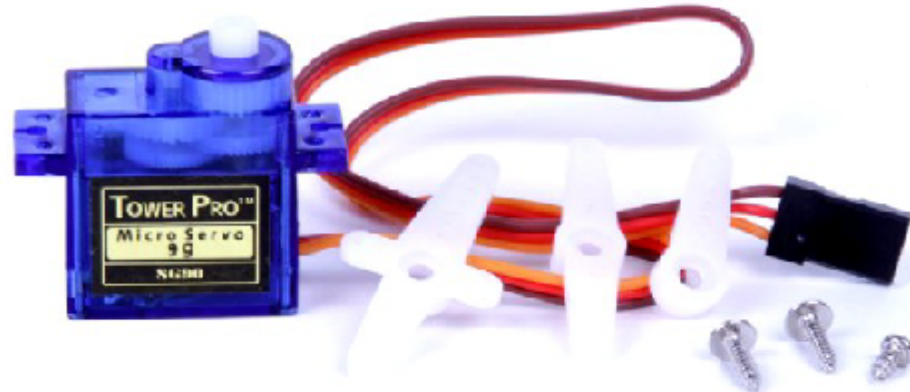
- 9 Mrovlje J, Vrančić D, Distance measuring based on stereoscopic pictures, 1-3.10.2008, Izola, Viitattu: 16.11.2021, Saatavilla: <http://dsc.ijs.si/files/papers/S101%20Mrovlje.pdf>
- 10 Photogrammetry pipeline, AliceVision, Viitattu: 20.11.2020, Saatavilla: <https://alice-vision.org/>
- 11 Sébastien L, Sébastien L, Cyril J, Photogrammetry Workflow, Unity Technologies - Rendering Research, Unity Technologies, Julkaistu: 23.6.2017, Viitattu: 25.10.2020, Saatavilla: https://unity3d.com/files/solutions/photogrammetry/Unity-Photogrammetry-Workflow_2017-07_v2.pdf
- 12 Brown K, Hamilton A. Photogrammetry and “Star Wars Battlefront”, DICE, GDC 2016, Viitattu: 20.11.2020, Saatavilla: <https://www.gdcvault.com/play/1023272/Photogrammetry-and-Star-Wars-Battlefront>
- 13 General Image Capture tips, Agisoft, 2020, Viitattu: 4.11.2020, Saatavilla: <https://agisoft.freshdesk.com/support/solutions/articles/31000149337>
- 14 Innmann M, Süßmuth J, Stamminger M, BRDF-Reconstruction in Photogrammetry Studio Setups, 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Julkaistu: 1-5.5.2020, DOI: 10.1109/WACV45572.2020.9093320.
- 15 Big Alice – Welcome to the wonderful world of Alice, 3dcopysystems, 2017, Viitattu 30.10.2021, Saatavilla: <https://3dcopysystems.com/big-alice/>
- 16 Jääskeläinen O, Kuusimurto T, Luottokortin kokoiset monitaiturit, Mikrobitti, 2016, Vol. 20 (8), S 14–27, ISSN 2343-1431
- 17 Heinisuo O, Järvinen O, Malkamäki M, Kehitysalustoja joka lähtöön, Mikrobitti, 2020, Vol. 60 (4), S 14–23, ISSN 2343-1431
- 18 Getting started with the Camera Module, 22.10.2021, <https://projects.raspberrypi.org/en/projects/getting-started-with-picamera>
- 19 Raspberry Pi Documentation – Configuration, 22.10.2021, <https://www.raspberrypi.com/documentation/computers/configuration.html>

- 20 Raspberry Pi, Raspberry Pi Press, The official Raspberry Pi Camera Guide, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, 26.5.2020, 130 S, ISBN 978-1-912047-52-9
- 21 Jones D, Picamera, 2017, Viitattu: 22.10.2021, Saatavilla: <https://picamera.readthedocs.io/en/release-1.13/>
- 22 rdagger, Raspberry Pi stepper motor tutorial, 3.6.2017, Viitattu: 23.10.2021, Saatavilla: <https://www.rototron.info/raspberry-pi-stepper-motor-tutorial/>
- 23 Nema motor grame size and what they mean..., ZikoDrive, Viitattu: 23.10.2021, Saatavilla: <https://www.zikodrive.com/ufags/nema-motor-frame-sizes-mean/>
- 24 Texas Instruments, DRV8825 Stepper Motor Controller IC datasheet, www.ti.com, 23.10.2021, <https://www.ti.com/lit/ds/symlink/drv8825.pdf>
- 25 17HS4401S Specification, Handson Technology, Viitattu: 24.10.2021, Saatavilla: <https://www.handsontec.com/dataspecs/17HS4401S.pdf>
- 26 Miller L, How to Control a Servo with Raspberry Pi, 16.7.2020, Learn Robotics, Viitattu: 27.10.2021, Saatavilla: <https://www.learnrobotics.org/blog/raspberry-pi-servo-motor>
- 27 Barnatt C, Raspberry Pi Servo Motor Control, 12.1.2021, Explainingcomputers, Viitattu: 27.10.2021, Saatavissa: https://www.explainingcomputers.com/pi_servos_video.html
- 28 Alasdair A, Scheller A, GPIO and the 40-pin Header, 9.8.2021, Viitattu: 7.11.2021, Saatavilla: <https://www.raspberrypi.com/documentation/computers/os.html#gpio-and-the-40-pin-header>
- 29 Memory requirements for processing operations, Agisoft, 2021, Viitattu: 23.11.2021, Saatavilla: <https://agisoft.freshdesk.com/support/solutions/articles/31000157329-memory-requirements-for-processing-operations>

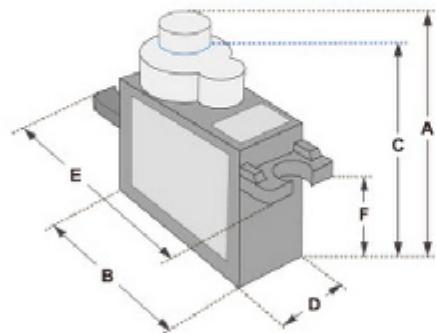
Litteet

SERVO MOTOR SG90

DATA SHEET



Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.



Dimensions & Specifications	
A (mm) :	32
B (mm) :	23
C (mm) :	28.5
D (mm) :	12
E (mm) :	32
F (mm) :	19.5
Speed (sec) :	0.1
Torque (kg-cm) :	2.5
Weight (g) :	14.7
Voltage :	4.8 - 6

Position 0° (1.5 ms pulse) is middle, 90° (~2ms pulse) is middle, is all the way to the right, -90° (~1ms pulse) is all the way to the left.

