Eetu Keränen

Stereokuvantamisen menetelmät

Insinööri (AMK)

Tieto- ja viestintätekniikka

Kevät 2022

KAMK • University of Applied Sciences

Tiivistelmä

Tekijä: Eetu Keränen

Työn nimi: Stereokuvantamisen menetelmät

Tutkintonimike: Insinööri (AMK), tieto- ja viestintätekniikka

Asiasanat: ohjelmistokehitys, stereokuvantaminen, pistepilvi, kamera, C++, OpenCV

Opinnäytetyö tehtiin Raute Oyj:n toimeksiantona. Työn tavoitteena oli toteuttaa stereokuvantamisjärjestelmä, joka tuottaa kameran alla liikkuvasta viilusta kuvavirtaa, syvyyskartan sekä laskee siitä pistepilven. Näitä voidaan hyödyntää viilun laadutuksessa. Työssä käytettiin Rauten normaalisti käytettävistä viivakameroista poiketen kolmea matriisikameraa. Matriisikameroilla toteutuksen kokonaishinta on edullisempi ja kuvauslaite on paljon matalampi kooltaan. Pienemmän kuvausjärjestelmän asentaminen on helpompaa, kuin aikaisemman.

Työn vaatimuksena oli tukea Windows- ja Linux-käyttöjärjestelmiä. Pääkohde oli kuitenkin Linux-käyttöjärjestelmä. Ohjelmisto toteutettiin C++-ohjelmointikielellä käyttäen hyväksi OpenCV- ja PCL-kirjastoja. Työ suunniteltiin toteutettavaksi kahdessa vaiheessa. Ensimmäisessä vaiheessa keskityttiin stereokuvan muodostavan ohjelmiston kehittämiseen paikallaan olevista kohteista käyttäen hyväksi kuluttajamarkkinoilta tuttuja web-kameroita. Toisessa vaiheessa kehitystyötä tehtiin liikkuville kappaleille, teollisuuden konenäkökameroita ja pulssitettua valaisua hyödyntävällä testilinjastolla.

Työn tuloksena saatiin aikaan toimiva ohjelmisto, joka muodostaa syvyyskartasta pistepilven. Jotta järjestelmää voitaisiin tulevaisuudessa käyttää tuotantolinjastolla, tulisi laskentaa pystyä nopeuttamaan merkittävästi. Ohjelmisto vastasi vaatimuksiin muuten, mutta nopeudessa on kehitettävää.

KAMK • University of Applied Sciences

Abstract

Author(s): Eetu Keränen

Title of the Publication: Stereo Imaging

Degree Title: Bachelor of Engineering, Information and Communication Technology

Keywords: Software documentation, stereo vision, point cloud, camera, C++, OpenCV

This thesis was commissioned by Raute Oyj. The goal of this thesis was to implement a stereo imaging system that produces an image stream, depth map and point cloud from the moving veneer. These can be used for the grading of the veneer. Normally Raute uses line cameras for the imaging, but this thesis uses three matrix cameras. Matrix cameras will make the product more affordable and smaller. A smaller imaging system is also easier to install than before.

The prerequisite was to support Windows and Linux operating systems. The main target was Linux. The software was implemented in C++ programming language using OpenCV and PCL libraries. The thesis was designed to be implemented in two parts. The first part focused on making software that produces a stereo image using two consumer grade web cameras. In the second part, development was done on a small test line with industry grade machine vision cameras and pulsed lighting.

The result of the thesis was software which produces a depth map and point cloud. In order to use the system in production it needs more optimizations. The software fulfilled the requirements other than the speed.

Symboliluettelo

OpenCV	Avoimen lähdekoodin konenäkökirjasto.
PCL ´	Avoimen lähdekoodin Point Cloud Library -kirjasto pistepilvien käsittelyyn.
C++	Ohjelmointikieli.
Pistepilvi	Kokoelma 3-ulotteisia pisteitä.
CUDA	Nvidian ohjelmointirajapinta näytönohjain ohjelmointiin
Kernel	Näytönohjaimella suoritettava ohjelma.
Basler	Kameravalmistaja.
BSD	BSD-lisenssi on vapaa ohjelmistolisenssi. Sallii käytön ja muokkaamisen myös kaupallisissa tuotteissa.

1 Johdanto

Työn aihe tulee Raute Oyj:n Kajaanin yksiköltä. Työssä tutkitaan, kuinka stereokuvantamismenetelmät sopivat viilun muodon laskemiseen reaaliajassa tarvittavalla tarkkuudella. Stereokuvantamista on jo aiemmin hyödynnetty Rautella, mutta suurimmaksi osaksi vain viivakameroilla. Tässä työssä kuitenkin tullaan käyttämään aivan normaaleja matriisikameroita. Matriisikameroilla voidaan potentiaalisesti saavuttaa huomattavasti halvempi kamerajärjestelmä.

Järjestelmän etuna on myös se, että siitä tulee kooltaan paljon matalampi kuin viivakamera toteutuksesta. Pienempi koko mahdollistaa järjestelmän lähettämisen helposti postissa. Koska järjestelmä ei vaadi niin paljoa kalibrointia paikan päällä, voidaan sen asentaminen jättää ei-asiantuntijalle.

Työssä toteutetaan stereokuvantamisjärjestelmä, joka toimii 3 kameralla. Kamerat kuvaavat niiden alta liikkuvaa viilua, josta sitten lasketaan syvyyskartta. Syvyyskartta voidaan jatkojalostaa pistepilveksi. Syvyyskarttaa voidaan käyttää suoraan viilussa olevien reikien tunnistamiseen ja laaduttamiseen. Pistepilven avulla voitaisiin laskea miltä aaltoileva viilu näyttäisi, jos se olisi täysin suorassa.

Työssä päästään tutustumaan konenäköratkaisuihin ja kameroiden toimintaan. Työssä lähdetään ensin tutkimaan ihmissilmän toiminnan kautta, kuinka stereokuvantaminen toimii. Huomioitava on myös valaistuksen ja kuvan liipaisemisen vaikutus kuvan laatuun. Kameroiden tuottamassa kuvissa on vääristymää, joka täytyy korjata kalibroimalla kamerat. Teoriaosuuden jälkeen työssä esitellään kaksi erilaista kamera-asetelmaa. Lopuksi käydään läpi työn suunnittelu, toteutus ja lopputulokset.

2 Kuvantaminen

Kuvan otosta syvyyskartan laskemiseen tapahtuu monta eri vaihetta. Seuraavissa kappaleissa esitellään olennaisimmat vaiheet tästä prosessista.

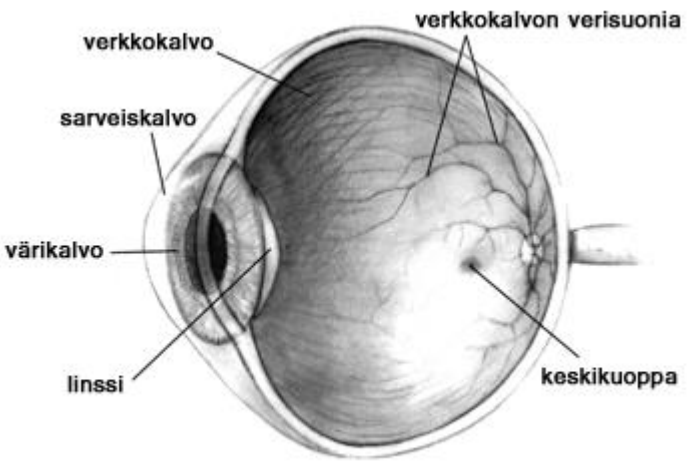
2.1 Kuvanmuodostuminen

Ihmissilmä kuvaa uskomattomalla tarkkuudella maailmaa. Kameroita on jo pitkään yritetty rakentaa ihmissilmän kaltaisiksi. Stereokuvantaminen on hyvä esimerkki onnistuneesta ihmissilmän matkimisesta.

2.1.1 Ihmissilmä

Kuvantamismenetelmän ymmärtämiseksi kannattaa tutustua ihmissilmän toimintaan. Näiden tietojen pohjalta on mahdollista kehittää käsitteellisiä malleja, joita voidaan hyödyntää kuvakäsittelyssä ja kuvien tarkkuuden arvioinnissa. [1.]

Kuvassa 1 on piirros ihmissilmän osista. Silmän etuosaa suojaa läpinäkyvä sarveiskalvo. Muuta silmää ympäröi kovakalvo, jonka päällä taas sijaitsee silmän keskimmäinen kerros suonikalvo. Viimein suonikalvon päällä on verkkokalvo. Verkkokalvo koostuu kahdenlaisista aistisoluista, joiden nimet ovat sauvasolut ja tappisolut. [1.]



Kuva 1. Ihmissilmän rakenne [2]

Valo kantautuu silmään, joko kimmonneena objektista tai sen tuottamana. Valo kohdistuu verkkokalvolle linssin kautta silmän lihasten säätämänä. Lihakset säätävät silmän linssin muotoa, jotta kohdistus olisi tarkka lähellä tai kaukana olevalle asialle. [1.]

Silmän verkkokalvolla olevat sauvasolut ovat pitkiä hoikkia reseptoreita ja tappisolut ovat taas lyhyempiä ja paksumpia rakenteeltaan. Sauvasolut ovat paljon herkempiä valolle kuin tappisolut ja ovat vastuussa hämäränäöstä. Tappisolut ovat herkempiä valon aallonpituuksille ja ovat merkittävässä osassa värien havaitsemisessa. Tappisolut jakautuvat vielä kolmeen eri ryhmään värien havaitsemisessa: S-; M- ja L-tyyppiin. Tyyppi määrittelee, minkä aallonpituuden värisävylle tappisolu on herkkä. Ihmissilmässä on noin 6,5 miljoonaa tappisolua ja 100 miljoonaa sauvasolua. [1.]

Lähellä näköhermoa löytyy keskikuoppa, joka sisältää suurimman keskittymän tappisoluja ja sen takia näköaisti on tässä kohtaa terävin. Silmän takaosasta lähtee ulos näköhermo, joka yhdistää silmän aivoihin. Näköhermon ympärillä ei ole yhtään fotoreseptoria ja tämä aiheuttaa ihmiselle niin sanotun sokean pisteen. [1.]

2.1.2 Kamera ja linssi

Kamera on kuvien tallentamiseen tarkoitettu laite. Nimi kamera on peräisin latinankielisestä sanasta camera obscura – pimeä huone. Camera obscura on ilmiö, jossa valo kulkiessaan hämärään tilaan muodostaa sinne kuvan ulkopuolisesta maisemasta ylösalaisin. Ilmiöön perustuvat kaikki kamerat ja silmämme. Tämä optinen ilmiö tunnettiin jo ennen ajanlaskumme alkua. [3.]

Kamerassa on oltava linssi, joka kerää objektin pinnasta heijastuneen valon. Linssi heijastaa valon linssin takana olevalle valoherkälle alueelle, joka on yleensä CCD- tai CMOS-kuvakenno. Kuva muodostuu kennolle ylösalaisin. [4.]

Linssin polttoväli on mitta sille, miten paljon se taittaa valoa. Polttoväli kertoo, kuinka pitkä matka linssistä on polttopisteeseen. Kameran kuvakennon on oltava polttopisteen takana, tätä kennon sijaintia kutsutaan kuvatasoksi. Polttoväliä mitataan millimetreissä. Kameran näkökentän laajuus kertoo, mikä on kuvan suhde verrattuna kuvattavan kohteen kokoon. [4.]

Gaussin kuvausyhtälö (1) kertoo, miten linssin esineestä muodostaman kuvan etäisyys linssistä riippuu esineen etäisyydestä, sekä linssin polttovälistä.

$$\frac{1}{f} = \frac{1}{a} + \frac{1}{a'}, \tag{1}$$

missä f on polttoväli, a on kuvattavan objektin etäisyys linssiin ja a’ on kuvan etäisyys linssistä kennolle.

2.2 Kuvan liipaiseminen

Liikkuvan kuvan otossa on tärkeää käyttää yleistä suljinta (global shutter). Yleinen suljin mahdollistaa terävien kuvien oton nopeasti liikkuvista objekteista, koska kaikki pikselit valotetaan yhtä aikaa. liukuva suljin (rolling shutter) aiheuttaa virhettä liikkuvassa kuvassa, koska valotus ei tapahdu yhtä aikaa jokaiselle pikselille. Tämä aiheuttaa virheen, jossa kuvan yläosassa olevat pikselit ovat otettu aikaisemmin kuin alaosassa. [4, s245.]

Liikkuvissa kuvissa pitää käyttää niin pientä valotusaikaa kuin mahdollista, ettei tapahdu vauhtiviiva efektiä. Valotusajan pitää olla tarpeeksi pieni, että kuvattava objekti liikkuu vähemmän kuin yhden pikselin verran valotuksen aikana. [4, s245.]

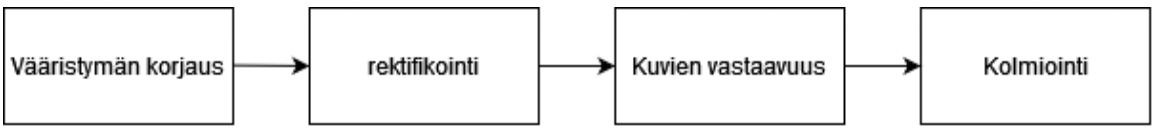
2.3 Valaisu

Valotusajalla tarkoitetaan, kuinka kauan kameran sensori saa kerätä valoa. Pieni valotusaika vähentää kuvan kohinaa. Koska liikkuva kuva halutaan jähmettää paikalleen, tarvitaan pientä valotusaikaa. Tämän takia täytyy käyttää runsaasti valoa, jotta kuvista saadaan tarvittavan kirkkaita. Tyypillisesti valaisuun käytetään nykyisin riittävän valaisutehon aikaansaamiseksi puolijohteilla, kuten ledeillä tai laserilla toteutettuja valaisimia. Niiden etuna ovat monipuoliset liipaisu- ja pulssitusmahdollisuudet nopean syttymisen ansiosta [4, s250.]

2.4 Stereokuvantaminen

Stereokuvantaminen pohjautuu ihmisen kykyyn muodostaa syvyysvaikutelma käyttäen kahta silmää. Tässä niin sanotussa binokulaarisessa visiossa ihmisen aivot saavat signaalit molemmilta silmiltä ja pystyvät pienen eroavaisuuden avulla arvioimaan etäisyyden näkökentässä oleviin asioihin. Stereokuvantamisessa onkin kyse samasta ilmiöstä, mutta aivojen tilalla on tietokone ja silmät ovat vaihtuneet kameroiksi.

Stereokuvantamisessa on käytännössä 4 askelta, kun käytetään 2 kameraa. Kuvassa 2 vaiheet on esitetty lohkokaaviona. Aluksi poistetaan linssin vääristymä eli vääristymän korjaus. Seuraavaksi kuva rektifioidaan eli asetetaan pikselirivit vastaamaan toisiaan kuvien välille. Kolmannessa vaiheessa etsitään kuvista yhtenäisiä piirteitä. Tämä vaihe tuottaa eroavuuskartan, joka kertoo, kuinka paljon kuvassa olevat piirteet eroavat toisistaan x-suunnassa kameroiden välillä. Neljännessä vaiheessa voidaan kuva kolmioida eli laskea, kuinka paljon etäisyyttä kappaleeseen on kamerasta.



Kuva 2. Lohkokaavio stereokuvantamisen päävaiheista

2.4.1 Vääristymän korjaus

Kamerat aiheuttavat kuviin erilaisia vääristymiä. Vääristymien korjaaminen on tärkeää, jotta myöhempien vaiheiden matematiikka pitää paikkaansa. Vääristymät pystytään korjaamaan kalibroinnin antamien korjausarvojen avulla. Kuvassa 3 on esimerkki tynnyri- ja neulatyynyvääristymästä.

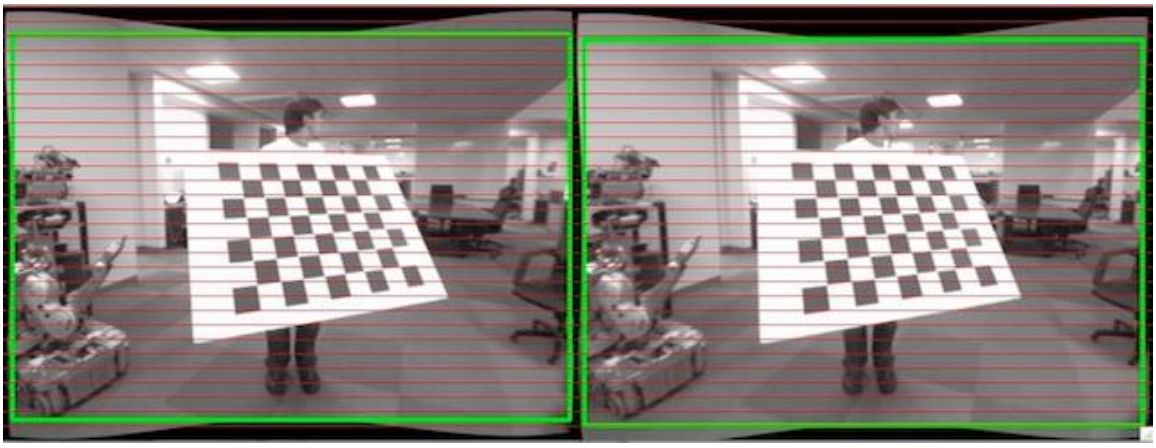
Kuva 3. Erilaisia kameravääristymiä [5]

2.4.2 Stereokalibrointi

Stereokalibrointia varten tarvitaan kuvia, joissa shakkilaudasta on otettu kuvia molemmilla kameroilla. Kuvista lasketaan shakkilaudan sijainti ja rotaatio, josta saadaan selville paljonko, kameroiden välillä on välimatkaa ja missä kulmassa ne ovat toisiinsa nähden. Stereokalibrointia tarvitaan rektifioinnin laskemiseen.

2.4.3 Rektifiointi

Kun tiedossa on kuvien stereokalibraatio, voidaan kuvat rektifioida. Tässä vaiheessa kuvat lasketaan samaan tasolle. Seurauksena kuvien epipolaariset janat asettuvat rinnakkain ja näin yksinkertaistaan seuraavan vaiheen tiheää stereovastaavuusongelmaa [6]. Kuvassa 4 voidaan havainnollistusviivojen avulla huomata, että kuvissa pikselit ovat samassa tasossa.



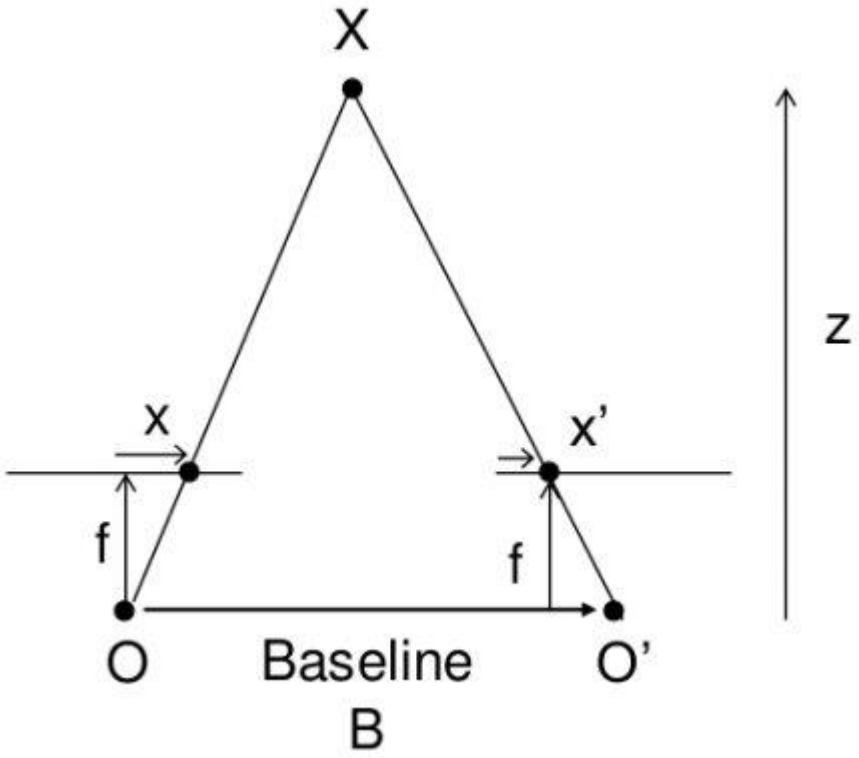
Kuva 4. Rektifioitu kuva [7]

2.4.4 Kuvien vastaavuus

Rektifioinnin seurauksena voidaan nyt laskea kuvien pisteet, jotka ovat näkyvissä molemmissa kuvissa ja laskea niille eroavaisuus x-suunnassa. Kuvien pikselit ovat samalla tasolla y-suunnassa rektifioinnin ansiosta, joten pikselien erotusta laskettaessa tarvitsee vain vertailla pikseleitä yhdeltä riviltä.

2.4.5 Kolmiointi

Tässä vaiheessa voidaan laskea etäisyys kuvasta. Kuvassa 5 on hahmotettu, kuinka kaksi kameraa, O ja O’ näkevät pisteen X eri kohdassa. Parametri B on välimatka kameroiden välillä. Mikäli halutaan tarkasti laskea etäisyys lähellä oleviin kohteisiin, on kameroiden välimatkan B oltava pieni.



Kuva 5. Kuva kolmiointiprosessista [8]

Alla on esitetty kolmiointiin liittyvät matemaattiset kaavat

$$\frac{B-(x-x')}{Z-f} = \frac{B}{Z} \Rightarrow Z = \frac{fB}{x-x'}, \quad (2)$$

$$d = x - x' \text{ ja} \quad (3)$$

$$Z = \frac{fB}{d} \quad , \quad (4)$$

missä f tarkoittaa polttoväliä, Z on välimatka kamerasta objektiin, T on kameroiden välimatka, d on kameroiden eroavaisuus, joka laskettiin vaiheessa 3.

Näin ollen etäisyys voidaan laskea kaavalla 3 eli jaetaan polttoväli kertaa kameroiden välimatka kameroiden eroavaisuudella. Välttääkseen jakolaskua voidaan myös kertoa kameroiden eroavaisuus jakolaskun ylemmän termin käänteisluvulla.

Koska syvyys on käänteisesti verrannollinen eroavaisuuteen, on niiden suhde epälineaarinen. Jos eroavaisuus on lähellä nollaa, pienet muutokset aiheuttavat suuria syvyyseroja. Kun eroavaisuus

on suuri, pienet syvyyserot eivät aiheuta suuria muutoksia syvyyteen. Seurauksena stereonäkösysteemit näkevät syvyyden muutokset paremmin lähellä kameraa olevista objekteista [9, s301.]

2.5 Kamerakalibrointi

Kamerat aiheuttavat vääristymiä kuviin. Vääristymä voidaan kuitenkin korjata esimerkiksi shakkilautakalibroinnilla. Tässä menetelmässä otetaan lukuisia kuvia shakkilaudasta eri asennoissa ja eri etäisyydellä kamerasta. Koska shakkilaudan ruutujen koko tiedetään, voidaan sen avulla laskea pois vääristymät kuvista. Kuvassa 6 on web-kameralla kuvattu shakkilauta.

Kuva 6. Web-kameralla otettu kuva shakkilaudasta

Opinnäytetyössä hyviin kalibrointituloksiin päästiin noin 100 kuvan avulla. Vääristymän korjaukseen käytettiin OpenCV-kirjastoa. Yli 100 kalibrointikuvan ottaminen kasvatti laskenta-aikaa merkittävästi ilman huomattavaa parannusta kalibroinnin laatuun.

Menetelmässä kamera ottaa kuvia jatkuvasti ja shakkilautaa liikutetaan kameran alla. Tämä menetelmä tuottaa paljon kuvia, joissa shakkilautaa ei voida tunnistaa kuvasta. Tämän takia kuvat

jaettiin hyvien ja huonojen kuvien kansioihin ohjelmallisesti. Tämä helpottaa jatkokäsittelyä, koska kuvat, joista shakkilautaa ei löydy, on paljon hitaampi tunnistaa kuin kuvat, joista se löytyy.

Kameran ottamat shakkilautakuvat arvottiin satunnaiseen järjestykseen, jotta valintaprosessissa saataisiin useista eri kulmista ja etäisyyksistä kuvia. On hyvin tärkeää, että kuvat ovat eri kulmista, koska pelkät kameran suuntaiset kuvat aiheuttavat kalibrointiin virhettä. Näistä kuvista tunnistettiin shakkilaudan ruutujen koordinaatit käyttäen OpenCV:tä.

Tunnistettujen shakkilautojen ruutujen koordinaatit järjestettiin vielä paremmuusjärjestykseen ja parikymmentä huonointa tiputettiin pois. Huonoimmissa tunnistuksissa shakkilaudan ruutujen koordinaatit menevät hieman ohi. Järjestäminen auttoi hieman parantamaan kalibrointitulosta.

Kalibrointi on kamerakohtainen ja kun se on kerran tehty, voidaan korjausarvoilla korjata kameran ottamat kuvat. Uudelleenheijastusvirhe kertoo, kuinka tarkka kalibroinnista tuli. Uudelleenheijastusvirhe lasketaan kalibroinnin korjausarvoista.

2.6 Työkalut

Ohjelmistokehityksessä kirjastot ovat valikoima aliohjelmia ja luokkia. Opinnäytetyössä Käytetyt työkalut ja sovellukset on esitelty seuraavissa luvuissa.

2.6.1 OpenCV

OpenCV (Open Source Computer Vision Library) on konenäköön ja koneoppimiseen erikoistunut kirjasto. Se on tehty nopeuttamaan kaupallisten konenäkösovellusten kehittämistä. OpenCV on tarjolla BSD-lisenssillä, joten se on ilmainen kaupalliseen ja tutkimuskäyttöön. Lähdekoodi on avointa eli sitä pääsee helposti lukemaan ja muokkaamaan itse. [10.]

OpenCV tarjoaa yli 2500 algoritmia konenäköön ja koneoppimiseen. Kirjasto sisältää kattavan valikoiman perinteisiä ja moderneja algoritmeja. Näitä voidaan käyttää esimerkiksi stereokuvantamiseen, esineiden tunnistukseen, kasvontunnistukseen, pistepilvien luomiseen sekä kuvien yhdistämiseen. [10.]

Tunnetut suuret yritykset kuten Google, Microsoft, Intel, Sony ja IBM käyttävät OpenCV-kirjastoa moniin eri tarkoituksiin. Esimerkiksi karttaohjelmien katunäkymät on rakennettu liittämällä

useita kuvia yhteen OpenCV:n avulla. Myös tuotteiden etikettien tarkistus tehtaissa ympäri maailman tapahtuu saman koodikirjaston mahdollistamana. OpenCV-yhteisöön kuulu yli 47000 ihmistä ja latauksien arvioitu määrä on yli 18 miljoona. [10.]

OpenCV tukee alustoina Windows-, Linux-, Android- ja Mac-käyttöjärjestelmiä. OpenCV:llä on rajapinnat C++, Python-, Java- ja MATLAB-ohjelmointikielille. OpenCV on kirjoitettu C++-ohjelmointikielellä. Useille algoritmeille löytyy myös näytönohjaimella pyörivä versio. Useissa kuvankäsittelyalgoritmeissa näytönohjainversio voi olla monta kertaa nopeampi. [11.]

2.6.2 PCL

PCL (Point Cloud Library) on avoimen lähdekoodin kirjasto, joka on tarkoitettu 2D/3D-kuvien ja pistepilvien prosessointiin. PCL on julkaistu BSD-lisenssillä, joten se on ilmainen kaupalliseen ja tutkimuskäyttöön. PCL-algoritmeihin kuuluvat mm. rekisteröinti, pistepilven suodatus, avainkohtien tunnistus, datastruktuureja pistepilville ja visualisointi. Näitä voidaan käyttää hyväksi esimerkiksi robotiikassa. PCL toimii Windows-, Linux-, Android- ja macOS-käyttöjärjestelmillä. PCL on kirjoitettu C++-ohjelmointikielellä. Opinnäytetyössä PCL on käytetty pistepilvien rekisteröintiin ja visualisointiin. [12.]

2.6.3 Pylon Viewer

Pylon Viewer on Baslerin kehittämä sovellus kameroiden säätämiseen. Pylon Viewer sisältää työkaluja kuvanottamiseen, fokuksen säätämiseen ja kaistanleveyden työkalun. Pylon Viewerissä voidaan säätää etukäteen kameroiden kuvanottotaajuudet, valotusajat ja kuvaformaatit. Pylon Viewerissä voidaan myös luoda .pfs-päätteisiä konfigurointitiedostoja kameroiden asetuksille, jotka voidaan myöhemmin ladata Pylon SDK:n kautta kameroita käytettäessä.

3 Kamera-asetelmat

Kehityksen aikana käytettiin kahta erilaista kamera-asetelmaa tutkimuksen edetessä. Aluksi lähdettiin kehittämään yleistä ohjelmistorakennetta web-kameroilla. Tätä kutsutaan nimellä linja 1. Tämän jälkeen siirryttiin seuraavalle linja 2:lle. Linjalla 2 käytettiin parempia konenäköön tarkoitettuja kameroita ja valaistusta.

3.1 Kamerat ja niiden sijoitus

Kameroiden sijoituksella on stereokuvantamisessa paljon merkitystä. Kamerat sijoittaminen vaikuttaa siihen, minkälainen niiden lomittuva kuvausalue on.

3.1.1 Linja 1

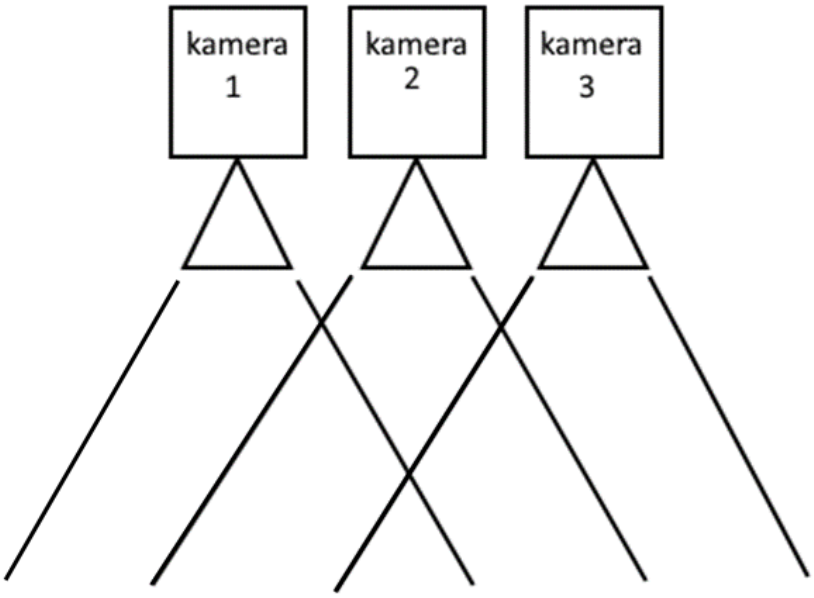
Kehityksen aikana testattiin kahta erilaista kamera-asetelmaa. Kuvassa 7 on kamera-asetelma. Ensimmäinen asetelma koostui kahdesta yhteen teipatusta Logitechin web-kamerasta, joiden hinta oli luokkaa 30 € kappaleelta. Tämä huonolaatuinen kamera-asetelma oli hyvä ensimmäiseen kehitysvaiheeseen. Tätä asetelmaa testattiin kolmella eri kameroiden välillä: 10 cm, 6 cm ja 3,2 cm. Parhaat mittatulokset 30–60 cm etäisyydellä oleville testikappaleille tuli 3,2 cm välimatkalla. Kamerat on teipattu tiukasti yhteen, koska jos kameroiden välimatka tai kulma muuttuu, kalibrointi menee pilalle.

Kuva 7. Kuva kahdesta yhteen teipatusta web-kamerasta

Tämä asetelma oli hyvä ensimmäisille testikoodeille, joilla harjoiteltiin ja tutkittiin OpenCV:tä ja stereonäköä. Suurin osa kuvista onkin huoneen nurkasta olevista arkisista esineistä ja materiaaleista, kuten pullot, kirjat, säästöpossut. Näistä kuvistakin opittiin jo muutamista ongelmista, kuten tekstuurit, jotka vaihtavat radikaalisesti väriä tai kiiltoa.

3.1.2 Linja 2

Seuraava versio kamera-asetelmasta oli pieni linjasto, joka oli kooltaan 70 cm x 140 cm. Kuvassa 8 on esitetty kolmen kameran asettelu, jota käytettiin linjastolla 2. Linjastoon pystyy kiinnittämään pienehköjä erilaisia viiluja, joita linjasto sitten automaattisesti liikuttaa. Linjaston liikerata alkuasennosta loppuun on noin 50–70 cm. Linjastossa on kolme vierekkäistä kameraa, jotka ottavat kuvia pulssitettuna kappaleen liikkuessa kameroiden alla. Kun linjasto pääse loppuasentoon, palaa kuvattava kappale takaisin alkuasentoon ilman, että kamera ottaa niistä kuvia.



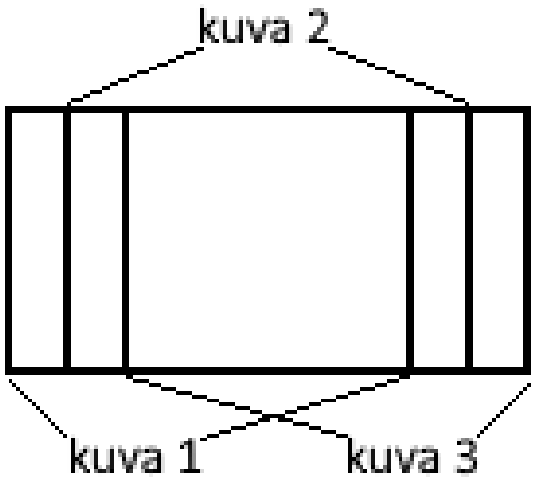
Kuva 8. kuva kamera-asetelmasta linjastosta 2.

Linjaston liikkuessa 0.2 m/s kuvattava objekti kerkesi liikkua kuvissa noin 5 cm. Linjasto ottaa 14 * 3 * 3 kuvaa per yhden kerran liikuttaessaan viilun alusta loppuun. Linjaston valaisee kolme suorakaiteen muotoista led-valaisinta. Valaisimet ovat asetettu yksi suoran viilun päälle ja kaksi muuta 45 asteen kulmaan sivuille. Valot ovat päällä yksi kerrallaan, joten joka valaisulta tulee oma kuva. Tätä voidaan hyödyntää myöhemmin kuvankäsittelyssä.

Verrattuna ensimmäiseen asetelmaan toinen linjasto on jo suuri harppaus kohti lopullista tuotetta. Tällä linjastolla on paljon helpompi tuottaa testikuvia viilujen liikkumisen, paremman valaisun ja realistisemman ympäristön ansiosta.

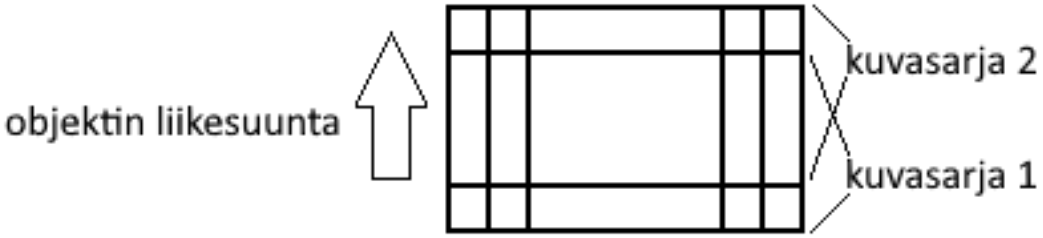
3.1.3 Objektien liikkuminen linjastolla 2

Stereokuvantamiseen tarvitaan kuvia, joissa on lomitusta. Linjastolla 2 syntyneisiin kuviin tuli noin 90 %:n lomitus. Kuva 9 havainnollistaa, kuinka kuvien näkökentät menevät päällekkäin. Syvyyskartta laskettiin kuvien 1 ja kuva 2 välille, sekä kuvien 2 ja 3 välille.



Kuva 9. Kuva kameroiden näkökenttien lomituksesta.

Kun kuvan objektit liikkuvat, syntyy lomittaisia kuvasarjoja kameran tämänhetkisten kuvien ja edellisten kuvien välillä, kuten kuvassa 10. Tällöin voimme käyttää esimerkiksi kameran 1 uusintakuvaa ja edellistä kuvaa stereokuvantamiseen, koska ne lomittuvat.



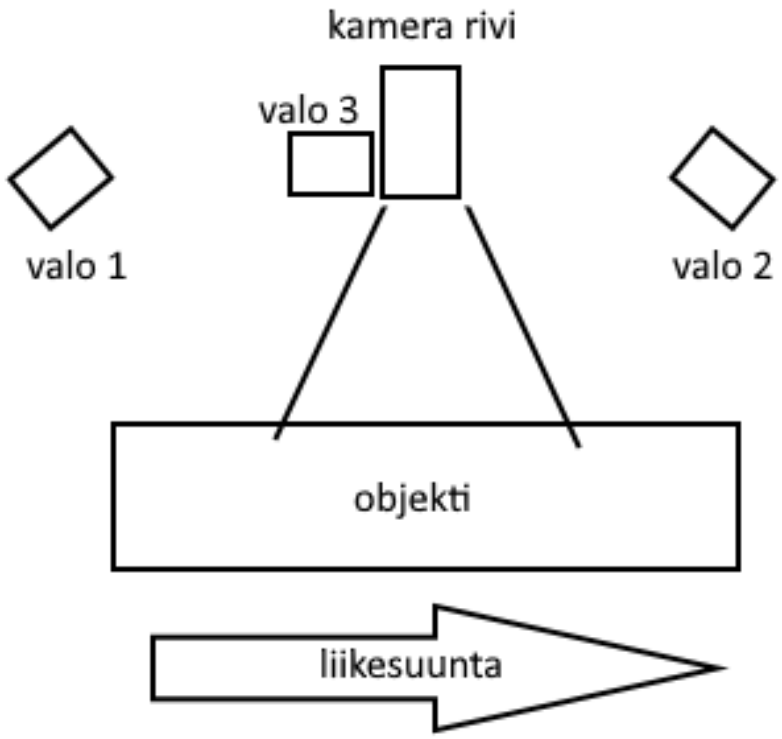
Kuva 10. Kuva kameroiden näkökenttien kuvasarjojen lomituksesta, kun objekti liikkuu.

Tämä menetelmä on myös edullinen, jos halutaan peittää suuri alue vähällä määrällä kameroita. Kamerat voidaan sijoittaa siten, että ne ovat kaukana toisista ja niillä ei ole suurta näkökentän päällekkäisyyttä. Kunhan kuvia otetaan vain tarpeeksi nopealla tahdilla, saadaan kuitenkin y-akselilla hyvät päällekkäisyydet uusimman ja edellisen kuvan kanssa.

Koska kuvaava kamera oli sama tässä tilanteessa, kalibrointi on täysin identtinen kuvien välillä. Kun kuvat tulevat kahdelta kameralta voi kalibroinnin tarkkuudessa olla pieniä eroavaisuuksia. Toinen etu on myös, että saamme 2 eri näkemystä syvyyskartasta. Nämä pystyttäisiin esimerkiksi keskiarvoistamaan tuloksen parantamiseksi.

3.2 Valaistus

Linjastolla liikkuvat objektit valaistaan kolmesta eri suunnasta suorakaiteen muotoisten led-lamppujen avulla. Kuvassa 11 on esitelty valojen sijoitus. Valot ovat päällä vuorotellen pulssitettuna, jotta saadaan eri valaisuprofiileilla kuvia samasta objektista samassa kohtaa. Näitä kuvia, joissa objekti on eri tavalla valaistu, voidaan käyttää hyväksi kiillon määrittelyssä. Valot 1 ja 2 ovat sijoitettu kaksi 45 asteen kulmaan kameroista ja valo 3 suoraan kameroiden tasolle.



Kuva 11. Valaistus

4 Työn toteutus

Toteutuksessa kerrotaan aluksi vaatimukset. Näiden pohjalta tehtiin suunnitelma, jota lähdettiin toteuttamaan. Työn toteutus suunniteltiin kaksivaiheiseksi. Kehitystyö tultaisiin aloittamaan web-kameroilla ja seuraavassa vaiheessa siirryttäisiin hyödyntämään vielä rakenteilla olevaa linja 2:sta.

4.1 Vaatimusmäärittely

Ohjelman on pystyttävä tuottamaan kuvavirtaa reaaliajassa kaksi metriä sekunnissa millin natiiviresoluutiolla X määrällä kameroita. Ohjelmointikielenä työn toteutuksena tuli käyttää C++-ohjelmointikieltä, joka on laajasti käytössä työn toimeksiantajalla. Työssä kehitettävä koodi tuli olla suoritettavissa sekä Windows- että Linux-käyttöjärjestelmäympäristössä. Lopullisena tavoitteena suoritusympäristöksi oli Linux-käyttöjärjestelmällä varustettu UP Squared yhden piirilevyn tietokone.

4.2 Suunnittelu ja toteutus

Vaatimusmäärittelyn myötä työssä päädyttiin käyttämään seuraavia kirjastoja; OpenCV:tä konenäköalgoritmeihin ja Point Cloud Librarya 3D-pistepilvien muodostamiseen. Aluksi suunniteltiin OpenCV-kirjaston testaamista kahdella web-kameralla. Näin päästään alkuun toteuttamaan ohjelmiston perusrakennetta. Sen jälkeen, kun kahdella webbikameralla alkaa tulla stereokuvaa, siirrytään käyttämään kolmea Baslerin kameraa. Baslerin kameroissa kuvaresoluutio ja laatu nousevat roimasti. Kuvaresoluution kasvaminen tarkoittaa tietenkin laskuajan nousua, mutta laatua tarvitaan hyvään lopputulokseen. Baslerin kameroihin valmistajalla on tarjolla Pylon SDK -ohjelmointirajapinta.

Pistepilvien laskenta tuli mukaan varsin dynaamisesti sen jälkeen, kun olimme saaneet laskettua syvyyskarttoja. Pistepilvien kanssa mietittiin, olisiko se hyvä tapa yhdistellä useiden kameroiden kuvaa yhdeksi kuvaksi. Pistepilven kautta tulevassa kuvayhdistelyssä on hyvänä puolena se, että olisi voitu testata, voisiko kuvattavaa aaltoilevaa viilua samalla suoristaa matemaattisesti.

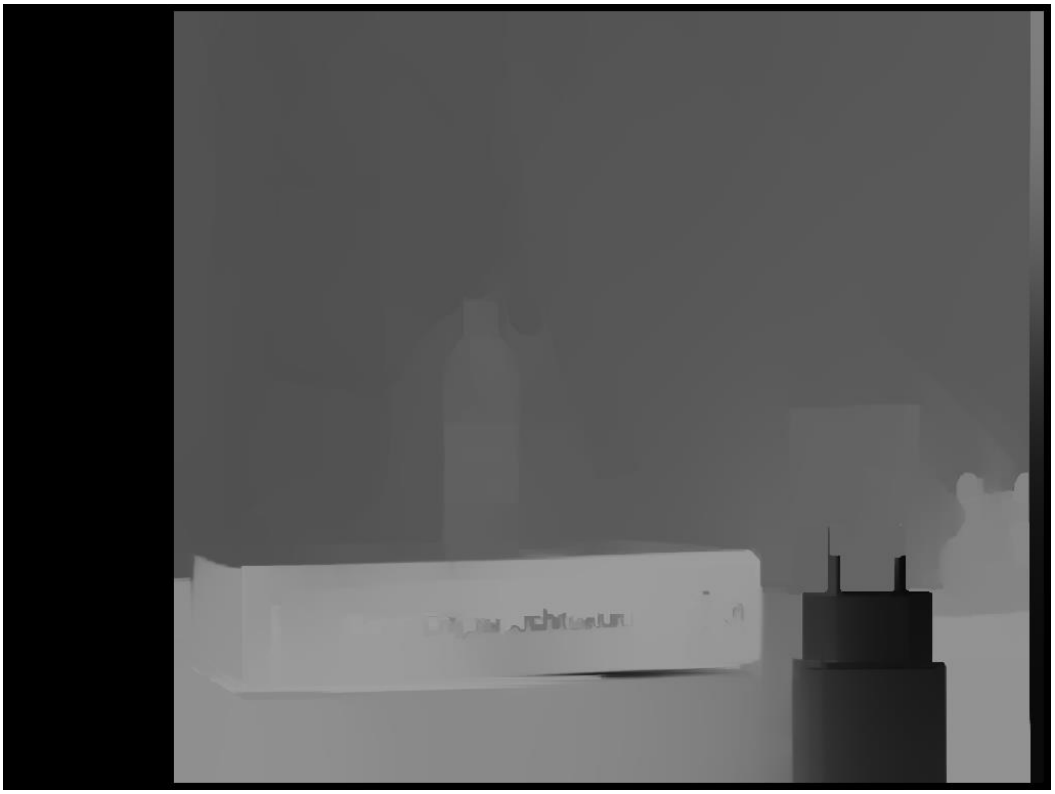
4.3 Kehitysversio 1

Kehitys aloitettiin yksinkertaisella kamera-asetelmalla ja keskitytään lähinnä saamaan hyvänlaatuisia syvyyskarttoja sekä pistepilviä. Tässä vaiheessa oli myös hyvä aika tutkia, kuinka algoritmien säädökset vaikuttavat lopputulokseen. Tässä versiossa käytettiin OpenCV-kirjastoa rajapintana kameroiden kuvien ottoon.

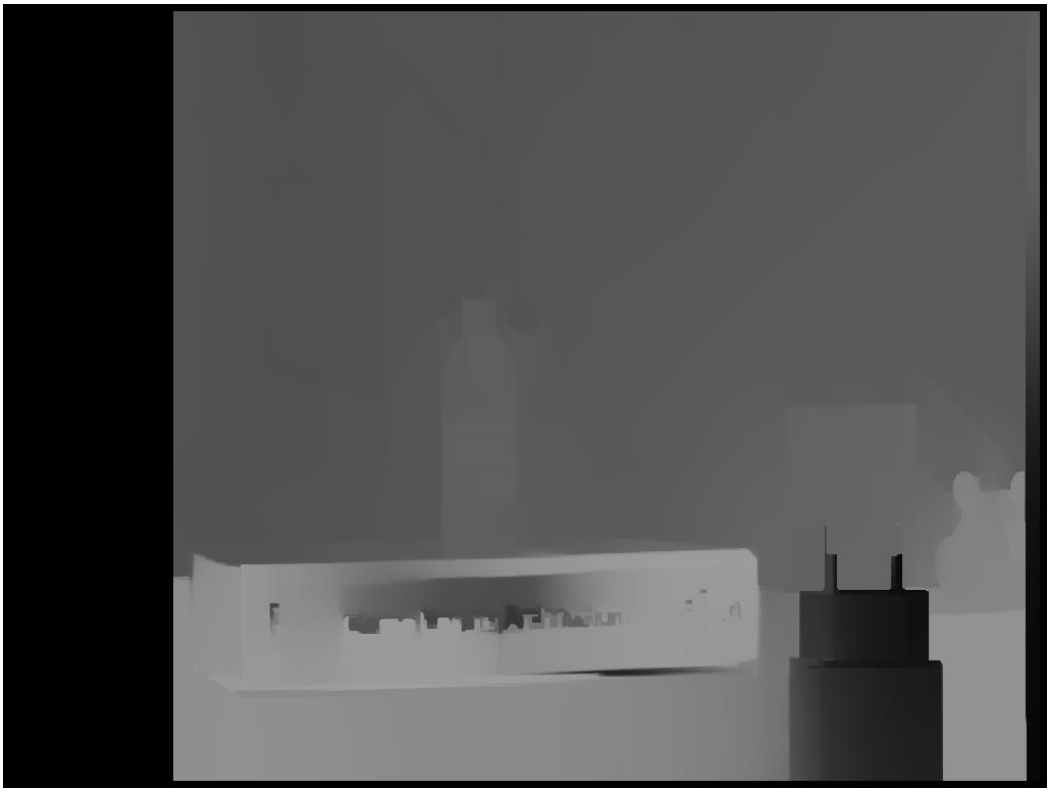
Web-kameroilla kuvattiin pelkästään paikalla olevia kappaleita. Web-kameroilla olisi ollut hyvin vaikeaa kuvata tarkasti liikkuvia kuvia liukuvan sulkimen takia. Web-kameroissa myös kuvien liipaisu valojen kanssa olisi ollut hyvin kankeaa.

Kuvassa 12 on eräs web-kameroiden tuottamasta syvyyskartta. Kuvassa on taustalla muovipullo vasemmalla, oikealla etualalla puhelin laturi, puhelin laturin takana pieni pahvilaatikko, pahvilaatikon etupuolella oikealla säästöpossu ja keskellä kirja. Muodostetuista syvyyskartoista saadut etäisyydet kuvattuihin kohteisiin erosivat kohteiden todellisista etäisyyksistä keskimäärin muutamia kymmeniä millimetrejä kuvausetäisyyden ollessa 50 cm. Laturin pään metallinen pinta oli systeemille ongelmallinen ja se yhdistyi takana olevaan laatikkoon. Kirjan oikeaan reunaan kirjan alle muodostuva varjo aiheuttaa myös virheellistä etäisyyden muodostumista.

Kuvassa 13 näkyy, kuinka valaisu ja web-kameran kohina aiheuttavat häiriötä kirjanpintaan. Häiriön suurin aiheuttaja on luultavasti kirjan kiiltävä pinta. Muut objektit ovat kuitenkin säilyneet varsin hyvinä.



Kuva 12. Kuva web-kameroiden tuottamasta syvyyskartasta



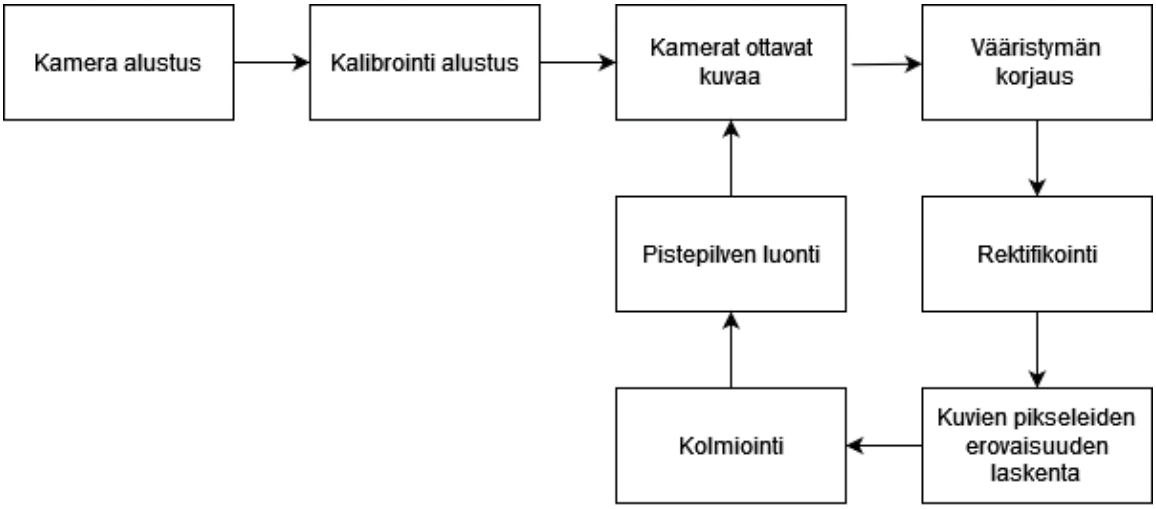
Kuva 13. Kuva, jossa näkyy kirjassa web-kameroiden aiheuttamaa kohinaa

4.4 Kehitysversio 2

Kehitysversiossa 2 siirryttiin halvoista web-kameroista parempiin Baslerin kameroihin. Baslerin kameroiden hyvä puoli on monipuoliset kamera-asetukset ja niiden säätämiseen löytyvä Pylon Viewer -sovellus.

Lisäksi siirryttiin käyttämään isompaa linjastoa (linja 2), joka vastaa hieman enemmän lopullisia olosuhteita. Tämän linjaston etuna on juuri automaattinen kuvaaminen ja viilun liikkuminen. Näin saadaan entistä enemmän testikuvia verrattuna siihen, että kuvat otettaisiin manuaalisesti ensimmäisellä testiympäristöllä. Tällä linjalla tuli projektiin mukaan myös pistepilvien luonti syvyyskartoista.

Kuvassa 14 on ohjelman lopullinen yleislogiikka. Aluksi alustetaan kamerat ja ladataan jokaisen erilliset kalibroinnit. Kamerat alkavat ottamaan kuvaa, joista korjataan vääristymät. Vääristymän korjauksen jälkeen kuvat rektifioidaan. Sitten lasketaan pikseleiden eroavaisuus, josta kolmioidaan syvyyskartta. Tämän jälkeen voidaan laskea syntyvästä syvyyskartasta pistepilvi.



Kuva 14. Vuokaavio ohjelman logiikasta

4.5 Testaus

Stereokuvantamisen tarkkuuta testattiin lähinnä manuaalisesti parilla erilaisella testikappaleella. Yhteen oli porattu erikokoisia reikiä ja toisessa oli taas liimattuna erikokoisia pultteja, muttereita ja prikkoja.

Ensimmäisen puupalikan reikien oikea koko tiedetään ja se oli merkitty tussilla puupalikkaan reikien yläpuolelle. Syvyyskartasta pystyi katsomaan, olivatko reiät tulleet näkyviin hyväksyttävällä tarkkuudella eli oliko reikä oikean muotoinen.

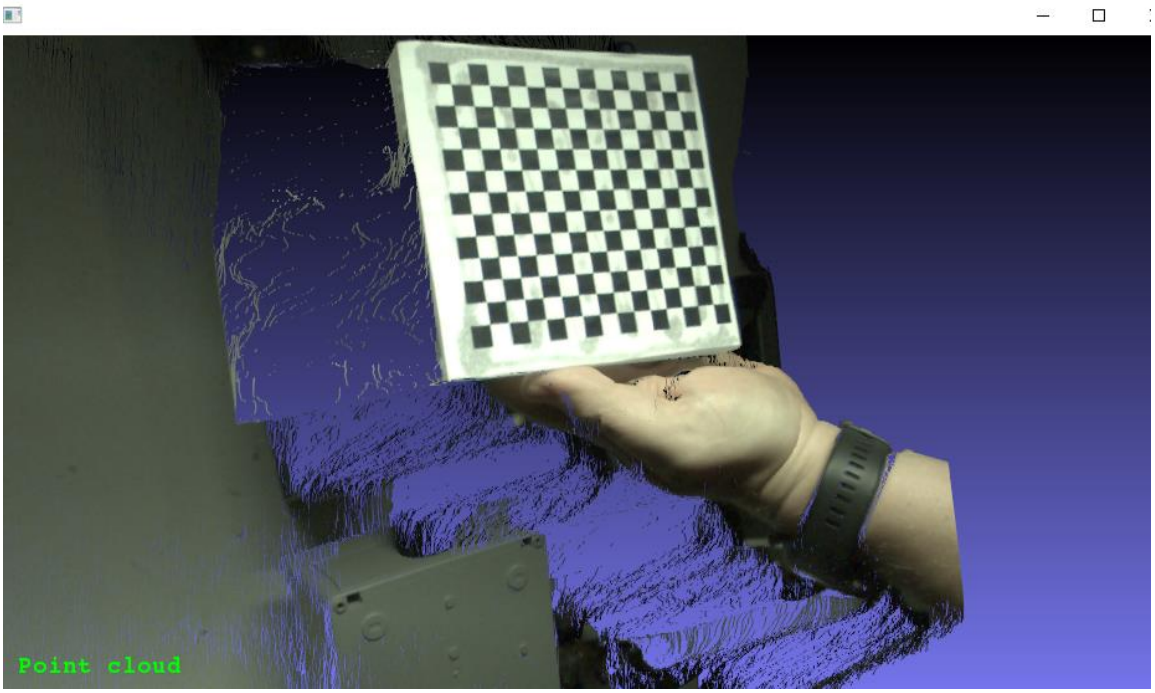
Pultti, mutteri ja prikka puukappaleen objektien oikea koko olivat myös tiedossa. Kun sovellus näytti syvyyskarttaa, siitä pystyttiin klikkaamaan ja tarkistamaan, olivatko kappaleet oikean kokoisia. Pultit oli liimattu kiinni kannastaan, joten varsinkin niiden päistä sai hyvän muutaman sentin korkeuseron puuhun. Näitä kappaleita oli useita samanlaisia vierekkäin, näin pystyttiin varmistamaan toimivuuden toistuvuus.

5 Tulokset

Tuloksia on aina hyvä tarkastella. Jotta työn lopputulosta voidaan hyödyntää käytännössä, täytyy syvyyskarttojen ja pistepilvien olla tarpeeksi tarkkoja.

5.1 Tuloksien tarkastelu

Systeemi pystyy tuottamaan stereokuvavirtaa, mutta nopeus ei tässä vaiheessa ole mitenkään hyvä. Stereokuvan tarkkuus on tämän harjoituksen kannalta sopiva. Kuvassa 15 on esimerkki, miltä syvyyskartasta muodostettu pistepilvi näyttää.



Kuva 15. Pistepilvivisualisointi

Kuvassa 16 on esitetty syvyyskartta mutteri, pultti ja prikka puupalikasta. Hieman vaaleammalla näkyvät kohdat ovat etäisyys kamerasta objektiin. Laskettujen etäisyyksien heitot olivat luokkaa 0,5 mm – 50 mm.

Kuva 16. kuva muttereista, pulteista ja prikoista

Taulukossa 1 on mitattu, kuinka paljon tietyn kokoiselle prikalle tai pultille tapahtuu mittausvirhettä. Tulokset olivat hyvin tarkkoja prikoissa, jotka olivat kooltaan pienempiä, uurteisia ja mattapintaisia. Kooltaan suuremmassa pultissa (pultti 1) tarkkuus heittää myös moninkertaisesti. Kooltaan matalampi pultti 2 antoi epätarkimmat tulokset. Tämä johtunee pultin täysin mustasta väristä, joka on aiheuttanut algoritmille ongelmia oikean eroavaisuuden määrittelyyn kolmiointia varten.

Kohde	prikka 1, 0.415 cm	prikka 2, 0.3 cm	pultti 1, 3.675 cm	pultti 2, 2.49 cm
Mittausvirhe (cm)	0.105519	-0.032892	0.333917	0.463288
	0.084587	-0.037752	0.316629	0.508099
	0.007461	-0.062066	0.297212	0.512980
	0.089545	-0.047474	0.297212	0.512980
	0.152999	0.012417	0.358199	0.453587
	0.023106	-0.044233	0.323097	0.472984

Taulukko 1. Mittausvirhe kuudesta eri syvyyskartasta

5.2 Pohdinta

Nopeusasioita ei ehditty kehitystyön aikana pohtimaan juuri ollenkaan. Pistepilvien laskennan mukaantulo kesken kaiken aiheutti myös sen, että optimoinnille ei jäänyt juuri yhtään aikaa. En-

nen optimointia kannattaa kuitenkin varmistaa, että systeemi toimii. Työn tuloksena saatiin varsin hyvä demonstraatio siitä, millaisella laadulla voidaan luoda syvyyskarttaa ja pistepilviä työssä käytettyjä kirjastoja hyödyntäen.

5.3 Kehityskohteet

Sovelluksen nopeutta voidaan parantaa massiivisesti käyttäen näytönohjainta. Näytönohjaimella monet kuvankäsittelyalgoritmit nopeutuvat moninkertaisesti. Kuvankäsittelyalgoritmit soveltuvat näytönohjaimille usein hyvin, koska ne ovat helposti rinnastettavissa. OpenCV:n mukaan näytönohjain parantaa stereokuvantamisen nopeutta seitsemän kertaa nopeammaksi [6].

OpenCV-kirjasto tukee valmiiksi Nvidian CUDA-rajapintaa. CUDA:lla on mahdollista ohjelmoida käyttäen C++-ohjelmointikieltä näytönohjaimelle. Koodiin pitää tietenkin tehdä muutoksia ennen kuin se on mahdollista kääntää näytönohjaimella pyöriväksi. CUDA tarjoaa hyvän integraation kehitysympäristöihin, mahdollistaen jopa debuggerin käytön. Nvidia on myös tehnyt työkalun näytönohjain-kerneleiden profilointiin.

Nopeutta voidaan myös parantaa säikeistämällä. Kun kuvat tulevat kameralta, voitaisiin niille tapahtuvat tehtävät toteuttaa omalla säikeellään. Säikeistyksen tarve tietenkin laskisi, jos suurin osa kalliista laskemisesta olisi näytönohjaimella. Parempi ratkaisu on laittaa kaikki työ näytönohjaimelle, koska tässä työssä käytetyt kuvankäsittelyalgoritmit soveltuvat hyvin näytönohjaimelle.

Testaamisen kannalta kannattaisi tulevaisuudessa ajaa järjestelmää myös oikean kokoisella testiympäristöllä. Tällä tavalla olisi mahdollista saada parempaa dataa, siitä miten järjestelmä käyttäytyy oikeassa ympäristössä.

6 Yhteenveto

Tämän opinnäytetyön tavoitteena oli toteuttaa Raute Oyj:lle stereokuvantamisjärjestelmä. Järjestelmä ottaa kuvia liikkuvasta kohteesta ja tekee niistä kuvavirtaa, syvyyskarttaa ja pistepilven. Työn tuloksena syntyi järjestelmä, joka tekee juuri nämä laskut, mutta hieman hitaamman puoleisesti.

Stereokuvantamalla voidaan laskea etäisyyksiä kahden tai useamman kameran avulla, niiden näkökentän päällekkäisyyden alueelle. Etäisyyden laskeminen perustuu kolmiointiin.

Työlle asetettuna vaatimuksena oli, että kuvavirtaa pitäisi syntyä 2 metriä sekunnissa. Optimointivaiheeseen ei ehditty käyttämään aikaa, koska aluksi täytyi keskittyä tarkkuuteen. Kuvavirtaa syntyi noin 0,3 metriä sekunnissa. Suurin syy hitaaseen kuvavirran muodostamiseen oli se, että työssä ei hyödynnetty näytönohjainlaskentaa. Varsinkin syvyyskuvan laskenta vei yli 85 % prosenttia koko ohjelman suoritusajasta. Tulevaisuuden kannalta projektiin tulee isoja muutoksia, mutta stereolaskentaan varmasti palataan vielä.

Työtä tehdessä opittiin paljon uutta konenäön hyödyntämisestä käytännöntilanteessa. Työn toteutusvaihe eteni mutkattomasti, mutta dokumentointivaihe oli rankka. Asiatekstin kirjoittaminen on raskasta ja vaati useita iterointikierroksia.

Tulokset olivat mainioita pienillä prikoilla testattaessa. Isommilla kappaleilla virheen määrä moninkertaistuu. Työ täyttää vaatimusmäärittelyn stereokuvantamisen laadun suhteen, koska sitä ei suoraan määritelty. Asiakas oli tyytyväinen lopputulokseen ja työstä opittuja asioita voidaan käyttää jatkossa. Pistepilveä ja syvyyskarttaa voidaan hyödyntää tulevaisuudessa viilun analysoinnissa ja suoristamisessa.

Lähteet:

Kirja:

1 Digital Image Processing 2nd edition William K. Pratt: 21-28

Kuva:

2 Kuva ihmissilmästä. Saatavilla 15.5.2022. Internetosoite: https://commons.wikimedia.org/wiki/File:Ihmisen_silm%C3%A4.png

Blogi:

3 Blogi. Saatavilla 15.5.2022 internetosoite: https://www.haaraamo.fi/2011/04/camera-obscura.html

E-Kirja:

4 Stemmer imaging & vision handbook. Saatavilla 29.5.2022. Internetosoite: https://reader.paperlit.com/read/prj_599d388f9f483/pub_599d38901ad39/5000-01-05?token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJwcm9qZWN0SWQiOiJwcmpfNTk5ZDM4OGY5ZjQ4MyIsInByb2plY3RQZXJtaXNzaW9uc0JpdG1hc2siOjF9.w32x0nBNNrqF2Wx-YyVNtAOy3CQQPjuZH-PoFrQE9VmE&hasNewsstand=true&page=14

Kuva:

5 Kuva vääristymistä. Saatavilla 22.5.2022. internetosoite: https://docs.opencv.org / 3.4/distortion_examples.png

Dokumentaatio:

6 OpenCV dokumentaatio. Saatavilla 22.5.2022. internetosoite: https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html#ga617b1685d4059c6040827800e72ad2b6

Kuva:

7 Kuva stereo rektifioinnin tuloksesta. Saatavilla 29.5.2022. internetosoite: https://docs.opencv.org/3.4/stereo_undistort.jpg

8 Kuva kolmioinnista. Saatavilla 29.5.2022. internetosoite: https://docs.opencv.org/4.x/stereo_depth.jpg

Kirja:

9 Berthold Klaus, Paul Horn. Robot Vision. Ninth printing (1993). The Mit Press.

Kotisivut:

10 OpenCV kotisivut. Saatavilla 28.5.2022. internetosoite: https://opencv.org/about/

11 OpenCV kotisivut. Saatavilla 28.5.2022. internetosoite: https://opencv.org/platforms/cuda/

12 Pointcloud library. Saatavilla 28.5.2022. internetosoite: https://pointclouds.org/about/