

# **Kamerakalibraatio yhden kameran järjestelmässä**

Tietotekniikan laitos, Teknillinen tiedekunta  
Kandidaatintutkielma

Laatija:  
Onni Erjala

Ohjaaja:  
Jukka Heikkonen

huhtikuu 2025

Turun yliopiston laatujärjestelmän mukaisesti tämän julkaisun alkuperäisyys on tarkastettu  
Turnitin OriginalityCheck -järjestelmällä.

**Kandidaatintutkielma**  
**Tietotekniikan laitos, Teknillinen tiedekunta**  
**Turun yliopisto**

**Oppiaine:** Tietotekniikka

**Tutkinto-ohjelma:** Tietotekniikka

**Tekijä:** Onni Erjala

**Otsikko:** Kamerakalibraatio yhden kameran järjestelmässä

**Sivumäärä:** 21 sivua

**Päivämäärä:** Huhtikuu 2025

Tutkielmassa perehdytään kamerakalibraatioon OpenCv-konenäkökirjaston avulla. Keskeisiä tutkimuskysymyksiä ovat: Miksi, ja miten kamerakalibraatiota suoritetaan? Tutkielman tavoitteena on tarjota lukijalle riittävä ymmärrys kamerakalibraation perusteista, jotta lukija kykenee itse syventymään aiheeseen tarkemmin haluamallaan osa-alueella. Tutkielma on itseriittävä, eli sen tavoitteena on tarjota tarvittavat ennakkotiedot aiheen ymmärtämiseksi.

Aiheena konenäkö on ajankohtainen ja tärkeä, sillä tekoälyn ja koneoppimisen yleistyessä, myös konenäön sovellutukset ovat yleistyneet kautta linjan. Konenäön yleistyessä monet ovat kiinnostuneita aiheesta, mutta kattavaa valmiiksi koottua tietoa voi olla haastava löytää. Tutkielman on tarkoitus vastata tähän tarpeeseen.

Tutkielmassa perehdytään alan kirjallisuuteen ja artikkeleihin, sekä tarjotaan tarvittava aineisto kamerakalibraation suorittamista varten käyttäen OpenCv-kirjastoa. Erityistä huomiota kiinnitetään kalibroitamiseksi läpikäyntiin sekä virheanalyysiin, sillä uudelle käyttäjälle toimintamallin tarjoaminen on keskeistä aiheen sisäistämiseksi. Virheanalyysiosiossa tarjotaan menetelmiä ja teoriaa, joita hyödyntämällä käyttäjä voi arvioida saamiaan tuloksia, sekä täten kasvattaa tulostensa luotettavuutta. Keskeisiä tutkimuskohteita tutkielmassa ovat kamerakalibraatio, OpenCv ja Zhangin menetelmä.

**Asiasanat:** Tietokonenäkö, Kamerakalibraatio, OpenCv, Zhangin menetelmä

# Sisällysluettelo

<b>1</b>	<b>Johdanto</b>	<b>5</b>
<b>2</b>	<b>Yksittäisen kameran kalibrointi</b>	<b>6</b>
<b>2.1</b>	<b>Kameran sisäiset ja ulkoiset parametrit</b>	<b>6</b>
<b>2.2</b>	<b>Kalibroitimenetelmät</b>	<b>7</b>
2.2.1	Shakkilautamenetelmä	7
<b>2.3</b>	<b>Virheanalyysi</b>	<b>10</b>
2.3.1	Systemaattisen virheen analysointi	11
2.3.2	Epävarmuustekijän luoman virheen analysointi	11
<b>3</b>	<b>Kalibroinnin suorittaminen</b>	<b>12</b>
3.1.1	Minimivaatimukset	12
3.1.2	Kalibrointiaineisto	13
3.1.3	Ohjelmisto	14
3.1.4	Virheanalyysi ja apuohjelmat	16
<b>4</b>	<b>Yhteenveto</b>	<b>19</b>
	<b>Viitteet</b>	<b>20</b>



# 1 Johdanto

Kamerakalibraatio on keskeisessä roolissa luotettavan konenäön luomiseksi. Luotettavasti kalibroidun kameran avulla ympäristöä voidaan havainnoida kaksiulotteisten kuvien sijaan kolmiulotteisesti, jolloin konenäköä voidaan käyttää mittavälineenä. Nykyaikaiset kalibrointialgoritmit ja -menetelmät mahdollistavat yksinkertaisen kalibrointiprosessin, joka mahdollistaa luotettavien tulosten saavuttamisen yksinkertaisimmillakin järjestelmillä.

Konenäön sovellutuksista keskeisimmät liittyvät asioiden ja esineiden tunnistamiseen, sekä ympäristöjen, kuten pistepilvien luomiseen. Yksinkertaiset esineentunnistustehtävät voidaan toteuttaa ilman kalibrointia, mutta erityisesti teollisuudessa käytetyt sovellutukset vaativat kameran tarkkaa kalibrointia. Kalibroidessa pyritään määrittämään kameran luomien linssivääristymien perusteella kameran erinäisiä parametreja, näiden parametrien avulla voidaan luoda kaksiulotteisesta kuvasta kolmiulotteista dataa.

Tutkielmassa tutustutaan kamerakalibraation perusteisiin yhden kameran järjestelmässä. Keskeisiä termejä ovat kameran sisäiset ja ulkoiset parametrit, neulanreikäkamera ja Zhangin kalibrointimenetelmä. Tutkielma pohjautuu OpenCV-konenäkökirjaston tarjoamiin työkaluihin ja syventyy erityisesti Zhangin menetelmällä suoritettuun kamerakalibraatioon. Tavoitteena on luoda kattava kuva kamerakalibraation perusteista sekä tarjota tarvittavat lähtötiedot kalibrointijärjestelmän luomiselle Zhangin menetelmällä.

Tutkielmassa edetään perusteista, kuten neulanreikäkameramallista, aina kameran onnistuneeseen kalibrointisuoritukseen. Lisäksi käydään läpi kamerakalibraation teoriaa sekä perehdytään aiheesta tehtyihin artikkeleihin sekä kirjallisuuteen. Lähteet on pyritty valitsemaan niin, että ne sisältävät riittävät tiedot, mutta käsittelevät aihetta tavalla, joka vasta-alkajan on helppo ymmärtää. Lyhyempien tutkimusartikkelien tueksi on haettu kattavia teoksia, joiden avulla lukija voi perehtyä aiheeseen laajemmin. Loppusuoritteena esitetään menetelmä kameran kalibroimiseksi OpenCV-kirjaston avulla ja tarkastellaan, kuinka käyttäjä voi varmistua tuloksensa luotettavuudesta virheanalyysin avulla.

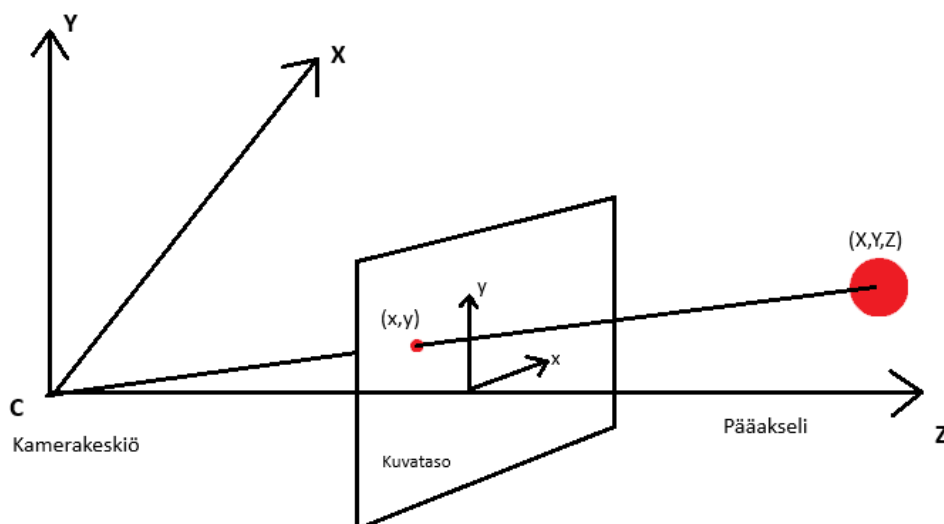
## 2 Yksittäisen kameran kalibrointi

Kamerakalibraatio on yksi tietokonenäön keskeisistä sovellutuksista. Kalibraation avulla pyritään määrittämään kameran ulkoiset ja sisäiset parametrit mahdollisimman tarkasti. [1] Tällaista kalibrointia kutsutaan myös resektioksi. Se tarkoittaa valon säteiden määrittämistä pikseliä kohden. Tarkoin kalibroidulla kameralla kyetään tuottamaan tarkempaa dataa esimerkiksi silloin, kun yritetään määrittää fyysisen objektin paikkaa kolmiulotteisessa tilassa. [2]

### 2.1 Kameran sisäiset ja ulkoiset parametrit

Onnistuneen kalibraation kannalta etsittävät parametrit voidaan jakaa kameran sisäisiin ja ulkoiisiin parametreihin. Sisäiset parametrit tarkoittavat linssin ominaisuuksia, kuten pysty- sekä vaakasuuntaista polttoväliä, kameran optista keskipistettä ja linssistä aiheutuvia vääristymiä. Ulkoiset parametrit tarkoittavat olosuhteita, kuten kameran sijaintia ja katselukulmaa kuvattavaan objektiin nähden. [3]

Lisäksi parametrit voidaan luokitella kolmeen erilliseen koordinaatistoon: Todellisen maailman koordinaatisto, joka kertoo sijainnin kolmiulotteisessa tilassa parametrein  $X, Y$  ja  $Z$ , kameran koordinaatit  $XYZ$ , jotka kertovat kameran katselukulman origon ollessa kameran keskipiste, ja kuvan koordinaatisto  $xy$ , jonka avulla määritetään kuvattavan objektin sijainti kuvatasolla. [4] Kuvassa 1 esitetään neulanreikäkameramalli, jossa mainitut koordinaatit sekä koordinaatistot on visualisoitu.



Kuva 1: Neulanreikäkameramalli.

Neulanreikäkameran kalibrointi perustuu paikanmäärittämiseen, jossa uudelleenprojektion pohjana määritetään reaali maailman piste  $(X, Y, Z)$  kuvatason pisteeseen  $(f X/Z, f Y/Z, f)$ . Viimeinen kuvatason koordinaatti voidaan mitätöidä, sillä kuvataso on kaksiulotteinen ja sijaitsee pisteessä  $Z=f$ ,  $f$ = kameran polttoväli, jolloin saadaan  $(X, Y, Z) \rightarrow (f X/Z, f Y/Z)$ , jossa  $fX/Z = x$  ja  $fY/Z = y$ .

Tällöin voidaan tunnettujen  $x, y$  ja  $f$ -arvojen avulla luoda neulanreikäkameran ideaalmatriisi:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad [5]$$

## 2.2 Kalibrointimenetelmät

Kameran parametreja voidaan laskelmoida eri tavoilla. Metodeista tunnetuin on shakkilautamenetelmä, jossa kuvasta tunnistettavan ruudukon avulla voidaan laskea kameran sisäiset parametrit. Shakkilautamenetelmän etuihin lukeutuu sen helppokäyttöisyys ja laajat sovellutukset, sekä tarkkuus, mikäli tunnistettavan ruudukon parametrit, kuten ruudun koko ja rivien määrä, ovat tunnettuja. [6] Toinen tunnettu menetelmä tasomaisella kalibrointivälineellä on geometrinen kalibraatio pyöreiden merkkien avulla, joka mahdollistaa jopa 1/50 pikselin tarkkuuden. [7] Pyöreiden merkintöjen tunnistaminen on kuitenkin herkempi vääristymille, kuin shakkilautamenetelmä, minkä vuoksi menetelmä on työläämpi ja tarjoaa vähemmän joustavuutta. [6] Näiden lisäksi voidaan käyttää myös valon heijastumiseen ja valovoimaan liittyviä menetelmiä, mutta niiden käytännöllisyys stereonäössä on toistaiseksi hyvin rajallinen. [6] Valon heijastusmenetelmässä valonsäteet ohjataan kameran linssille esimerkiksi toisen linssin avulla, jolloin voidaan tietää valonsäteiden suunta, sijainti ja kulma. Tyypillisesti näitä menetelmiä käytetään yhden kameran järjestelmissä, joiden tarkennuspiste on äärettömän kaukana. Tällaisia kameroita käytetään pitkän matkan kuvantamisessa, esimerkiksi karttoja tehtäessä, kun kuvataan maan pinnan muotoja. [8] Seuraavassa alaluvussa perehdytään tarkemmin shakkilautamenetelmään.

### 2.2.1 Shakkilautamenetelmä

Shakkilautamenetelmä on Zhangin kalibrointimenetelmä, jossa kameralla havainnoidaan tasomaista kalibrointivälinettä. Tasomainen kalibrointiväline muodostaa tason  $XY$ , sillä oletetaan, että kaikki pisteet tasolla sijaitsevat  $Z=0$  tasolla. Näin kuvaa analysoimalla saadaan kameran sisäisten parametrien lisäksi laskettua myös ulkoiset parametrit, eli rotaatio- ja translaatiovektorit. Zhangin menetelmässä kaksiulotteinen koordinaatti kirjataan  $m = [u, v]$  ja kolmiulotteinen vastaava piste on  $M = [X, Y, Z]$ . Täten takaisinprojektio voidaan laskea kaavalla  $s m = A[R \ t]M$ , jossa  $s$  on pisteen skaala,  $R$  ja  $t$  ovat ulkoiset parametrit ja  $A$  on kameran sisäiset parametrit, eli aikaisemmin mainittu neulanreikäkameran matriisi. Tuntemattomaksi muuttujaksi jää vain ulkoiset parametrit, jotka voidaan laskea kaavalla:

$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$  Z voidaan eliminoida, sillä oletusarvoisesti kalibrointitasolla  $Z = 0$ . Kaava

voidaan jälleen jalostaa rotaatio ja translaatiovektoreiden laskemiseksi muotoon  $sm = HM$ , jossa  $H = A[r_1 \ r_2 \ t]$ . [9, 10]

Shakkilautamenetelmä perustuu kulmanmäärittämiseen, jossa valkoisten ja mustien ruutujen kulmat tunnistetaan kuvasta, minkä jälkeen niiden etäisyyttä sekä kulmaa toisiinsa nähden arvioimalla lasketaan kameran linssin piirteitä. Shakkilautamenetelmän suosio perustuu sen helppouteen, sillä tarvittava ruudukko voidaan tulostaa paperille ja tavanomainen lasertulostin pystyy tuottamaan kuvan tarvittavalla tarkkuudella. Lisäksi shakkilautamenetelmälle löytyy tuki yleisimmistä työkalupakeista, kuten Matlab, DLR Camera Calibration Toolbox, sekä Intelin OpenCv-kirjastosta. [3] Tyypillisesti käyttäjän tarvitsee syöttää ohjelmistolle vain kuvat ruudukosta ja ruudukon piirteitä, kuten rivien määrä ja ruutujen koko. Lisäksi ohjelmisto voi pyytää käyttäjää paikantamaan ruudukon kuvasta, virhetunnistuksien välttämiseksi. Lauta voidaan paikantaa myös automaattisesti kulmien tunnistuksen avulla, mikäli pysty- sekä vaakarivien määrä on tiedossa ja ruudukon kaikki uloimmat kulmat näkyvät kuvassa. [3] Sijainti voidaan tunnistaa myös merkintöjen avulla (AprilTags), jotka määrittävät ruudukon kulmat sekä keskipisteen että orientaation. Näin välttyään virheiltä, kun ruudukko näkyy kuvassa vain osittain ja mahdollistetaan ruudukon automaattinen tunnistaminen. Toistaiseksi tietokonenäön työkalupakeista DLR Camera Calibration Toolbox tarjoaa parhaimman tuen AprilTagien käytölle, sillä se mahdollistaa jyrkemmän katselukulman kuin esimerkiksi OpenCv. [6]

Automaattisen ruudukon tunnistamisen avulla voidaan välttyä yleisimmiltä virheiltä kalibrointia tehtäessä, sillä inhimillinen tekijä on pienemmässä roolissa. Automaatio pienentää myös vaadittua ammattitaitoa kalibrointia tehtäessä, sillä käyttäjän ei tarvitse kattavaa tuntemusta aiheesta. [3] Tutkielmassa perehdytään kamerakalibraatioon OpenCv:n avulla, sillä OpenCv tarjoaa laajan kirjon erilaisia valmiiksi toteutettuja työkaluja ja on ilmainen. Lisäksi OpenCv:n suosio takaa laajan

dokumentaation, joka edesauttaa sen käyttöönottoa taitotasosta riippumatta. Seuraavaksi perehdytään tarkemmin OpenCv-kirjaston shakkilautamenetelmään yksittäistä kameraa kalibroitaessa.



*Kuva 2: Kalibroitiruudukko, jossa kulmat määritetty ja visualisoitu.*

Kalibrointiin käytetty ruudukko on jaettu kalibroinnin kannalta olennaisiin osiin, joita ovat mustat ja valkoiset ruudut, niiden risteymäkohdat ja ruudukon kulmaruudut. Ruuduilla tarkoitetaan mustien tai valkoisten alueiden luomaa suorakulmiota, eikä ruutujen täten tarvitse välttämättä olla tasasivuisia. Tutkielmassa perehdytään kuitenkin yksinomaan neliöistä muodostuviin ruudukoihin ja kalibrointiin niiden avulla. Kulmat tarkoittavat neliöiden risteymäkohtia, joissa neljän ruudun kulmat kohtaavat ja kulmaruudut luonnollisesti tarkoittavat ruudukon uloimpia ruutuja ruudukon laitojen kohtaamiskulmassa, katso kuva 2. Lisäksi ruudukko voidaan jakaa riveihin ja sarakkeisiin, joiden määrä voidaan syöttää kalibrointiin käytetylle järjestelmälle, jolloin järjestelmän ei tarvitse itsenäisesti laskea ruudukon kokoa suunnissa X ja Y. [9]

OpenCv tarjoaa kattavan kirjaston shakkilautamenetelmän käyttämiselle olettaen, että otettujen kuvien olosuhteet ovat suotuisat. Keskeisimmässä roolissa ovat ruudukon näkyvyys, eli kuvakulma sekä valaistus. Hyvässä valaistuksessa ruudukko voidaan tunnistaa tarkemmin, mutta kuten mainittu, liian jyrkkä kuvakulma voi aiheuttaa virhelukemia kameran parametrejä laskettaessa. [9] Olennaista on myös ruudukon näkyvyys kuvassa, sillä OpenCv:n ruudukon tunnistus perustuu kulmien paikantamiseen, jolloin ruudukon on oltava näkyvissä kokonaisuudessaan, mikäli ruudukko tahdotaan tunnistaa

automaattisesti. OpenCv ei siis tue AprilTagejä, eikä voi täten määrittää ruudukon orientaatiota erillisten merkintöjen avulla. [6] Yksinkertaistettuna OpenCv kalibroitintiprosessi voidaan jakaa kolmeen osaan:

1. Kalibroitavalla kameralla otetaan kuvia kalibroitiruudukosta.

Parhaan mahdollisen tuloksen takaamiseksi ruudukon asentoa ja sijaintia suhteessa kameraan tulisi muuttaa kuvien ottamisen välillä. Luotettavan kalibraation saavuttamiseksi kuvia tulisi olla vähintään kolmesta eri kuvakulmasta, sillä haluttu kalibroitimatriisi sisältää kuusiulotteisen vektorin. [9] Kuvien määrää lisäämällä voidaan parantaa kalibroinnin tarkkuutta. [11]

2. Etsitään ruudukon keskeiset piirteet OpenCV kirjaston funktioiden avulla. Aluksi etsitään kuvasta ruudukon kulmaruudut funktiolla *cv.findChessboardCorners()*, joiden avulla voidaan määrittää ruudukon sijainti kuvassa. Tämän jälkeen voidaan etsiä ruudukon risteymäkohdat funktiolla *cv.cornerSubPix()*.

3. Viimeisenä voidaan laskea kameran sisäiset parametrit funktiolla *cv.calibrateCamera()*, joka palauttaa kameran sisäiset parametrit matriisinä A, sekä ryhmänä K. [12]

Kalibroinnin jälkeen kuva voidaan uudelleenprojisoida backprojection-algoritmillä, joka poistaa linssivääristymän ja pyrkii luotettavasti linjaamaan kuvan kaksiulotteiset pisteet reaali maailman kolmiulotteisten pisteiden kanssa. Tyypillisesti tähän käytetään lineaarimuunnosta, eli Direct Linear Transform -metodia, DLT- tai Fourier-muunnosta. [9] Ennen kalibroitituloksen käyttämistä on kuitenkin hyvä tarkastaa tuloksen luotettavuus suorittamalla kattava virheanalyysi.

## 2.3 Virheanalyysi

Virheet voidaan jakaa kahteen kategoriaan: epävarmuustekijöihin, kuten kalibroitiruudukon näkyvyys ja systemaattisiin virheisiin, eli virheisiin kalibroitintiprosessissa. Tällaisia virheitä voivat olla esimerkiksi kalibroiintiin käytettävä kuva, jossa ruudukko näkyy vain osittain, on liian jyrkässä kulmassa kameraan nähden tai väärät raja-arvot kalibroitiohjelmassa. [13] Tyypillisesti keskimääräistä takaisinprojektivirhettä voidaan pitää luotettavana, mikäli käytetty kalibroitidata on tarpeeksi laaja. Mikäli käyttäjä ei kykene tarjoamaan laajaa kirjoa kuvia kalibroitivälineestä eri kuvakulmissa, tai muusta syystä epäilee kalibroitituloksen luotettavuutta, tulee suorittaa kattavampi virheanalyysi. [13-15]

Kalibroitituloksen luotettavuutta voidaan arvioida erilaisin keinoin. Yksinkertaisimmillaan virheanalyysi voidaan suorittaa manuaalisesti arvioimalla tai käyttämällä erilaisia vakiotyökaluja, kuten esimerkiksi Matlab-kirjaston *displayErrors* metodia. [16] Tarkemman tuloksen varmistamiseksi käyttäjän tulisi luoda erilaisia keinoja arvioida keskimääräistä virhettä kalibroitintiprosessissa, sekä työkalu, jolla saatu virhe voidaan muuntaa reaali maailman arvoihin.

### 2.3.1 Systemaattisen virheen analysointi

Artikkelissa [14], ehdotetaan keinoa systemaattisen virheen analysoinnille. Standardoitu metodi virheen analysoinnille tarjoaa helpokäyttöisen keinon virheen arvioinnille. Kamerakalibroinnin kehittyessä kalibrointi itsessään on yksinkertaista myös amatöörikäyttäjälle, mutta virheanalyysi on toistaiseksi ollut haastavaa.

Tarjottu metodi perustuu kalibrointidatan kannalta suppean tuloksen analysointiin, jossa ruudukosta erotellaan pienempiä kokonaisuuksia virheen arviointia varten. Esimerkkisuoritteessa aineiston supistus on viety äärimmilleen ja tutkitaan yksittäisen ruudun kulmia. Supistetun kalibrointidatan tutkiminen on analyysin kannalta edullista, sillä virheen vaikutus on oletettavasti pienempi. Kun pienempien osavirheiden keskiarvoa verrataan kalibrointimallin laskennalliseen virheeseen, voidaan arvioida laskennallisen virheen luotettavuutta. Analyysitulosta verrataan vähentämällä menetelmän tuottama osavirhe keskimääräisestä laskennallisesta virheestä, tuloksen luotettavuus kasvaa erotuksen lähestyessä nolaa.[13]

### 2.3.2 Epävarmuustekijän luoman virheen analysointi

Epävarmuustekijöitä on haastavaa analysoida tarkasti, mutta epävarmuustekijöiden minimointi on yksinkertaisempaa. Käyttäjä voi itse vaikuttaa epävarmuustekijään luomalla kattavan määrän kalibrointikuvia, eikä epävarmuustekijän luoma virhe ole täten riippuvainen kalibrointialgoritmin tarkkuudesta kuten projektiovirhe. [13-14]

Aikaisemmassa luvussa mainittu artikkeli tarjoaa keinon myös epävarmuustekijöiden arvioinnille, mutta tutkimus osoittaa, että analyysi tarjoaa vain viitteellisen ideaaliarvon reaali maailman toteutukselle. Analyysi suoritetaan vertaamalla pisteenmääritysvirhettä takaisinprojektiovirheeseen. Näin saadaan eroteltua systemaattinen ja epävarmuustekijöiden luoma virhe. Virheanalyysin keskeinen ongelma on epävarmuustekijälle ominainen ongelma, jossa häiriöisen kuvan arviointi on itsessään haastavaa. Epävarmuustekijän määrittäminen on haastavaa, sillä pyritään määrittämään mikä saatavilla olevasta tiedosta on virheellistä, kun absoluuttista virhettä ei tunneta. [14]

### 3 Kalibroinnin suorittaminen

Seuraavassa luvussa perehdytään kamerakalibraation suorittamiseen. Luvussa selvitetään yksittäisen, eli monokulaarisen, kamerajärjestelmän kalibrointiin tarvittavat minimivaatimukset ja tarjotaan saatavilla olevan tiedon perusteella käypä ratkaisu linssivääristymien poistamiselle. Lisäksi perehdytään reaali maailman mittauksiin kaksiulotteisen kuvan avulla ja tutustutaan tarkemmin openCV kirjaston tarjoamiin kalibrointi sekä virheanalyysi menetelmiin. Tarvittava lähdekoodi on saatavilla osoitteesta: <https://gitlab.utu.fi/ovverj/kamerakalibraatio>, tai etsimällä sivulta [gitlab.utu.fi](https://gitlab.utu.fi) käyttäjä @ovverj ja avaamalla projektin kamerakalibraatio. Kalibrointiohjelmisto on rakennettu openCV:n tarjoamien esimerkkien avulla. [12] Ohjelmistoa on laajennettu soveltaen tämän tutkimuksen lähteitä. [4-5, 17]

#### 3.1 Minimivaatimukset

Kalibrointi suoritetaan Zhangin shakkilautamenetelmällä käyttäen openCv:n tarjoamaa kamerakalibraatiokirjastoa. Minimivaatimukset ovat tulostettu kalibroituvuus, kalibroitava kamera ja laite, jolla kalibrointi suoritetaan. Teoriassa jokainen voi suorittaa kalibraation esimerkiksi kannettavan tietokoneen ja puhelimen avulla, jolloin ei tarvita tulostinta paperisen kalibroituvuuden tulostamiseen. Esimerkin mukainen kalibrointi voitaisiin yksinkertaistettuna suorittaa annetulla ohjelmistolla kaksivaiheisesti. Ensin otetaan puhelimen kameralla kuvia kalibroituvuudesta, joka näkyy tietokoneen näytöllä, sitten kuvat siirretään tietokoneelle ja suoritetaan kalibrointi.



*Kuva 3: Esimerkissä käytetyt välineet.*

Esimerkkisuoritteessa käytetään vähimmäisvaatimusten lisäksi lasilevyä, johon lasertulostimella tulostettu ruudukko kiinnitetään, kameratelineettä, sekä lasilevyyn kiinnitettyjä kahvoja, kuva 3. Lasilevy toimii tasaisena alustana ruudukon kiinnitykselle, jolloin paperiruudukko pysyy mahdollisimman suorana kuvia otettaessa. Laitteeksi sopii mikä tahansa Linux-, Mac-, tai Windows-käyttöjärjestelmää käyttävä laite. Esimerkissä käytetään Raspberry Pi 4 tietokonetta, jossa on linux käyttöjärjestelmä. Ohjelmisto on koeajettu myös windows käyttöjärjestelmällä. Kamera on raspberry Pi tarvikekamera, jolla otetaan 1080x1920-kokoisia kuvia. Käytetyn kameran ominaisuudet, kuten polttoväli ja kennon koko, on saatu valmistajan sivuilta. [18]

### **3.2 Kalibrointiaineisto**

Kalibrointiaineisto, eli käytettävät kuvat, otettiin päiväsaikaan kirkkaassa valaistuksessa ruudukon näkyvyyden varmistamiseksi. Kuvat on otettu staattisella kameralla, joka on asemoitu kameratelineen avulla. Mahdollisimman tarkan kalibrointituloksen varmistamiseksi otettiin 12 kuvaa, joiden välillä ruudukon kulmaa sekä etäisyyttä kameraan nähden muutettiin, katso kuva 3. Kameran ominaisuuksien johdosta valaistukseen oli kiinnitettävä erityistä huomiota, sillä käytetty kamera on alun perin varustettu

infrapunavaloin ja sitä on käytetty pimeänäkökamerana. Tämän takia kamera vaatii erityisen kirkkaan valaistuksen ottaakseen tarkkoja kuvia.

Kuvien ottamisen jälkeen ruudukon näkyvyys tarkistettiin kulmanmääritys-algoritmien avulla ja tulokset tarkistettiin manuaalisesti. Otetuista kuvista yksi hylättiin tarkistuksen myötä, sillä liian jyrkkä katselukulma ei mahdollistanut automaattista kulmanmääritystä. Välineistön puutteet eivät kuitenkaan luoneet merkittävää rasitetta, sillä mustavalkoinen kalibroitiruudukko on kohtalaisen helppo erottaa myös heikommassa valaistuksessa. Lopullinen kalibrointi suoritettiin 11 kuvan kuvasarjan avulla. Kuvassa 4 esimerkkejä kalibrointiin käytetyistä kuvista.



*Kuva 4: Esimerkkikuvia kalibroidatasta, ruudukon sijainnin, sekä kulman muutos, on keskeistä hyvän kalibrointituloksen saavuttamiseksi.*

### 3.3 Ohjelmisto

Kalibrointiohjelmisto voidaan jakaa kolmeen osaan: kalibrointiohjelmiin, virheanalyysiin ja ohjelmaan, jolla saatu kalibrointitulokset toimeenpannaan. Tavoitteena on saavuttaa mahdollisimman tarkka ja luotettava kalibrointitulokset. Lisäksi haluttiin muuntaa saadut tulokset reaali maailman arvoihin, jolloin

virheanalyysi ja kalibroititulos on helpompi käsittää myös suppeammalla taustatiedolla. Lopullinen ohjelmisto on helppokäyttöinen kalibroitipaketti, jolla amatööri voi saada luotettavia tuloksia ja perehtyä kamerakalibraation sovellutuksiin, kuten linssivääristymän poistoon, virheanalyysiin ja reaali maailman objektien mittaukseen kuvasta.

Kamerakalibraation kannalta keskeisin ohjelma, calibration, on ohjelmiston keskiössä. Calibration-ohjelma ottaa argumenteikseen kalibroitaineiston, suorittaa niille kulmanmäärittelyn sekä kertoo käyttäjälle, mikäli kulmanmäärittelyä ei voitu suorittaa. Ohjelma tarjoaa käyttäjälle visuaalisen palautteen kalibroinnin joka vaiheelta, joten käyttäjä voi tarkistaa tulokset ohjelmiston virheiden varalta. Olennaisia ennakkotietoja ohjelman käyttöönoton kannalta ovat ruudukon koko ja käsiteltävien kuvien tiedostonimi.

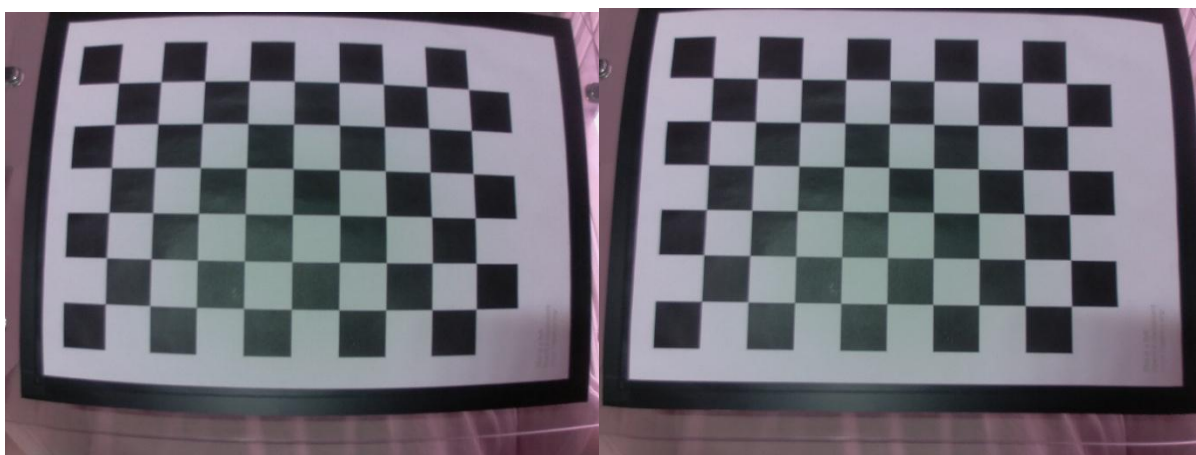
Ohjelma ei määritä ruudukon kokoa automaattisesti, jolloin mahdollinen kompastuskivi voi olla väärin määritetyt ruudukon parametrit. Tämän virheen kannalta on tärkeää saada visualisointi kulmanmäärittelyn jälkeen. Vaikka kuvasta tunnistettaisiin pyydetty ruudukko, voi pyydetty ruudukko olla määritetty liian pieneksi, jolloin kalibrointi saadaan suoritettua, mutta sen tarkkuus kärsii, kun kaikkea saatavilla olevaa dataa ei hyödynnetä. Ruudukon koon manuaalinen määrittely mahdollistaa kuitenkin laajemman kalibroitimateriaalin käytön sillä kuvat joissa ruudukko ei näy kokonaisuudessaan, voidaan hyödyntää pienentämällä ruudukon kokoa niin, että se vastaa kuvassa näkyvää osuutta ruudukosta.

Calibration-ohjelma suorittaa kalibroinnin kolmivaiheisesti. Ensimmäisenä ohjelma lataa käsiteltävän kuvan ja muuntaa sen harmaasävyiseksi ruudukon erottuvuuden varmistamiseksi. Tämän jälkeen suoritetaan automaattinen kulmanmäärittely annettujen ruudukkoparametrien mukaan ja tallennetaan välivaiheena kuva, jossa on visualisoitu kulmanmäärittely. Mikäli ohjelma ei kykene tunnistamaan ruudukkoa se ilmoittaa käyttäjälle, että ruudukkoa ei löydetty ja siirtyy automaattisesti seuraavan kuvan käsittelyyn.

Toisessa vaiheessa suoritetaan varsinainen kalibrointi. Kalibrointi suoritetaan openCV calibrateCamera-funktiolla. CalibrateCamera-funktio käyttää argumentteinaan edellisen vaiheen kulmanmäärittelyssä luotuja pisteitä ja laskee kameran sisäiset parametrit vertaamalla kuvatason kaksiulotteista koordinaatistoa reaali maailman koordinaatistoon. Tämän jälkeen saatu kalibrointidata tallennetaan erilliseen .npz tiedostoon, jotta kalibrointidataa voidaan hyödyntää myöhemmin. Erillisen kalibrointidatan käyttö mahdollistaa kalibroitituloksen käyttämisen myös kuvissa, joissa ei näy kalibroitiruudukkoa. Koska kalibrointidata on kamerasta riippuvainen, voidaan kalibrointidataa käyttää vain siihen kameraan, jolla alkuperäinen kalibrointi on suoritettu. Kameran sisäisten parametrien

lisäksi tässä vaiheessa lasketaan keskimääräinen virhe, jossa kulmanmääritys algoritmin luomia pisteitä verrataan reaali maailman pisteisiin ja niiden välinen ero määritetään pikseleinä.

Viimeisessä vaiheessa varsinainen kalibrointi on jo suoritettu, mutta tuloksen visualisoinniksi halutaan kuvasta luoda myös versio, jossa linssivääristymä on poistettu. Linssivääristymä poistetaan undistort-funktiolla, joka käyttää luotua ideaalimatriisia, joka saadaan openCV kirjaston `getOptimalNewCameraMatrix`-funktiolla. Funktio laskee kameraparametrien avulla kuvan koosta riippuvaiset ideaaliparametrit. Ideaaliparametreilla luotu vääristymätön kuva tarvitsee vielä leikata, sillä kameran kupera linssi luo vääristymän, jossa kuva on ikään kuin keskeltä pullistunut. Kun vääristymä poistetaan, syntyy kuvalle koverat reunat. Lopuksi reunat leikataan pois, jolloin saadaan tavanomainen vääristymäkorjattu kuva ilman puuttuvia reunoja, katso kuvat 2,5 ja 7.



*Kuva 5: Vasemmalla alkuperäinen kuva. Oikealla kalibroinnin lopputuotteena saatu kuva, jossa linssivääristymä on korjattu.*

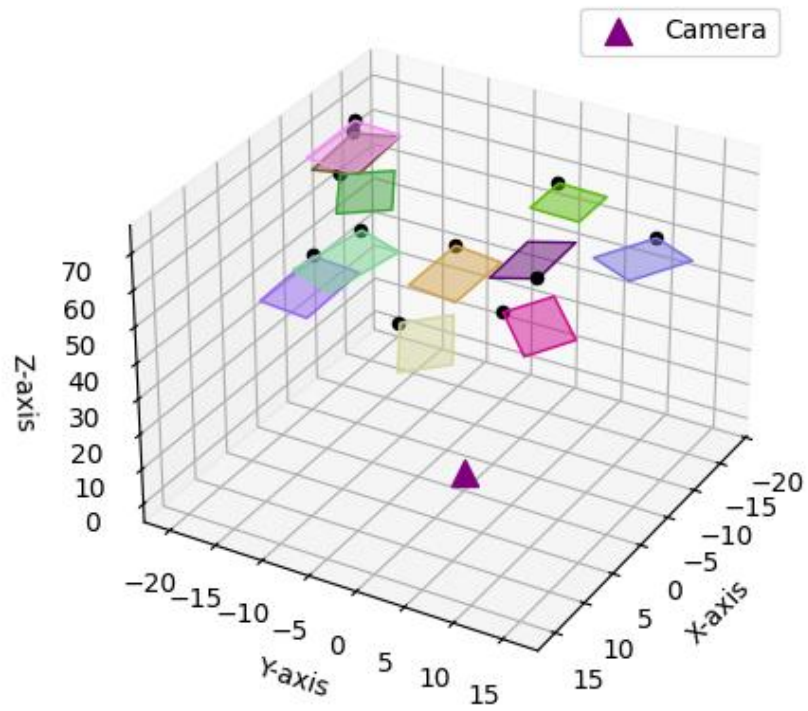
### 3.4 Virheanalyysi ja apuohjelmat

Kalibrointiohjelmisto sisältää työkalut myös virheanalyysiin. Virheanalyysi suoritetaan kolmiosaisesti, jolloin saadaan analysoitua keskimääräinen uudelleenprojektivirhe, reaali maailman virhe määrätyllä etäisyydellä sekä pikselin koko kalibrointiruudun luomalla tasolla. Ohjelmisto laskee keskimääräisen virheen automaattisesti `calibration`-ohjelman avulla, jossa keskimääräinen virhe lasketaan openCV:n `norm` funktiolla. Funktion antamat tulokset eri kuville lasketaan yhteen, ja niistä lasketaan keskiarvo. Perinteisen keskiarvolaskennan sijaan voitaisiin käyttää myös juurettua keskiarvoa, jossa arvojen neliöt lasketaan yhteen ja lopulta jaettu keskiarvo lasketaan neliöjuurella. Juurettu keskiarvo on yleisesti mielletty tarkemmaksi menetelmäksi, sillä se korostaa poikkeuksellisen suuria ja pieniä arvoja verrattuna tavanomaiseen keskiarvon laskutapaan. Esimerkissä käytetty kuvien määrä on kuitenkin verrattain pieni ja kuvien luomat yksittäiset uudelleenprojektivirheet ovat hyvin toisiaan vastaavat, joten päädyttiin käyttämään yksinkertaisempaa laskutapaa.

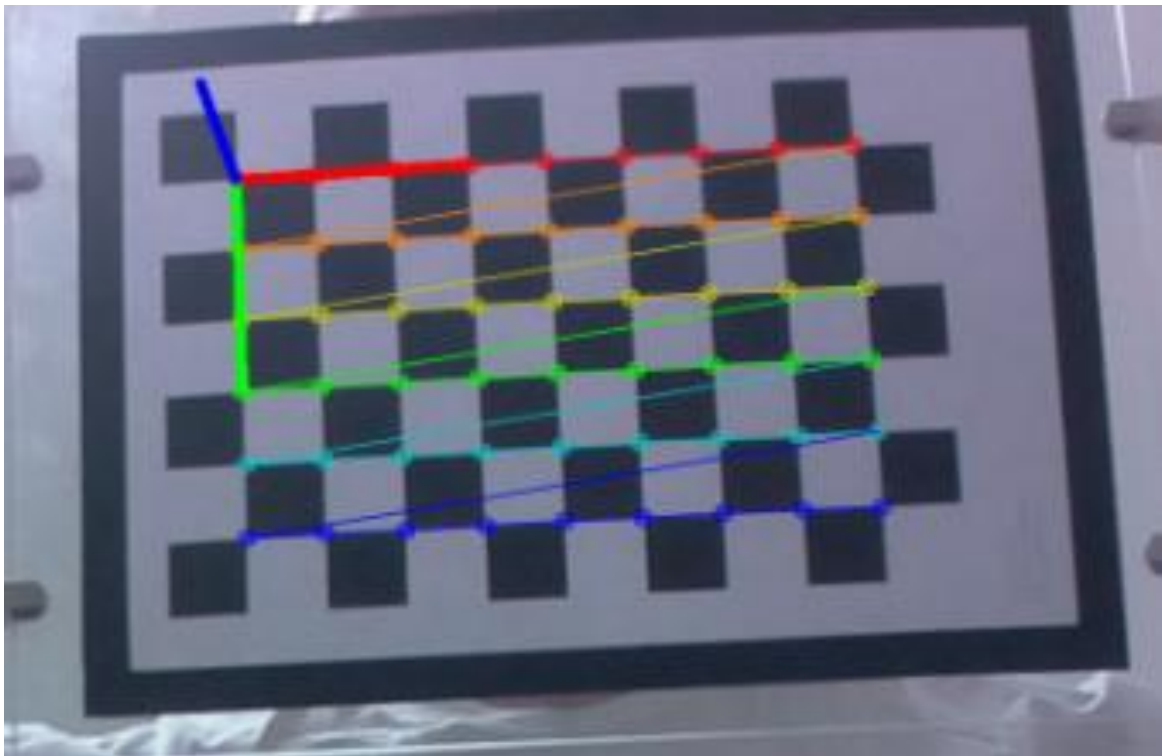
Uudelleenprojektivirheen laskennan lisäksi ohjelmistopaketti tarjoaa työkalun, jolla voidaan laskea pikselin koko, kun ruudukon koko on tunnettu. Ohjelmisto laskee ruudukon ruutujen keskimääräisen koon pikselinä ja jakaa ruudun koon saadulla arvolla. Näin lopulliseksi arvoksi saadaan pikselin koko millimetreinä. Huomioitavaa on kuitenkin se, että arvoa ei voida hyödyntää objekteihin, jotka ovat eri etäisyydellä kameraan nähden, vaan tarkan tuloksen varmistamiseksi tulisi kalibroitiruudukko kiinnittää mitattavaan objektiin. Tämä on yksi monokulaarisen kamerajärjestelmän heikkouksista, sillä järjestelmä ei kykene arvioimaan syvyyttä. Tämä johtaa kaukaisten objektien koon pienenemiseen mitattaessa ja yhtäläisesti objektit lähempänä kameraa vaikuttavat suuremmilta, mikäli ne mitataan pikselien avulla.

Etäisyyden arvioinnin tueksi on luotu työkalu, joka arvioi keskimääräistä virhettä halutulla etäisyydellä. Tämä työkalu on kuitenkin kyseenalainen, sillä se pohjaa laskentansa ilmoitettuihin kameraparametreihin. Erityisesti edullisen kameran kanssa työskennellessä käyttäjä ei välttämättä voi luottaa valmistajan antamiin mittoihin polttoväliin ja sensorin kokoon liittyen. Työkalua voidaan siis käyttää ohjeellisena arvona, mutta sen tarkkuudesta ei ole vahvaa näyttöä. Työkalu laskee virheen etäisyydellä vertaamalla sensorin tai kennon kokoa kuvan resoluutioon. Tämän jälkeen keskimääräinen uudelleenprojektivirhe voidaan muuntaa reaali maailman virheeksi halutulla etäisyydellä. Laskutapa perustuu kameraoptiseen malliin, joka esitellään kirjassa *Multiple View Geometry in Computer Vision*. [17]

Virheanalyysityökalujen tueksi luotiin myös visualisointi- ja tarkastuskeinoja, joilla käyttäjä voi tarkastella ohjelmiston luomaa dataa. Tällaisia työkaluja ovat `pose_estimation_*` työkalut, joilla käyttäjä saa visualisoitua kalibroitiruudukon luoman tason kameraan nähden sekä luodussa koordinaatistossa että reaali maailman kuvissa, (kuvat 6 ja 7). Nämä työkalut toimivat niiden manuaalisten virheanalyysimenetelmien tukena, joita `calibration`-ohjelmisto tuottaa jokaisessa kalibroitivaiheessa.



Kuva 6: Kalibroitidatan avulla luodut takaisinprojektiotasot kolmiulotteisessa koordinaatistossa. Värilliset tasot vastaavat kalibroitiruudukon sijaintia ja kuvakulmaa kameraan nähden.



Kuva 7: Pose estimation -kuva reaali maailmassa, kalibroitiruudukolle luotu origo visualisoi ruudukon kulman kameraan nähden.

## 4 Yhteenveto

Tutkielmassa käydään läpi kamerakalibraatioon liittyviä tieteellisiä julkaisuja ja tutustutaan laajemmin yhden kameran järjestelmän kalibrointiin Zhangin menetelmällä. Kamerakalibraatiossa keskeistä on kameran ulkoisten ja sisäisten parametrien selvittäminen, joihin tutkielma tarjoaa teoriaa sekä käytännön toteutuksen. Tavoitteena on että amatööritutkija voi itsenäisesti suorittaa luotettavaa kamerakalibraatiota tutkielman tiedoilla.

Luvussa 2 tutustutaan kamerakalibraation teoriaan ja käydään läpi erilaisia menetelmiä, joista syvennytään tarkemmin shakkilautamenetelmään. Menetelmän tueksi käydään läpi keskeiset taustatiedot neulanreikäkameramallista ja takaisinprojektion laskennasta. Tutkielman pohjana toimii ilmainen OpenCV konenäkökirjasto, mutta siinä sivutaan myös muita mahdollisia kalibrointiympäristöjä. OpenCV on valikoitu tutkielman pohjaksi laajan dokumentaation, sekä hyvän saatavuuden vuoksi. Luvussa syvennytään myös kalibroinnin kannalta keskeiseen virheanalyysiin, sekä siihen miten varsinainen kalibrointi suoritetaan.

Luvussa 3 aineistona kalibroinnin suorittamiseksi tarjotaan työkalut, jotka on saatavilla tutkielman kirjoittajan gitlab-sivuilta, sekä toteutetaan esimerkkisuorite kalibroinnista, joka käydään läpi vaiheittain. Tavoitteena on luoda vakaa ymmärrys siitä, mitä kalibrointiprosessissa tapahtuu kussakin vaiheessa ja selvittää kuinka ohjelmisto toimii ja miten sitä voidaan käyttää luotettavan kalibraation saavuttamiseksi. Kalibrointiohjelmiston lisäksi listataan minimivaatimukset kalibroinnin suorittamiselle, sekä hyviä toimintamalleja luotettavan kalibrointidatan luomiseksi.

Lopputuloksena lukija on perehdytetty nykypäivän suosituimpaan kalibrointimenetelmään, joka voidaan saavuttaa yksinkertaisella kalustolla. Keskeisessä roolissa onkin kamerakalibraation saavutettavuus suuremmalle yleisölle käyttäen tarvikkeita, jotka lähes jokainen omistaa valmiiksi, tai voi hankkia mahdollisimman pienillä investoinneilla, tai saavuttaa hyödyntäen esimerkiksi julkisesti saatavilla olevia laitteita.

## Viitteet

- [1] D.-Y. Ge, X.-F. Yao, W.-J. Xiang, E.-C. Liu, and Y.-Z. Wu, "Calibration on camera's intrinsic parameters based on orthogonal learning neural network and vanishing points," *IEEE Sensors Journal*, vol. 20, nro. 20, s. 11755–11763, Loka. 2020.
- [2] L. Zhao, C. Wu, and J. Ning, "A camera calibration method based on two orthogonal vanishing points," *Concurrency Computat.: Pract. Exper.*, vol. 26, nro. 5, s. 1032–1043, 2014.
- [3] A. de la Escalera and J. M. Armingol, "Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration," *Sensors*, vol. 10, nro. 3, s. 2027–2044, Maalis. 2010.
- [4] L. Zhang et al., "Implementation of camera calibration method based on OpenCV," *Applied Mechanics and Materials*, vol. 602–605, s. 3796–3799, 2014.
- [5] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, UK: Cambridge University Press, 2003.
- [6] J. Wohlfeil, D. Griebßbach, I. Ernst, D. Baumbach, and D. Dahlke, "Automatic camera system calibration with a chessboard enabling full image coverage," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W13, s. 1715–1722, 2019.
- [7] J. Heikkilä, "Geometric camera calibration using circular control points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, nro. 10, s. 1066–1077, Loka. 2000.
- [8] Optical Metrology Centre, "Handbook of practical camera calibration methods and models," 2003.
- [9] C. Wöhler, *3D Computer Vision: Efficient Methods and Applications*. Hoboken, NJ, USA: Wiley, s. 15-51 2012.
- [10] R. He, Y. Zhuang, B. Fu, and F. Liu, "Point Spread Function Estimation of Defocus," *arXiv preprint arXiv:2203.02953*, 2022.

- [11] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *Proc. International Conference on Computer Vision Theory and Applications (VISAPP)*, Funchal, Madeira, Portugal, Tammi. 2008, s. 250-257.
- [12] OpenCV, "Camera calibration," OpenCV Documentation, 2024.
- [13] A. Hagemann, M. Knorr, H. Janssen, and C. Stiller, "Inferring Bias and Uncertainty in Camera Calibration," *arXiv preprint arXiv:2107.13484*, 2021.
- [14] A. Hagemann, M. Knorr, H. Janssen, and C. Stiller, "Bias detection and prediction of mapping errors in camera calibration," *arXiv preprint arXiv:2107.13484*, 2021.
- [15] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, nro. 11, s. 1330–1334, Marras. 2000.
- [16] MathWorks, "Evaluating the Accuracy of Single Camera Calibration," *MathWorks Documentation*, 2024.
- [17] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed. Cham, Sveitsi: Springer, 2022, doi: 10.1007/978-3-030-34372-8.
- [18] Seeed Studio, "OV5647-75 FOV Camera module for Raspberry Pi 3B+4B", gitlab.utu.fi, ovverj, kamerkalibraatio/kameran\_valmistustiedot, 2025