

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Ernest Beličič

**Večkamerni sistem za lokalizacijo  
objekta v prostoru**

DIPLOMSKO DELO  
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Danijel Skočaj

Ljubljana 2015



Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogu:

Tematika naloge:

Besedilo teme diplomskega dela študent prepiše iz študijskega informacijskega sistema, kamor ga je vnesel mentor. V nekaj stavkih bo opisal, kaj pričakuje od kandidatovega diplomskega dela. Kaj so cilji, kakšne metode uporabiti, morda bo zapisal tudi ključno literaturo.



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Ernest Beličič sem avtor diplomskega dela z naslovom:

*Večkamerni sistem za lokalizacijo objekta v prostoru*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Danijela Skočaja,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 14. september 2015

Podpis avtorja:



*Hvala vsem.*



Hčerki Leneji.



# Kazalo

## Povzetek

## Abstract

<b>1 Uvod</b>	<b>1</b>
1.1 Motivacija . . . . .	1
1.2 Pregled področja . . . . .	2
1.3 Cilji . . . . .	3
1.4 Struktura diplome . . . . .	4
<b>2 Teoretična podlaga</b>	<b>5</b>
2.1 Parametri kamere . . . . .	5
2.2 Ocenjevanje parametrov . . . . .	12
2.3 Triangulacija . . . . .	16
2.4 Zaznavanje označevalnika . . . . .	16
<b>3 Implementacija</b>	<b>19</b>
3.1 Strojna oprema . . . . .	19
3.2 Programska oprema . . . . .	23
3.3 Kalibracija kamer . . . . .	24
3.4 Implementacija zaznavanja označevalnika . . . . .	33
3.5 Postavitev sistema . . . . .	34
<b>4 Rezultati</b>	<b>37</b>
4.1 Reprojekcija kalibracijskih točk . . . . .	37

*KAZALO*

4.2 Natančnost zaznavanja označevalnika . . . . .	38
4.3 Natančnost pozicioniranja . . . . .	43
<b>5 Zaključek</b>	<b>47</b>

# Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>GPS</b>	global positioning system	sistem za globalno pozicioniranje
<b>SLAM</b>	simultaneous localization and mapping	hkratno lokaliziranje in kartiranje
<b>IPS</b>	indoor positioning system	sistem za pozicioniranje v notranjih prostorih
<b>TDOA</b>	time difference of arrival	časovna razlika prihoda
<b>CCD</b>	charge-coupled device	nabojno povezana naprava
<b>DLT</b>	direct linear transformation	neposredna linearna transformacija
<b>SVD</b>	singular value decomposition	razcep na singularne vrednosti
<b>RGB</b>	red, green, blue	rdeča, zelena, morda
<b>HSV</b>	hue, saturation, value	ton, nasičenost, intenziteta
<b>HSL</b>	hue, saturation, lightness	ton, nasičenost, svetlost
<b>CGI</b>	common gateway interface	vmesnik za skupni dostop
<b>IP</b>	internet protocol	internetni protokol
<b>HTTP</b>	hypertext transfer protocol	protokol za prenos hiperteksta
<b>LED</b>	light-emitting diode	svetleča dioda
<b>PTZ</b>	pan, tilt, zoom	obrat, nagib, povečava



# Povzetek

V vzorcu je predstavljen postopek priprave diplomskega dela z uporabo okolja L<sup>A</sup>T<sub>E</sub>X. Vaš povzetek mora sicer vsebovati približno 100 besed, ta tukaj je odločno prekratek.

**Ključne besede:** lokalizacija, triangulacija, računalniški vid, kalibracija kamer.



# Abstract

This sample document presents an approach to typesetting your BSc thesis using L<sup>A</sup>T<sub>E</sub>X. A proper abstract should contain around 100 words which makes this one way too short.

**Keywords:** localization, triangulation, computer vision, camera calibration.



# Poglavlje 1

## Uvod

### 1.1 Motivacija

V robotiki je eden izmed glavnih problemov lokalizacija robota v prostoru. Beseda robot ima dandanes zelo širok pomen in lahko zajema naprave od avtomobilov na daljinsko upravljanje, do popolnoma avtonomnih kvadrokopterjev. Z besedami lokacija, pozicija in položaj, mislim na točke v 3-dimenzionalnem prostoru (svetu). Za določanje položaja obstaja veliko rešitev, ki pa se razlikujejo po natančnosti, težavnosti in ceni implementacije. Če nas zanima kje na Zemlji se robot približno nahaja, bomo najverjetneje uporabili GPS (*ang. Global Positioning System*). Prednost tega sistema je robustnost saj je v uporabi že od leta 1978 [10], pomanjkljivost pa premajhna natančnost, če želimo izmeriti položaj do centimetra ali milimetra natančno. GPS deluje slabše v zaprtih prostorih, saj strehe in stene odbijejo precejšen del mikrovalovnih signalov. V primeru, da je naš robot opremljen s kamero za zaznavanje globine pa lahko uporabimo tehniko SLAM (*ang. Simultaneous Localization and Mapping*). Pri tej tehniki poleg položaja dobimo tudi kartu okolja.

Sam sem se osredotočil razviti sistem za določanje položaja v notranjem prostoru (*ang. IPS - Indoor Positioning System*). Taki sistemi morajo biti natančni vsaj na nekaj decimetrski lestvici, odvisno od namena uporabe.

Glavna motivacija za temo diplomskega dela je sistem kvadrokopterjev, ki so ga razvili na ETH Zurich pod okriljem prof. Raffaella D'Andree [3]. Za zaznavanje položaja uporabljajo enega od sistemov za zaznavanje gibanja (*ang. motion capture system*), ki pa so lahko zelo dragi. Že najcenejši sistem z le dvema kamerama se giblje nad 1000 EUR.

Ker so natančni sistemi za določanje položaja v notranjem prostoru dragi, vendar nujni za avtonomno upravljanje kvadrokopterja v prostoru, sem v sklopu diplomske naloge razvil večkamerni sistem za določanje lokacije objekta v prostoru. Objekt je, tako kot v komercialnih sistemih, barvni označevalnik v obliki krogle.

## 1.2 Pregled področja

Veliko sistemov za pozicioniranje v prostoru ne uporablja kamere, ampak druge vrste oddajnikov/sprejemnikov s katerimi se lahko izmeri razdalja do objekta, nato pa s pomočjo trilateracije [13] ali multilateracije [11] izračuna pozicija objekta. Te sistemi delujejo po enakem principu. V prostor postavijo t.i. sidra z znanim položajem. To so lahko ultrazvočni ter radijski oddajniki ali sprejemniki, mikrofoni itd. Sidra služijo kot referenčne točke v prostoru s katerimi se izračuna razdalja ali pa razlike razdalj do objekta zanimanja. Ko so razdalje ali razlike razdalj znane, se lahko uporabi prej omenjena postopka za izračun pozicije objekta.

Na Univerzi v Kaliforniji so razvili sistem, ki izračuna pozicijo s pomočjo zvoka [7]. Za sidra so uporabili mikrofone, objekt zanimanja pa je bil mobilni telefon. V 97 % poskusov so položaj izmerili do 50 cm natančno. Na Univerzi v Yorku so naredili podobno z Wi-Fi signali in znanim modelom zaprtega prostora [2]. Na Univerzi v Parani pa so za merjenje razdalje uporabili ultrazvočni signal [1]. Natančnost njihovega sistema je nekaj desetink milimetra v  $7 \times 7 \times 7 m^3$  velikem prostoru.

Do sedaj opisana dela ne uporabljajo kamer za določanje položaja objekta. V nadaljevanju pa se bom osredotočil na dela, ki jih uporabljajo. Namesto

o trilateraciji [13] ali multirateraciji [11], pri računanju položaja s kamerami, govorimo o triangulaciji [12]. Slika iz navadne kamere nam sama po sebi ne pove razdalje do neke točke ali objekta, vendar le smer žarka, ki pa implicitno določa kote s katerimi lahko izračunamo razdaljo do objekta. Za to tehniko potrebujemo vsaj dve kamери ali pa vsaj dve slike, ki opazujeta isti objekt iz različnih zornih kotov.

Na Univerzi v Xi'an Jiaotong so razvili stereo sistem za določanje položaja planarne tarče [6]. Uporabili so dve CCD [9] kamери, ki hkrati zajameta sliko ploskve in nato iz dobjenih točk triangulirajo pozicijo ploskve v prostoru. Na Tehnični univerzi v Madridu so uporabili večkamerni sistem za določanje položaja ljudi v prostoru [8]. Najprej segmentirajo vse premikajoče dele slike od statičnih, saj predpostavljajo, da se bodo ljudje po prostoru premikali. Nato na teh segmentih zaznajo kje se nahajajo ljudje. Največji problem predstavlja okluzija. Pri tem jim pomaga ravno večkamerni sistem, ki zajema slike ljudi iz različnih zornih kotov. Ko določijo kje na slikah se nahaja glava človeka pa lahko izračunajo njegov položaj v prostoru. Na Državni univerzi v Ohiju pa so naredili sistem [5], ki je, od opisanih, najbolj soroden mojemu delu. V prostor so postavili štiri visokoločljive omrežne kamere, s ciljem izračunati položaj nekega orodja. Kamere so med seboj kalibrirane. Od štirih morata vsaj dve kamери videti orodje, da se lahko določi njegov položaj. Pred sledenjem objekta, ga morajo najprej zaznati. To naredijo z ujemanjem predlog. S sledenjem oz. zaznavanjem dobijo le približno sredino objekta, ki pa jo z večimi kamerami izpopolnijo.

### 1.3 Cilji

Glavni cilj diplomskega dela je ustvariti večkamerni sistem za določanje položaja objekta v prostoru. Pri zaznavanju bo lahko prihajalo do okluzij, zato je minimalna zahteva, da vsaj dve kamери v nekem trenutku vidita objekt. Za obdelavo slik se bo uporabil centralni računalnik, ki bo s kamerami povezan v lokalno omrežje. Zaradi lažjega zaznavanja, bo objekt zanimanja

barvni označevalnik. Napaka sistema mora biti v povprečju manjša od 5 cm. Določanje položaja bo delovalo v realnem času z vsaj 10 meritvami na sekundo. Glavna omejitev sistema bo ta, da bo deloval ob predpostavki, da se v prostoru nahaja le en objekt, kar onemogoča določanje orientacije npr. robota. To omejitev pa se bo lahko v prihodnosti odstranilo in bo sistem zmožen hkrati določiti položaj veliko točkam v prostoru. Sistem bo moral omogočati postavitev kamer, ki med seboj nimajo preseka vidnega polja.

## **1.4 Struktura diplome**

Diplomsko delo je razdeljeno na 5 poglavij. V uvodu je predstavljena motivacija, pregled področja in cilji dela. Drugo poglavje zajema teoretično podlago, ki je nujna za kasnejšo implementacijo rešitve. Tretje poglavje opisuje dejansko implementacijo rešitve in konkretno uporabo prej opisane teorije. V četrtem poglavju so predstavljeni rezultati dela. Zadnje poglavje pa poleg zaključka naslovi tudi možne izboljšave sistema.

# Poglavlje 2

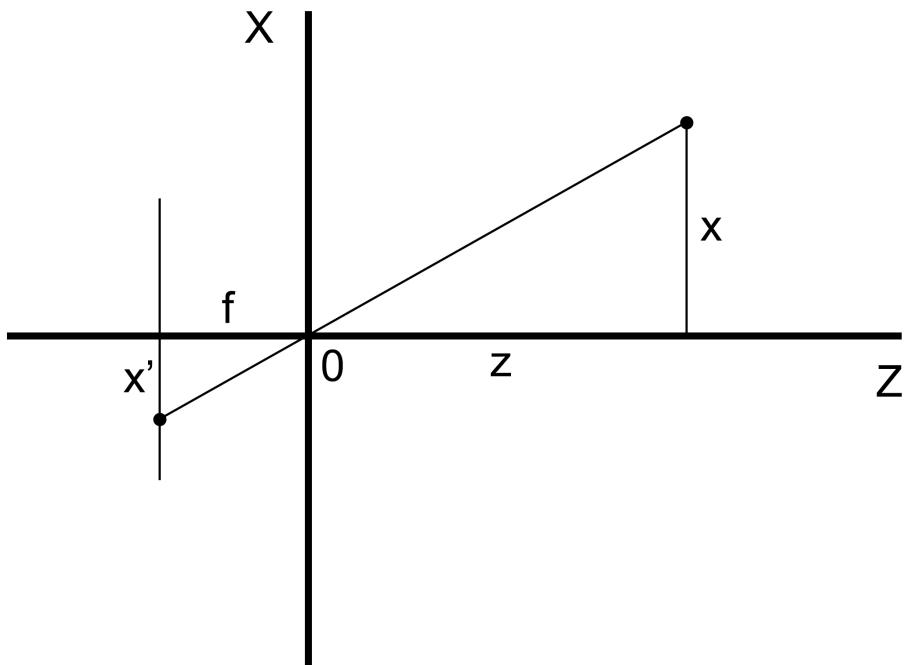
## Teoretična podlaga

### 2.1 Parametri kamere

Ugotoviti moramo kakšen matematični model najbolj ustreza današnjim kamерam. Večino kamer lahko modeliramo z modelom kamere z luknjico (*ang. pinhole camera*). Model mora opisovati kako se neka točka v svetu preslika na točko v sliki. Model lahko opišemo s t.i. notranjimi in zunanjimi parametri kamere. V grobem, notranji opisujejo interne lastnosti kamere kot je goriščna razdalja, principalna točka, popačenost itd., zunanji pa položaj kamere v prostoru.

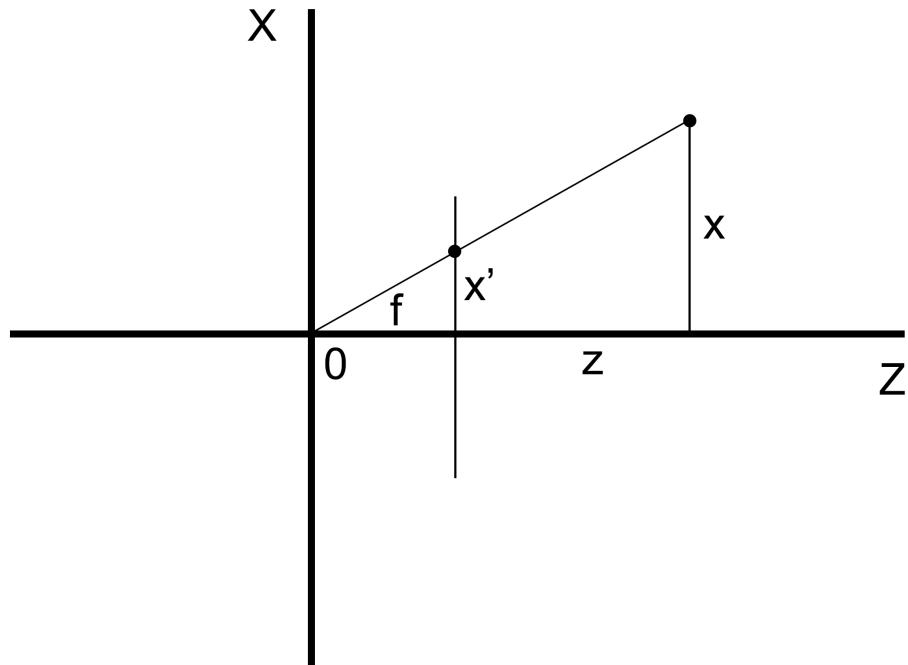
#### 2.1.1 Notranji parametri

Na sliki 2.1 je ponazorjena preslikava točke v svetu na točko v sliki. Ugotovitve v eni dimenziji lahko apliciramo tudi na drugo dimenzijo.



Slika 2.1: Točka v svetu (desno) se preslika na slikovno ploskev (levo).

Izhodišče predstavlja luknjico kamere. Višina točke v svetu je predstavljena z  $x$ , oddaljenost od kamere pa z  $z$ . Točka na sliki je označena z  $x'$ , goriščna razdalja pa s  $f$ . Točka v svetu se preko luknjice (koordinatnega izhodišča) preslika na slikovno ploskev. Vse točke v svetu so na slikovni ploskvi obrnjene za  $180^\circ$ , kar pa kamere samodejno popravijo, ko vrnejo sliko. Za lažje računanje pa lahko to ugotovitev uporabimo za vpeljavo navidezne slikovne ploskve.



Slika 2.2: Na sliki se jasno vidi dva podobna pravokotna trikotnika.

Hitro opazimo, da je preslikava iz sveta na sliko linearна operacija. Na sliki 2.2 vidimo podobna pravokotna trikotnika, ki ju določa točka v svetu in točka na sliki. Vzpostavimo lahko relacijo,

$$\frac{x'}{f} = \frac{x}{z} \quad (2.1)$$

$$x' = f * \frac{x}{z} \quad (2.2)$$

Če formulo posplošimo na obe koordinatni osi dobimo,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \frac{f}{z} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.3)$$

Enačba (2.3) predstavlja najpreprostejši model kamere z luknjico. Ima seveda veliko pomanjkljivost, ki pa jih bomo postopoma odpravili. Enačbo

(2.3) se da zapisati še preprosteje z uvedbo homogenih koordinat.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} f * x \\ f * y \\ z \end{bmatrix} \quad (2.4)$$

Točka v homogenih koordinatah ni enolično določena z vrednostmi točke saj za homogeno točko  $x$  velja  $x \sim \lambda x$ , kjer  $\lambda \neq 0$ . Homogeno točko preslikamo nazaj v evklidski prostor tako, da vse koordinate delimo z zadnjo vrednostjo točke.

Pri predstavitvi digitalnih slik je izhodišče običajno v levem zgornjem kotu. Zgornji model pa predpostavlja izhodišče v sredini slike. V model moramo vpeljati dve konstanti  $u$  in  $v$ , ki bosta prestavili izhodišče koordinatnega sistema slike. To se lahko kompaktno zapiše kot množenje točke v svetu z matriko.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & u \\ 0 & f & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.5)$$

Zgornji model predpostavlja, da so pike na senzorju kamere kvadratne. Dandanes je za veliko kamer to tudi res. Ker pa se da pravokotne ali celo poševne pike enostavno vključiti v obstoječi model, bomo to naredili.

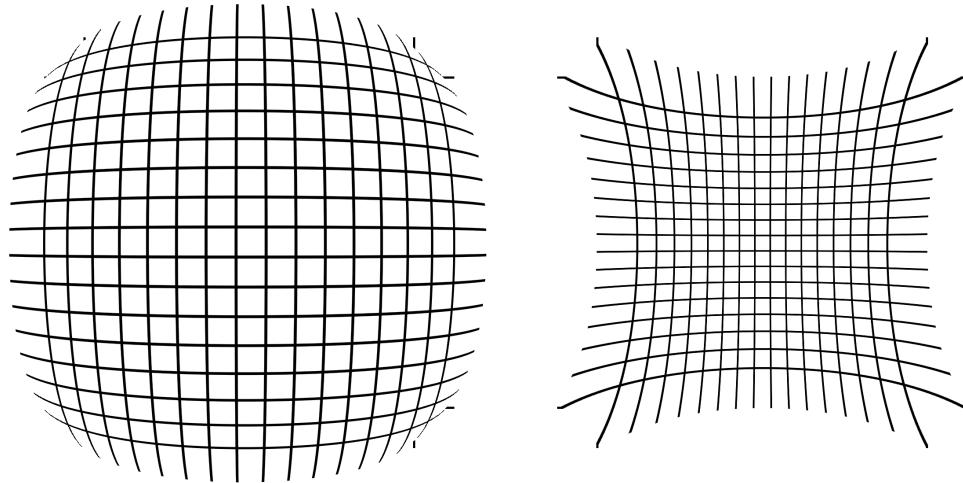
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} f * m_x & s & u \\ 0 & f * m_y & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.6)$$

Enačba (2.6) predstavlja notranji model kamere. Goriščna razdalja je označena s  $f$ ,  $m_x$  in  $m_y$  predstavljata velikost,  $s$  pa poševnost pike,  $u$  in  $v$  pa določata principalno točko. Vektor  $[x \ y \ z]^T$  določa točko v svetu,  $[x' \ y' \ 1]^T$  pa kam se ta točka preslika na sliko.

Slike iz kamer so lahko tudi popačene. Obstajata dve vrsti popačenosti, ki ju prej opisani model ne more modelirati. To sta radialna in tangencialna popačenost. Radialno popačenost povzroči oblika leče in poznamo dve glavni različici:

1. sodčasto, ki spominja na obliko soda in
2. blazinasto, ki spominja na obliko blazine

Obstaja še kombinacija obeh, ki pa se imenuje brkato popačenje (ker spomina na obliko brk).



Slika 2.3: Sodčasta popačenost (levo), blazinasta popačenost (desno).

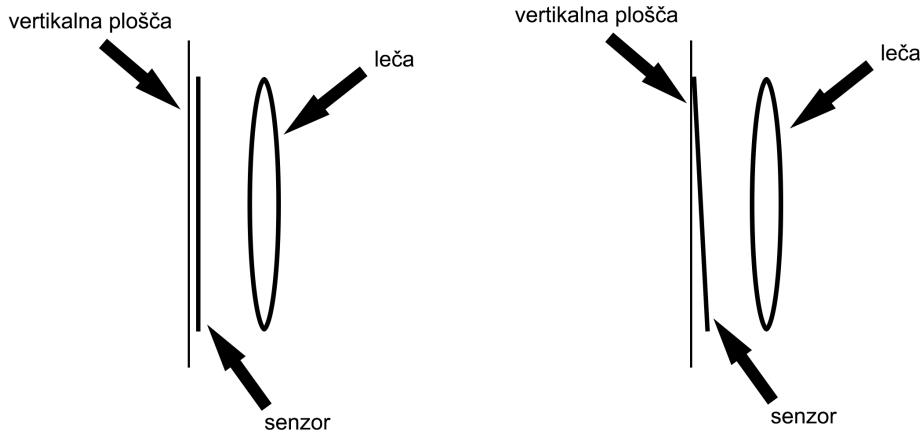
Moč radialne popačenosti slikovne točke je odvisna od razdalje do principalne točke. Modeliramo jo lahko z vsotami polinomov sodih stopenj. Za večino radialnih popačenj zadoščata že dva koeficienta.

$$r = \sqrt{x^2 + y^2} \quad (2.7)$$

$$x_{popaceno} = x * (1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6 + \dots) \quad (2.8)$$

$$y_{popaceno} = y * (1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6 + \dots) \quad (2.9)$$

Druga popačenost, ki jo poznamo pa je tangencialna in nastane zaradi slabe poravnanoosti leč in senzorja.



Slika 2.4: Senzor in leča sta vzporedna zato do tangencialne popačenosti v tem primeru ne pride (levo).

Modeliramo jo lahko kot,

$$r = \sqrt{x^2 + y^2} \quad (2.10)$$

$$x_{popaceno} = x + (2 * p_1 * x * y + p_2 * (r^2 + 2 * x^2)) \quad (2.11)$$

$$y_{popaceno} = y + (p_1 * (r^2 + 2 * y^2) + 2 * p_2 * x * y) \quad (2.12)$$

Če povzamemo, notranje parametre kamere določa matrika A in koeficienti popačenja. Koeficiente se lahko predstavi z vektorjem  $\vec{r}$ , ki določa radialno popačenost in z vektorjem  $\vec{t}$ , ki določa tangencialno popačenost.

$$A = \begin{bmatrix} f * m_x & s & u \\ 0 & f * m_y & v \\ 0 & 0 & 1 \end{bmatrix}$$

$$\vec{r} = [k_1 \ k_2 \ k_3 \dots]$$

$$\vec{t} = [p_1 \ p_2]$$

### 2.1.2 Zunanji parametri

Do sedaj smo predpostavljeni, da je koordinatno izhodišče sveta sama kamera. Uporabno bi bilo, če bi lahko poljubno določili koordinatni sistem sveta in vanj postavili kamere. Ravno temu so namenjeni zunanji parametri kamere. Med različnimi koordinatnimi sistemi lahko enolično prehajamo z rotacijo in translacijo. Če napišem malo drugače: iz enega koordinatnega sistema lahko dobimo kateri koli drug koordinatni sistem (z enakim št. dimenzij) tako, da izhodišči poravnamo s translacijo in nato rotiramo osi, da sovpadajo.

Rotacijo v treh dimenzijah lahko predstavimo z množenjem matrike  $R$  velikosti  $3 \times 3$ . Translacija pa je vsota točke v svetu  $\vec{X}$  in translacijskega vektorja  $\vec{T}$ .

$$\vec{X}_{\text{premaknjen}} = R * \vec{X} + \vec{T} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} \quad (2.13)$$

Izkaže se, da če točko v svetu predstavimo s homogenimi koordinatami, lahko enačbo (2.14) zapišemo kot množenje matrike  $[R|\vec{T}]$  z vektorjem  $\vec{X}$ .

$$\vec{X}_{\text{premaknjen}} = [R|\vec{T}] * \vec{X} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x_t \\ r_{21} & r_{22} & r_{23} & y_t \\ r_{31} & r_{32} & r_{33} & z_t \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.14)$$

Zavedati se moramo, da rotacijska matrika  $R$  in translacijski vektor  $\vec{T}$  ne predstavlja rotacijo in pozicijo kamere v svetu, ampak rotacijo in pozicijo sveta relativno na kamero. Rotacijo kamere lahko dobimo z  $R^{-1} = R^T$ , pozicijo v svetu pa z  $-R^{-1} * \vec{T} = -R^T * \vec{T}$ .

### 2.1.3 Model kamere

Model (2.6) lahko dopolnimo z zunanjimi parametri ter tako dobimo popolen model kamere.

$$\vec{X}' \sim A * [R|\vec{T}] * \vec{X} \quad (2.15)$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} f * m_x & s & u \\ 0 & f * m_y & v \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} r_{11} & r_{12} & r_{13} & x_t \\ r_{21} & r_{22} & r_{23} & y_t \\ r_{31} & r_{32} & r_{33} & z_t \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.16)$$

$$\vec{r} = [k_1 \ k_2 \ k_3 \dots] \quad (2.17)$$

$$\vec{t} = [p_1 \ p_2] \quad (2.18)$$

## 2.2 Ocenjevanje parametrov

V poglavju 2.1 je opisan model kamere. Kamero določajo razni parametri, ki pa so za vsako kamero različni. V tem poglavju bom opisal kako oceniti posamezne parametre kamere, da bo model postal uporaben.

### 2.2.1 Ocenjevanje notranjih parametrov

Za ocenjevanje notranjih parametrov kamere obstaja veliko algoritmov, ki pa se razlikujejo po hitrosti, težavnosti in natančnosti. Dve popularni metodi sta Tsajiev kalibracijski algoritem [4] in Zhangova fleksibilna tehnika za kalibracijo kamer [14]. Sam sem za ocenjevanje notranjih parametrov uporabil MATLAB-ovo kalibracijsko orodje, ki uporablja variacijo Zhangove tehnike. Več o samem postopku kalibracije je opisano v poglavju 3.3.

### 2.2.2 Ocenjevanje zunanjih parametrov

Zunanji parametri kamere določajo kje v prostoru se kamera nahaja. Zgoraj omenjena Zhangova metoda poleg notranjih vrne tudi zunanje parametre, ki pa za namen diplomskega dela niso uporabni, saj bi morali kamere kalibrirati

z vsaj eno sliko ploskve, ki jo v celoti vidijo vse kamere. Sistem mora delovati tudi, če imajo kamere minimalno ali celo ničelno prekrivanje vidnega polja. V nadaljevanju bom opisal teoretično podlago metode, ki sem jo uporabil za ocenjevanje zunanjih parametrov. Doseči želimo, da kameram določimo skupen koordinatni sistem sveta.

Metoda predpostavlja, da imamo za neko kamero že izračunane notranje parametre. Za oceno zunanjih parametrov je dovolj le ena slika iz kamere, ki pa ne sme biti popačena. Najprej je torej potrebno popačeno sliko popraviti. Enačbe (2.7) in (2.10) določajo neposredno preslikavo med popačenimi in nepopačenimi točkami.

$$\vec{X}' \sim A * [R|\vec{T}] * \vec{X} \quad (2.19)$$

Zunanje parametre določa rotacijska matrika  $R$  in translacijski vektor  $\vec{T}$ , kar je v zgornji enačbi kompaktno predstavljeno z matriko  $[R|\vec{T}]$ . Za oceno teh parametrov moramo poznati notranje parametre  $A$ , točko v svetu  $X$  in njeni projekciji na sliko  $X'$ . Matriko  $[R|\vec{T}]$  lahko ocenimo z metodo DLT (*ang. Direct Linear Transformation*). Ker računamo s homogenimi koordinatami, sta si leva in desna stran enačbe enaka do poljubnega neničelnega faktorja  $\lambda$ . Enačbo (2.19) lahko zapišemo z

$$\lambda * \vec{X}' = A * [R|\vec{T}] * \vec{X} \quad (2.20)$$

Z eno znano točko (na sliki in v svetu) dobimo 3 enačbe, vendar pa je ena linearna kombinacija drugih dveh, zato nam pri ocenjevanju zunanjih parametrov ne pomaga. Oceniti moramo torej 12 neznank (ker je  $[R|\vec{T}]$  matrika velika  $3 \times 4$ ), z eno točko pa dobimo 2 neodvisni enačbi, kar pomeni, da potrebujemo najmanj 6 točk za katere poznamo svetovne koordinate in njihove projekcije na sliko. Dobimo torej sistem enačb,

$$\begin{aligned}\lambda_1 * \vec{X}'_1 &= A * [R|\vec{T}] * \vec{X}_1 \\ \lambda_2 * \vec{X}'_2 &= A * [R|\vec{T}] * \vec{X}_2 \\ \lambda_3 * \vec{X}'_3 &= A * [R|\vec{T}] * \vec{X}_3 \\ &\vdots\end{aligned}$$

Problem predstavljajo neničelni faktorji na levi strani enačb, saj jih ne poznamo in so odvisni od zunanjih parametrov kamere. Metoda DLT reši ravno tak sistem enačb. Leva stran je pravzaprav 3-dimenzionalni vektor za katerega vemo, da je vedno enak nič, če ga vektorsko pomnožimo s samim seboj. Vektorsko množenje pa lahko predstavimo z matričnim množenjem.

$$\vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad (2.21)$$

$$\vec{a} \times \vec{a} = [\vec{a}]_\times * \vec{a} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = 0 \quad (2.22)$$

Obe strani enačbe (2.20) lahko pomnožimo z leve z  $[\vec{X}']_\times$ .

$$\lambda * [\vec{X}']_\times * \vec{X}' = [\vec{X}']_\times * A * [R|\vec{T}] * \vec{X} \quad (2.23)$$

$$0 = [\vec{X}']_\times * A * [R|\vec{T}] * \vec{X} \quad (2.24)$$

S tem korakom smo se znebili neznanega parametra  $\lambda$ , vendar iz take oblike enačbe težko najdemo rešitev sistema. Za lažjo izpeljavo bomo enačbo (2.20) z leve pomnožili z inverzom matrike  $A$ .

$$A^{-1} * \vec{X}' = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.25)$$

$$[A^{-1} * \vec{X}']_{\times} = \begin{bmatrix} 0 & -w & v \\ w & 0 & -u \\ -v & u & 0 \end{bmatrix} \quad (2.26)$$

$$\lambda * A^{-1} * \vec{X}' = [R|\vec{T}] * \vec{X} \quad (2.27)$$

$$0 = [A^{-1} * \vec{X}']_{\times} * [R|\vec{T}] * \vec{X} \quad (2.28)$$

$$0 = \begin{bmatrix} 0 & -w & v \\ w & 0 & -u \\ -v & u & 0 \end{bmatrix} * \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.29)$$

$$0 = \begin{bmatrix} x(-wp_5 + vp_9) + y(-wp_6 + vp_{10}) + z(-wp_7 + vp_{11}) + (-wp_8 + vp_{12}) \\ x(wp_1 - up_9) + y(wp_2 - up_{10}) + z(wp_3 - up_{11}) + (wp_4 - up_{12}) \\ x(-vp_1 + up_5) + y(-vp_2 + up_6) + z(-vp_3 + up_7) + (-vp_4 + up_8) \end{bmatrix} \quad (2.30)$$

Sistem enačb (2.30) lahko zapišemo v obliki  $B * \vec{p} = 0$ , kjer

$$\vec{p} = [p_1 \ p_2 \ p_3 \ p_4 \ p_5 \ p_6 \ p_7 \ p_8 \ p_9 \ p_{10} \ p_{11} \ p_{12}]^T \quad (2.31)$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & -xw & -yw & -zw & -w & xv & yv & zv & v \\ xw & yw & zw & w & 0 & 0 & 0 & 0 & -xu & -yu & -zu & -u \\ -xv & -yv & -zv & -v & xu & yu & zu & u & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.32)$$

Zadnja vrstica v matriki  $B$  je linearnejša kombinacija prvih dveh. Za vsak par točk (v svetu in na sliki) generiramo prvi dve vrstici matrike  $B$  in vse vrstice združimo v skupno matriko  $B$  velikosti  $2n \times 12$ , kjer je  $n$  število parov točk. Sistem enačb lahko sedaj rešimo z razcepom na singularne vrednosti

(ang. *SVD - Singular Value Decomposition*). SVD razcepi matriko  $B$  na  $U\Sigma V^T$ , kjer je  $U$  matrika levih lastnih vektorjev,  $\Sigma$  matrika singularnih vrednosti in  $V$  matrika desnih lastnih vektorjev. Po definiciji za leve lastne vektorje matrike  $B$  velja  $B * \vec{v} = \lambda * \vec{v}$ , kjer je  $\lambda$  lastna vrednost. Če iz matrike  $V$  vzamemo vektor, ki ustrza najmanjši lastni vrednosti v matriki  $\Sigma$ , smo tako dobili netrivialno rešitev, ki najbolje zadošča pogoju  $B * \vec{p} = 0$ . Rešitev je potrebno le še preoblikovati v  $3 \times 4$  matriko zunanjih parametrov. V poglavju 3.3 naslovim problem, ki nastane, če matrika  $B$  nima ranga 12, kar se zgodi, če ocenjujemo parametre s koplanarnimi točkami v prostoru.

## 2.3 Triangulacija

Iz modela (2.15) želimo oceniti  $\vec{X}$ . Uporabimo lahko podoben postopek kot pri ocenjevanju zunanjih parametrov.

$$\lambda * \vec{X}' = A * [R|\vec{T}] * \vec{X} \quad (2.33)$$

$$0 = [\vec{X}']_\times * A * [R|\vec{T}] * \vec{X} \quad (2.34)$$

$$B = [\vec{X}']_\times * A * [R|\vec{T}] \quad (2.35)$$

$$B * \vec{X} = 0 \quad (2.36)$$

Rang matrike  $B$  mora biti 4, saj ocenjujemo štiri parametre, ki določajo  $X$ . Ena točka na sliki nam poda 2 linearne neodvisne enačbe. Potrebujemo torej vsaj 2 različni točki iz dveh različnih slik, ki predstavlja projekcijo iste točke v prostoru na sliko. Sistem zopet rešimo z razcepom na singularne vrednosti kot smo to naredili v prejšnjem poglavju.

## 2.4 Zaznavanje označevalnika

Pri zaznavanju označevalnika je potrebno sliko iz kamere segmentirati. Segmentacija je postopek ločevanja ozadja od objekta zanimanja. Označevalnik ima običajno izrazito drugačno barvo od okolice, zato da pri zaznavanju ne

prihaja do dvoumnosti. Segmentacija ponavadi vrne binarno sliko, kjer je ozadje predstavljen z vrednostjo 0, segmenti pa z vrednostjo 1.

Segmentiramo lahko tako sivinsko kot barvno sliko. Za vsako točko v sliki se moramo odločiti ali pripada ozadju (0) ali segmentu (1). Če so vrednosti pik predstavljene z monotono naraščajočimi vrednostmi, lahko določimo prag, ki ločuje ozadje od segmenta. Spodaj je podana psevdo koda segmentacije.

```
I //sivinska slika
w // sirina slike
h // visina slika
T //prag
for i = 1...h
    for j = 1...w
        if I [ i ] [ j ] > T
            I [ i ] [ j ] = 1
        else
            I [ i ] [ j ] = 0
```

Pri segmentiranju barvnih slik se je potrebno odločiti za primeren barvni model. Primarno so slike zapisane v rdečem, zelenem in modrem kanalu (model RGB). Običajno je bolj primeren model HSV ali HSL, saj H (ton, *ang. hue*) komponenta določa barvo neodvisno od njene intenzitete in nasičenosti. Slednja modela sta pri segmentaciji bolj robustna, ker sprememba svetlobe ne vpliva toliko na spremembo tonskega kanala.

Izbira barvnega modela in praga je za vsak primer specifična. Včasih je bolje uporabiti model RGB npr. v primeru, če ima označevalnik izrazito barvo ne glede na osvetljitev.

Iz segmentirane slike moramo izračunati središče zaznanega objekta (če je prisoten na sliki). To lahko storimo s slikovnimi momenti. Prostorski momenti na sivinski sliki  $I$  so definirani kot,

$$M_{ij} = \sum_x \sum_y x^i * y^j * I(x, y) \quad (2.37)$$

Poznamo še centralne momente, ki pa so definirani kot,

$$\bar{x} = \frac{M_{10}}{M_{00}} \quad (2.38)$$

$$\bar{y} = \frac{M_{01}}{M_{00}} \quad (2.39)$$

$$\mu_{ij} = \sum_x \sum_y (x - \bar{x})^i * (y - \bar{y})^j * I(x, y) \quad (2.40)$$

Masno središče je predstavljeno z  $\bar{x}$  in  $\bar{y}$ . Moment  $M_{00}$  je enak površini segmenta,  $M_{10}$  ter  $M_{01}$  pa sta vsoti vrednosti po  $x$  in  $y$  koordinatah.

# Poglavlje 3

## Implementacija

### 3.1 Strojna oprema

#### 3.1.1 Računalnik

Vsa obdelava podatkov se odvija na enem prenosnem računalniku. Zajemanje slik poteka sočasno v ločenih procesih. Računalnik ima dve fizični procesni jedri in štiri niti. V lokalno omrežje je povezan preko brezžične povezave.

CPE	Intel i7-4510U 2.6 GHz
Pomnilnik	8 GB
Št. jeder/niti	2 / 4
Arhitektura	64-bit
OS	Windows 8.1 Pro

Tabela 3.1: Specifikacije računalnika.

#### 3.1.2 Kamere

Uporabil sem štiri Axis 215 PTZ omrežne kamere. To so varnostne IP kamere namenjene primarno nadzoru okolja. Kamere so povezane v zvezdišče na omrežju ethernet, zvezdišče pa je povezano z brezžičnim usmerjevalnikom.

Kamere so zmožne v realnem času preko lokalnega omrežja posredovati do 30 slik na sekundo pri ločljivosti  $704 \times 576$ . Vseeno pa varnostne kamere niso namenjene za sinhronizirano zajemanje slik in ne podpirajo skupnega prožilca. Največja ločljivost je  $704 \times 576$ , kar je relativno malo za pozicioniranje objekta v  $7,5 \times 7,5 \times 3\ m^3$  velikem prostoru.



Slika 3.1: Axis 215 PTZ kamera.

Tip kamere	varnostna IP
Hitrost zajemanja	30 FPS
Formati pretoka	MPEG-4, MJPEG
Največja ločljivost	704 × 576
Optična povečava	12×
Digitalna povečava	4×
Povezava z omrežjem	ethernet
Leča	3,8 - 46 mm

Tabela 3.2: Specifikacije Axis 215 PTZ.

Axis kamere imajo CGI (*ang. Common Gateway Interface*) vmesnik, ki omogoča nadzor funkcionalnosti preko protokola HTTP (*ang. Hypertext Transfer Protocol*). Vsak model kamere podpira različen nabor ukazov, katerih spisek lahko dobimo z GET zahtevkom na naslov:

```
http://<ip-kamere>/axis-cgi/com/ptz.cgi?info=1
```

Vsi ukazi pa so podani kot parameter v zahtevku.

```
http://<ip-kamere>/axis-cgi/com/ptz.cgi?<ukaz>=<vrednost>
```

Spodaj je prikazan izpis ukaza `info=1` na Axis 215 PTZ kameri.

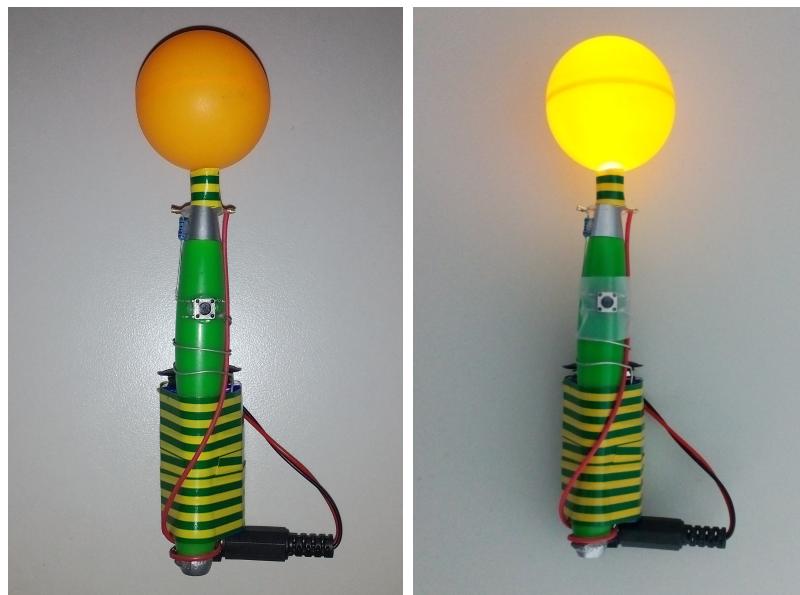
Available commands

```
:
{ camera=[n] }
whoami=yes
center=[x],[y]
    imagewidth=[n]
    imageheight=[n]
move={ home | up | down | left | right | upleft | upright | downleft }
pan=[abspos]
tilt=[abspos]
zoom=[n]
```

```
focus=[n]
rpan=[offset]
rtilt=[offset]
rzoom=[offset]
rfocus=[offset]
brightness=[offset]
rbrightness=[offset]
autofocus={ on | off }
ircutfilter={ on | off | auto }
backlight={ on | off }
continuouspantiltmove=[x-speed],[y-speed]
continuouszoommove=[speed]
continuousfocusmove=[speed]
auxiliary=function
setserverpresetname=[name]
setserverpresetno=[n]
removeserverpresetname=[name]
gotoserverpresetname=[name]
gotoserverpresetno=[n]
barcoord=[x],[y]
    panbar=[length],{ horizontal | vertical }
    tiltbar=[length],{ horizontal | vertical }
    zoombar=[length],{ horizontal | vertical }
    focusbar=[length],{ horizontal | vertical }
    irisbar=[length],{ horizontal | vertical }
    brightnessbar=[length],{ horizontal | vertical }
speed=[n]
query={ speed | position | presetposcam | presetposall }
```

### 3.1.3 Označevalnik

Označevalnik je v osnovi oranžna žogica za namizni tenis. Da pa bi spre-memba svetlobe čim manj vplivala na natančnost zazanavanja označevalnika, sem v žogico vgradil modro svetlečo diodo (*LED - Light Emitting Diode*) in stikalo. S pritiskom na stikalo prižgemo označevalnik, da ga kamere lahko zaznajo. Na slikah kamer je žogica izrazito rumene barve.



Slika 3.2: Ugasnjen označevalnik je oranžne barve (levo), prižgan pa izrazito rumene barve (desno).

## 3.2 Programska oprema

Z delom sem začel v programu MATLAB R2014a. MATLAB je visokonivojski programski jezik in razvojno okolje, ki je primarno namenjeno prototipiranju. Podpira računanje z visokonivojskimi strukturami kot so matrike in vektorji. V njem sem razvil konceptno rešitev, ki sem jo nato implementiral v drugi aplikaciji.

Za zajemanje statičnih slik iz kamer sem uporabil Node.js v0.12.3. Node.js

je JavaScript pogon z veliko različnimi moduli. Node.js arhitektura je po zasnovi asinhrona, kar pri vhodno/izhodnih operacijah močno pohitri sistem.

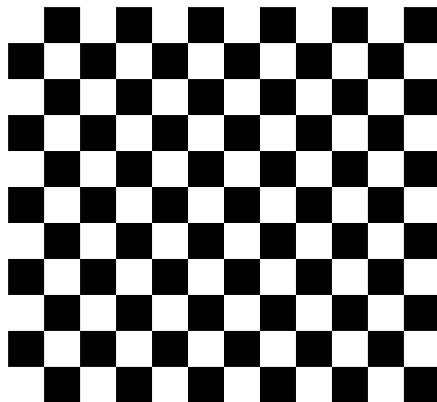
Za iskanje kamer v omrežju sem uporabil Axisov IPUtility, ki samodejno vrne IP naslove vseh kamer, ki so priključene v omrežje.

Za implementacijo glavnega sistema pa sem uporabil Python 2.7 s knjižnicami Numpy v1.9.2, SciPy v0.15.1 ter OpenCV v3.0.0. Morda se zdi Python za realnočasovni sistem slaba izbira, vendar je s pravilno uporabo knjižnic odlično orodje za hitri razvoj. Funkcije v knjižnicah so zaradi hitrosti implementirane v njenivojskih jezikih kot so C/C++. Knjižnica Numpy je namenjena računanju z matrikami, SciPy pa je njena razširitev. OpenCV (Open Computer Vision) pa je namenjen za probleme umetnega zaznavanja in pri svoji implementaciji uporablja strukture Numpy-ja.

### 3.3 Kalibracija kamer

#### 3.3.1 Ocenjevanje notranjih parametrov

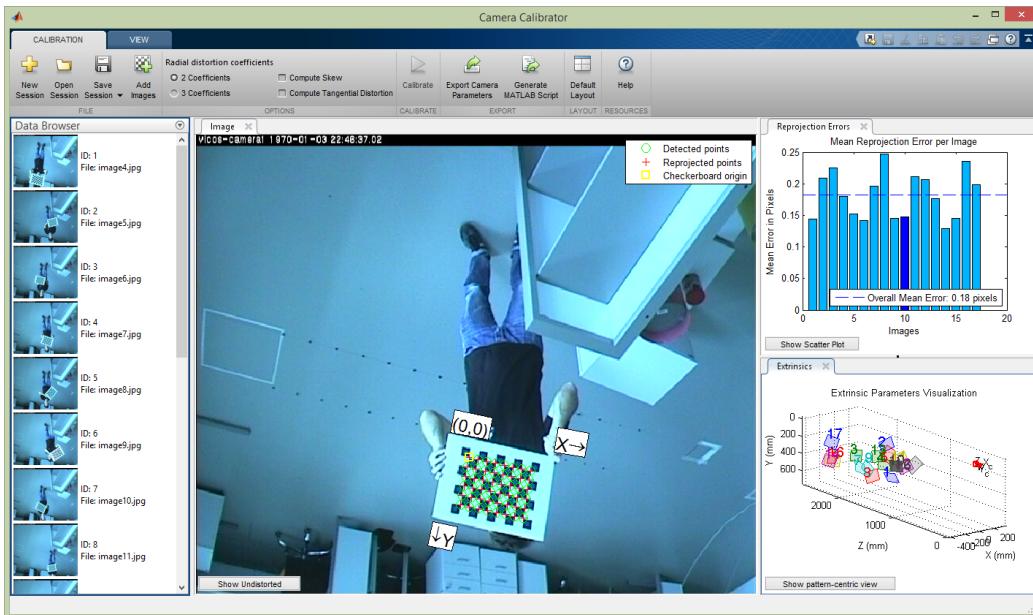
Za ocenjevanje notranjih parametrov sem uporabil MATLAB-ov kalibrator kamere. Za kalibracijo mu moramo podati slike šahovnice, ki so bile zajete s kamero, ki jo želimo kalibrirati. Širina šahovnice mora biti različna od višine, da lahko enolično določimo njeno orientacijo.



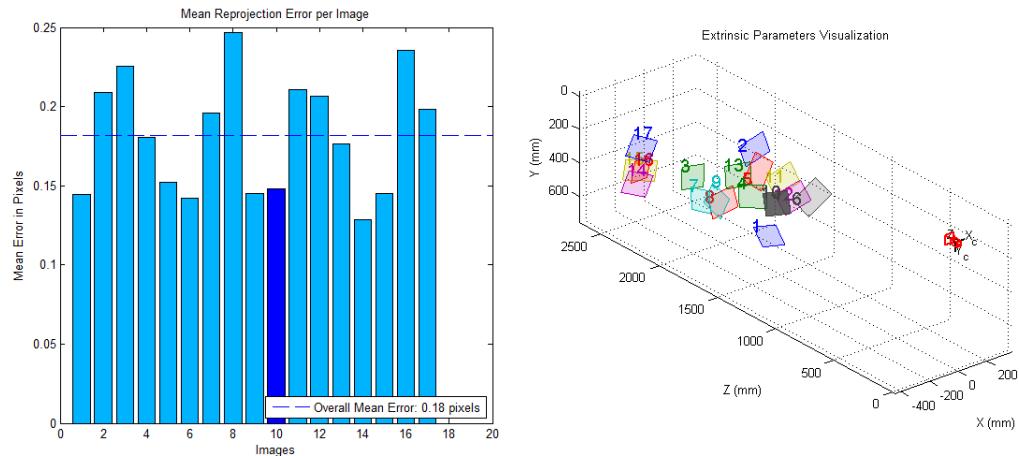
Slika 3.3:  $11 \times 12$  velika šahovnica.

Za učinkovito kalibracijo notranjih parametrov moramo zajeti okoli 10 - 20 slik z različnimi orientacijami in razdaljami šahovnice. Postopek kalibracije z orodjem je sledeč:

1. zajamemo slike šahovnice,
2. naložimo jih v orodje za kalibracijo,
3. izberemo število koeficientov za radialno popačenost (privzeto 2),
4. obkljukamo ali želimo oceniti poševnost in tangencialno popačenost,
5. kalibriramo,
6. postopek lahko ponavljamo s podmnožico slik na podlagi reprojekcijske napake,
7. ocnjene notranje parametre izvozimo za uporabo



Slika 3.4: Na sliki se vidi zaznane robove šahovnice in njena orientacija, graf reprojekcijske napake in vizualizacija zunanjih parametrov.



Slika 3.5: Graf reprojekcijske napake za vsako sliko (levo) in vizualizacija pozicij šahovnic s središčem v kameri (desno).

$$A = \begin{bmatrix} 761,6027 & 0 & 386,6599 \\ 0 & 835,939 & 362,0371 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

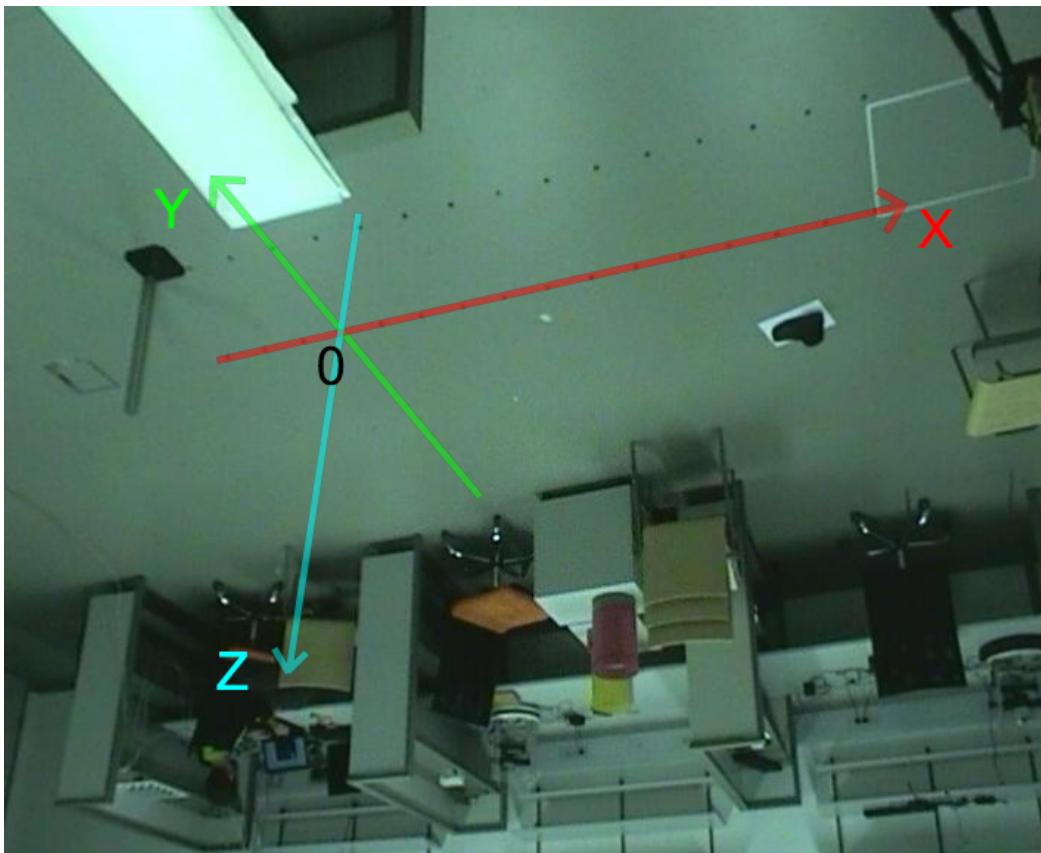
$$\vec{r} = [-0,2934 \ 0,1331] \quad (3.2)$$

$$\vec{t} = [0.0006 \ 0,0012] \quad (3.3)$$

Slika 3.6: Primer ocenjenih notranjih parametrov ene izmed kamer. Tangencialna popačenost je skoraj zanemarljiva.

### 3.3.2 Ocenjevanje zunanjih parametrov

Ocenjevanje zunanjih parametrov je pravzaprav postavljanje kamere v nek skupen koordinatni sistem sveta. Določiti moramo torej rotacijsko matriko  $R$  in translacijski vektor  $\vec{T}$ , ki predstavlja kje se nahaja koordinatni sistem glede na kamero. Za oceno parametrov uporabimo metodo, ki je opisana v poglavju 2.2.2. Potrebujemo le eno sliko na kateri lahko določimo točke v svetu. Za potrebe diplomskega dela sem v sobi na tleh označil in določil svoj koordinatni sistem. Med oznakami po  $x$  osi je razdala 20 cm, na  $y$  osi pa 60 cm. Enota je 1 cm.



Slika 3.7: Na sliki se vidi označen koordinatni sistem sveta. Točke označene s črnimi pikami so koplanarne in z njimi določimo zunanje parametre kamere.

Najprej moramo določiti pare  $(\vec{X}, \vec{X}')$ , ki določajo preslikavo med točko v svetu in točko na sliki. Imeti moramo minimalno 6 parov, ker moramo oceniti 12 parametrov (9 za rotacijsko matriko in 3 za translacijo), en par pa nam določi 2 linearno neodvisni enačbi. Točke v svetu določimo sami, nato pa jih ročno ali avtomatsko označimo na sliki. Sam sem določil 16 parov, s katerimi sem ocenil zunanje parametre. Če je slika kakorkoli popačena, jo moramo popraviti.

x	y	z	x'	y'
240	0	0	591.1441	139.8898
200	0	0	523.8898	156.1610
160	0	0	460.9746	171.3475
120	0	0	399.1441	185.4492
80	0	0	340.5678	198.4661
40	0	0	284.1610	211.4831
0	0	0	229.9237	223.4153
-40	0	0	177.8559	234.2627
240	60	0	584.6356	62.8729
200	60	0	508.7034	83.4831
160	60	0	437.1102	101.9237
120	60	0	368.7712	119.2797
80	60	0	303.6864	135.5508
40	60	0	241.8559	150.7373
0	60	0	183.2797	165.9237
-40	60	0	125.7881	177.8559

Tabela 3.3: 16 parov točk dobljenih iz slike 3.7.

Iz zgornje tabele se lepo vidi, da so vse točke v svetu koplanarne, saj je pri vseh  $z = 0$ . Sedaj lahko določimo matriko  $B$ , ki je za eno par točk definirana v enačbi (2.32). Za vsak par točk dodamo nove vrstice matriki  $B$  in tako dobimo matriko velikosti  $2n \times 12$  oz.  $3n \times 12$  (če izračunamo tudi zadnjo vrstico, ki je redundantna), kjer je  $n$  število parov točk. Zavedati se moramo, da so  $u, v, w$  določeni z enačbo (2.25), kjer uporabimo matriko notranjih parametrov  $A$ . S tem korakom odstranimo matriko  $A$  iz matrike  $A * [R|\vec{T}]$  in s tem lažje določimo matriko  $[R|\vec{T}]$ . Z razcepom na singularne vrednosti (SVD) rešimo sistem  $B * \vec{p} = 0$  in dobljeno rešitev preoblikujemo v  $3 \times 4$  matriko zunanjih parametrov. Rešitev je pravzaprav deveti stolpec matrike  $V$ , ki jo vrne SVD razcep in ne dvajsti kot je opisano pri izpeljavi teorije. To je posledica koplanarnosti svetovnih točk. Matrika  $B$  je ranga 9,

ker je  $z$  komponenta vseh točk v svetu enaka 0. Spodnja matrika prikazuje primer rešitve za pare točk iz tabele 4.1.

$$[R|\vec{T}] = \begin{bmatrix} -0.0018 & 0.0006 & 0 & 0.1989 \\ 0.0003 & 0.0007 & 0 & 0.1607 \\ 0.0005 & 0.0016 & 0 & -0.9668 \end{bmatrix} \quad (3.4)$$

Hitro opazimo, da podmatrika  $R$  ni prav nič podobna rotacijski matriki. Stolpci rotacijskih matrik so paroma ortonormirani. Ker računamo s homogenimi koordinatami, lahko dobljeno matriko pomnožimo z neničelnim skalarjem brez, da bi pri tem uničili model kamere. Sedaj lahko ročno popravimo podmatriko  $R$ , da bo karseda podobna rotacijski matriki.

$$[R|\vec{T}] = [\vec{R}_1 \ \vec{R}_2 \ \vec{R}_3 \ \vec{T}] \sim \frac{1}{\|\vec{R}_1\|} * \begin{bmatrix} -0.0018 & 0.0006 & 0 & 0.1989 \\ 0.0003 & 0.0007 & 0 & 0.1607 \\ 0.0005 & 0.0016 & 0 & -0.9668 \end{bmatrix} \quad (3.5)$$

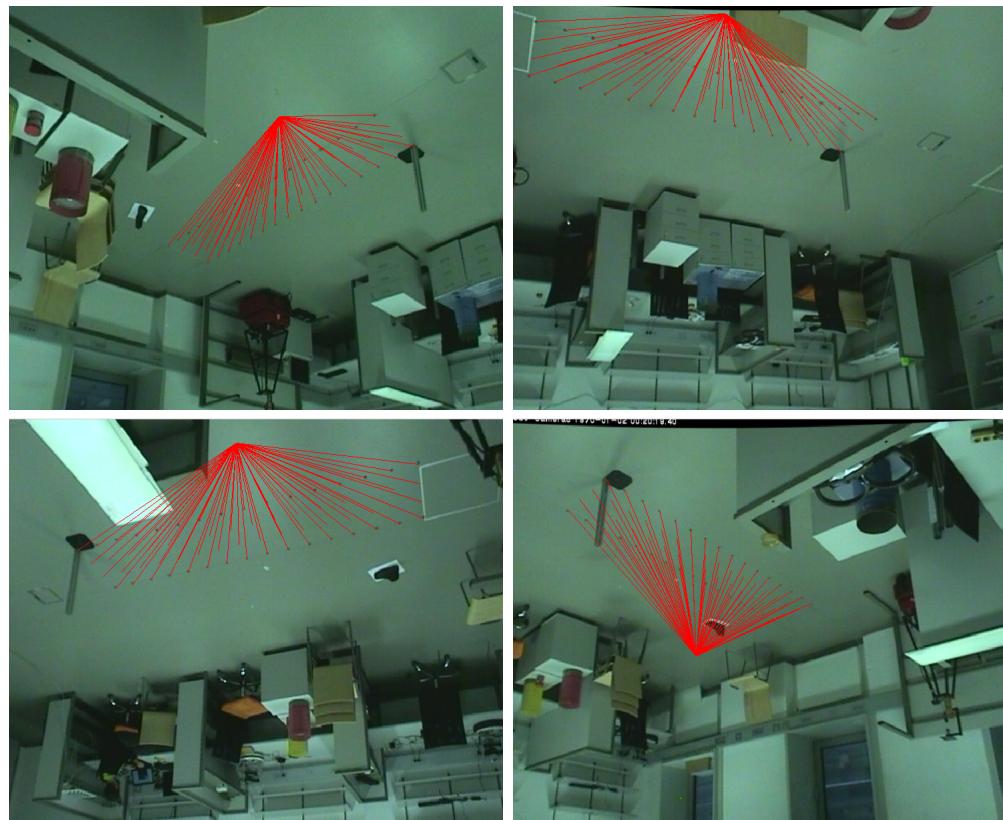
$$[R|\vec{T}] \sim \begin{bmatrix} -0.9544 & 0.3003 & 0 & 106.6226 \\ 0.1419 & 0.3877 & 0 & 86.1440 \\ 0.2627 & 0.8810 & 0 & -518.2979 \end{bmatrix} \quad (3.6)$$

Sedaj sta prva dva stolpca ortonormirana oz. skoraj ortonormirana (odvisno od natančnosti ocene parametrov). Sedaj je čas, da naslovimo še zadnji problem rotacijske matrike. Zaradi prej omenjene koplanarnosti je tretji stolpec ničelni vektor. Vemo, da mora biti paroma ortonormiran s prvima stolpcema, zato ga lahko določimo z vektorskim produktom.

$$R_3 = R_1 \times R_2 = \begin{bmatrix} -0.9544 \\ 0.1419 \\ 0.2627 \end{bmatrix} \times \begin{bmatrix} 0.3003 \\ 0.3877 \\ 0.8810 \end{bmatrix} \quad (3.7)$$

$$R_3 = \begin{bmatrix} 0.0232 \\ 0.9197 \\ -0.4127 \end{bmatrix} \quad (3.8)$$

Sedaj moramo le še določiti pravo smer novo izračunanega vektorja.  $R_3$  kot tudi  $-R_3$  sta legalni rešitvi in implicitno določata sučnost koordinatnega sistema. Sam sem pravilno rešitev določil ročno na podlagi projekcije točk, kot je vidno iz slike 3.8.



Slika 3.8: Projekcija piramide na sliko pokaže, da so prvi trije koordinatni sistemi levosučni, zato moramo obrniti smer zadnjega stolpca matrike  $R$ .



Slika 3.9: Na novo ocenjeni zunanji parametri pravilno modelirajo kamere, kar je razvidno iz projekcije piramide.

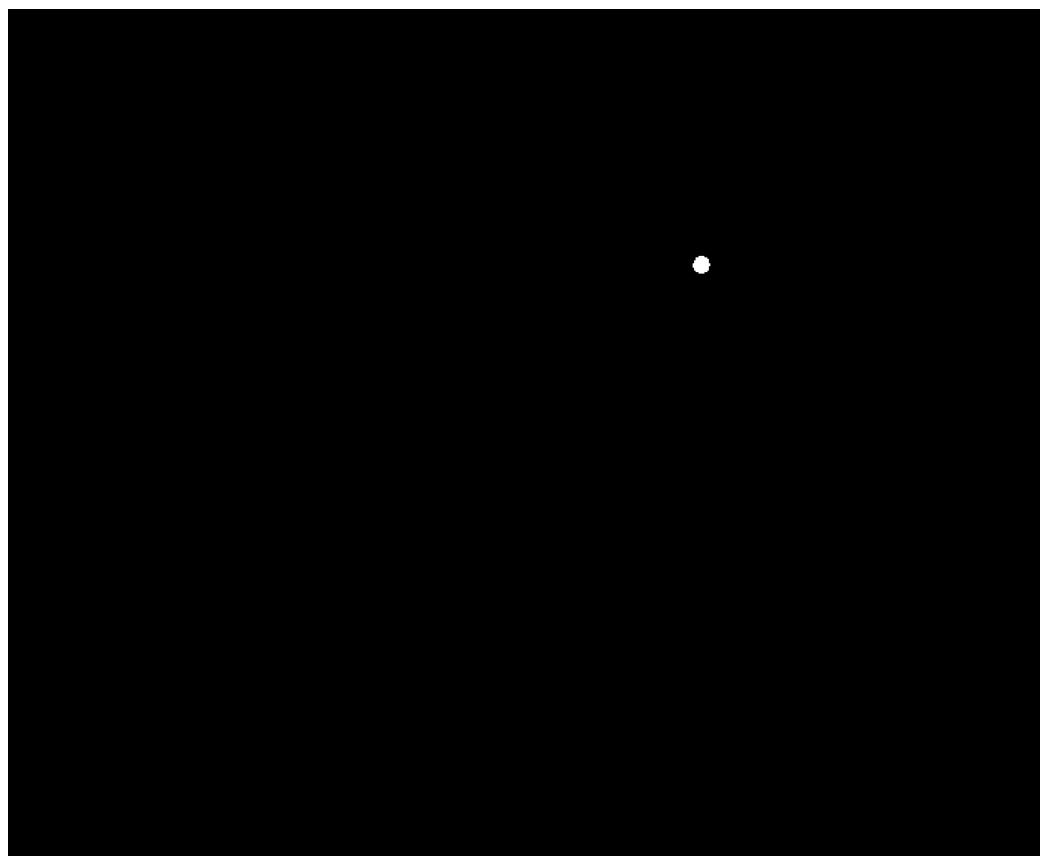
Končna matrika zunanjih parametrov za točke iz tabele 4.1 je,

$$[R|\vec{T}] = \begin{bmatrix} -0.9544 & 0.3003 & 0.0232 & 106.6226 \\ 0.1419 & 0.3877 & 0.9197 & 86.1440 \\ 0.2627 & 0.8810 & -0.4127 & -518.2979 \end{bmatrix} \quad (3.9)$$

Stolpci rotacijske matrike so le približno paroma ortonormirani, kar je posledica vnašanja napak pri ročnem določanju koordinatnega sistema. Če bi matriko prisilno ortonormirali bi prišlo do večje reprojekcijske napake.

### 3.4 Implementacija zaznavanja označevalnika

Za zaznavanje označevalnika uporabljam postopek opisan v poglavju 2.4. Za prag sem določil RGB interval, ker je označevalnik izrazito rumene barve. Spodnja meja intervala je [220 220 0], zgornja pa [255, 255, 200]. V prostoru ne sme prihajati do velikih sprememb v osvetljenosti, zato imajo kamere nastavljen fiksni kontrast, dnevna svetloba pa je izolirana od prostora. Spodaj so prikazane slike posameznih faz zaznavanja.



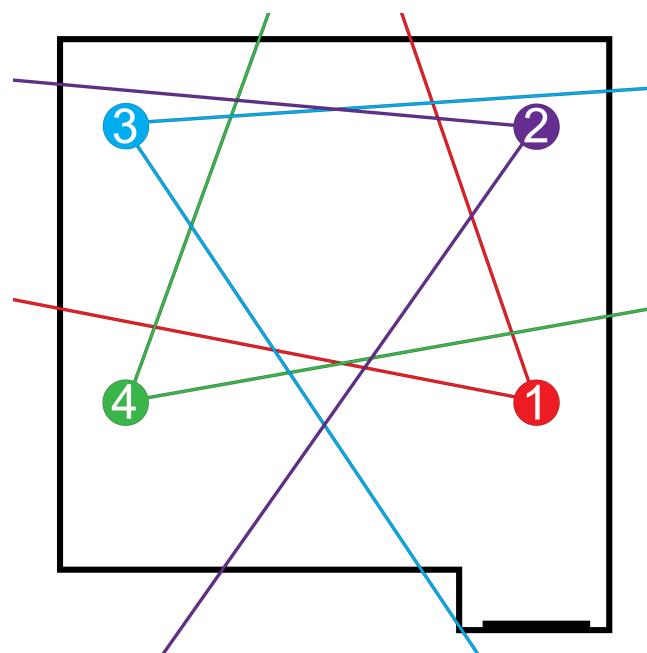
Slika 3.10: Segmentacija zajete slike z belo barvo označuje zaznan označevalnik.



Slika 3.11: Pozicijo označevalnika na sliki (zeleni +) dobimo z masnim središčem segmenta.

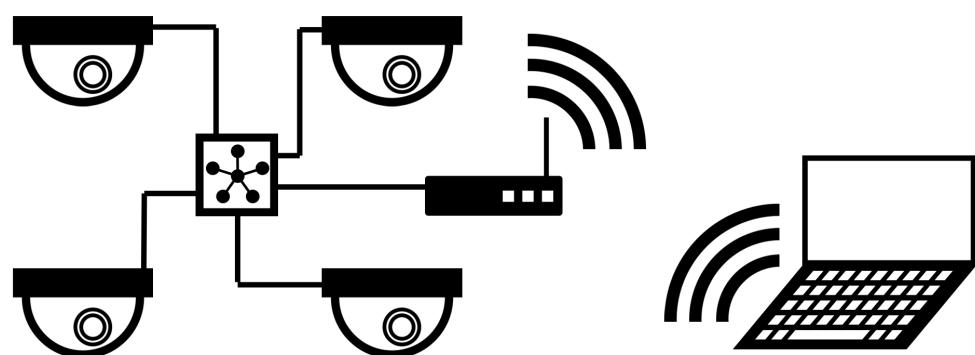
### 3.5 Postavitev sistema

V prostoru z dimenzijsami  $7,4 \times 7,7 \times 2,97 \text{ m}^3$  so bile na strop pritnjene štiri Axis 215 PTZ kamere. Sistem deluje v realnem času s 24 meritvami na sekundo. Spodnja slika ilustrira postavitev in orientacijo kamer ter njihovo oštevilčenost.



Slika 3.12: Kamere v prostoru.

Kamere so preko ethernet omrežje povezane v zvezdišče in nato v brezžični usmerjevalnik. Prenosni računalnik je v omrežje povezan brezžično. Na računalniku teče glavni program, ki komunicira s kamerami in računa položaj označevalnika v prostoru. Topologijo povezave predstavlja spodnja slika.



Slika 3.13: Topologija povezave.

Proces zaznavanja položaja je razdeljen v preproste korake, ki jih opisuje spodnja psevdo koda.

Glavni proces :

Zazeni n podprocesov , kjer je n stevilo kamer .

Ponavljam :

Za vsak podproces :

Sprejmi tocko oznacevalnika .

Dopolni matriko za triangulacijo .

Ce vsaj dve kamери zaznata oznacevalnik :

Trianguliraj položaj oznacevalnika v prostoru .

Izpisi položaj .

Podproces :

Vzpostavi komunikacijo s kamero .

Inicjaliziraj privzete parametre kamere .

Ponavljam :

Zajemi sliko iz kamere .

Zaznaj oznacevalnik .

Sporoci tocko oznacevalnika glavnemu procesu .

# Poglavlje 4

## Rezultati

### 4.1 Reprojekcija kalibracijskih točk

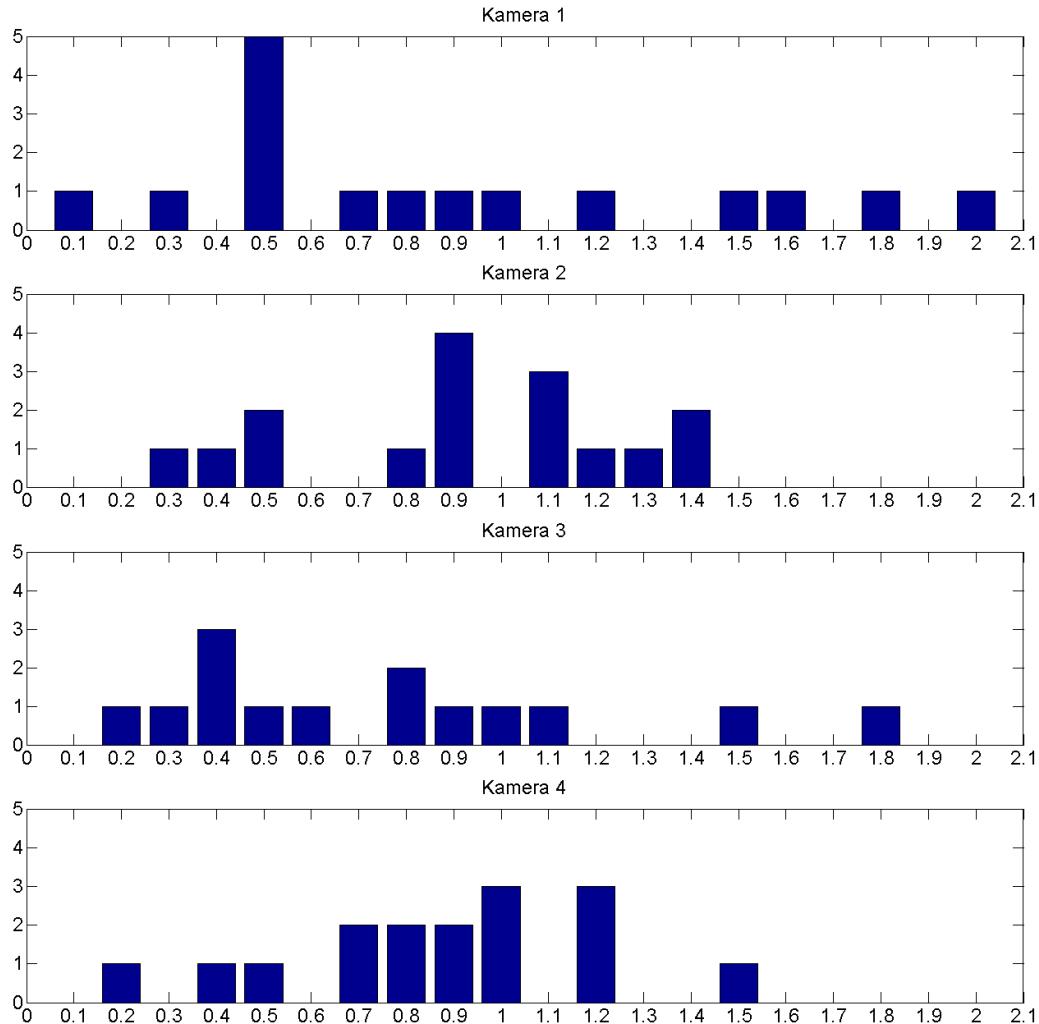
Povprečna reprojekcijska napaka pri kalibriranju notranjih parametrov je med 0.14 in 0.16 slikovnih pik. To je posledica selektivne izbire slik s katerimi zavestno zmanjšamo napako. Napaka pod 1 slikovnim elementom, je pri delu s slikami zelo dobra. Bolj zanimiva je reprojekcija kalibracijskih točk zunanjih parametrov. Točke na slikah so bile določene ročno z ločljivostjo meritve 1 slikovnega elementa. Pomemben faktor pri tem postopku je vnos človeške napake. Sistem se je kljub napakam kalibriral tako, da je vsota razlik kvadratov najmanjša.

	povprečna napaka	varianca povprečne napake
kamera 1	0.8542	0.2965
kamera 2	0.8739	0.1146
kamera 3	0.9786	0.6938
kamera 4	0.8289	0.1011

Tabela 4.1: Povprečna napaka reprojekcije kalibracijskih točk zunanjih parametrov. Napake so merjene v slikovnih elementih.

Opazimo lahko, da so povprečne napake dokaj konsistentne pri okoli 0.9 slikovnega elementa. Pri tretji kameri izstopa varianca, kar je lahko posledica

slabšega ročnega določanja točk.

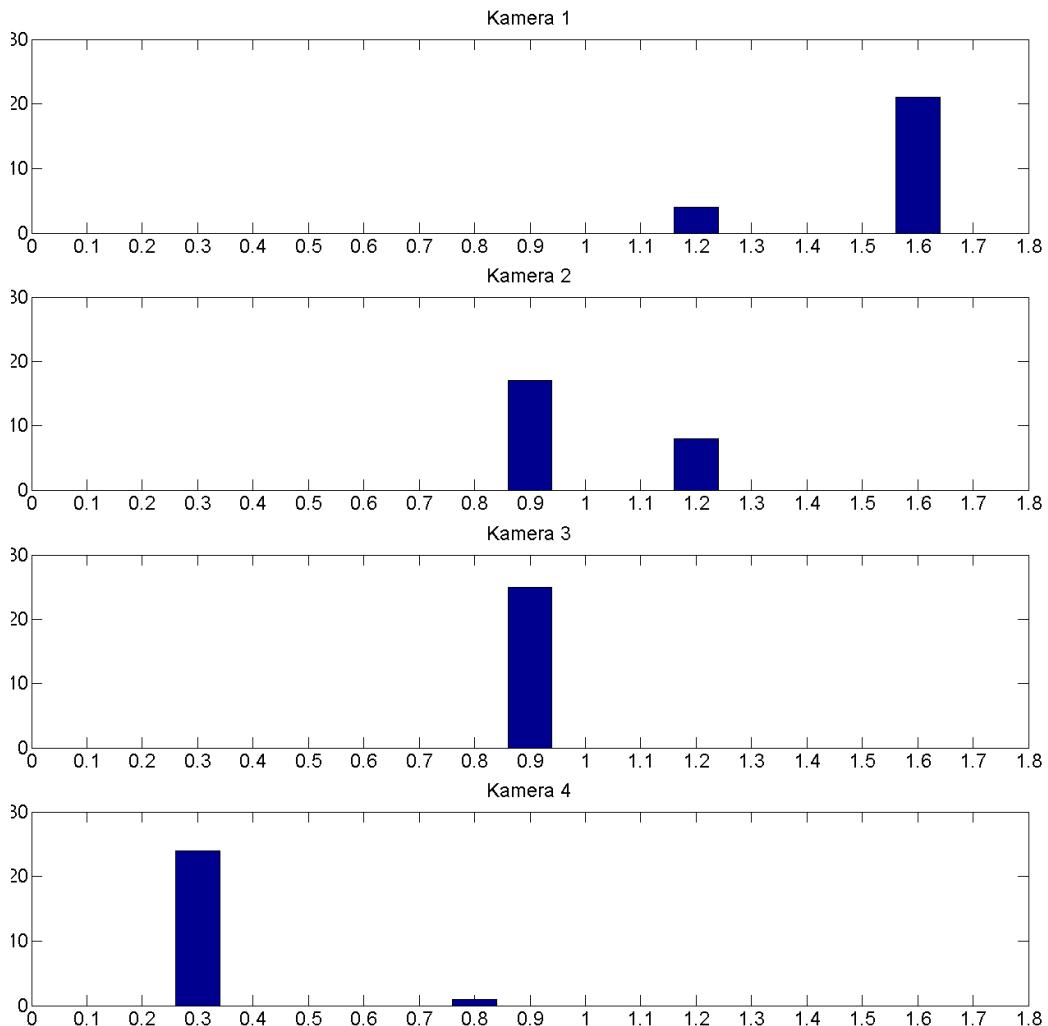


Slika 4.1: Histogrami reprojekcijskih napak. Velikost koša je 0.1 slikovnega elementa.

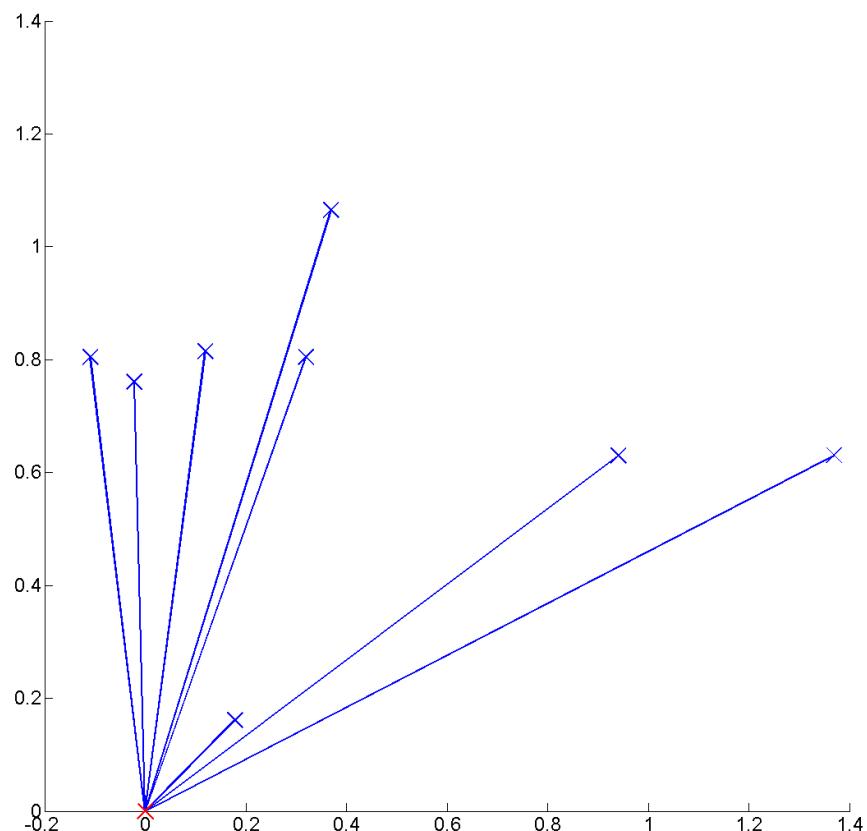
## 4.2 Natančnost zaznavanja označevalnika

Množica podatkov obsega okoli 25 slik označevalnika iz vsake kamere. Skupaj torej okoli 100 slik. Na vseh slikah je bila ročno izbrana sredina označevalnika

in ta točka velja za resnico. Ocenjeni sta dve vrsti množic. Prva ima na vseh slikah statično pozicijo označevalnika, saj je bil v času zajemanja slik označevalnik na istem mestu. Zaradi šuma pri zajemu slik pa algoritom za zaznavanje označevalnika ne zazna vedno enako točko na sliki. Povprečje napake je 0,8614 slikovnega elementa z varianco 0,1899. Iz spodnjih histogramov se lepo vidi, da je varianca majhna. Slike iz tretje in četrte kamere skoraj vedno določijo enako točko označevalnika. Pri slikah iz prve in druge kamere pa šum povzroči majhno spremembo intenzitet pik, kar posledično zmede algoritmom za zaznavanje označevalnika.

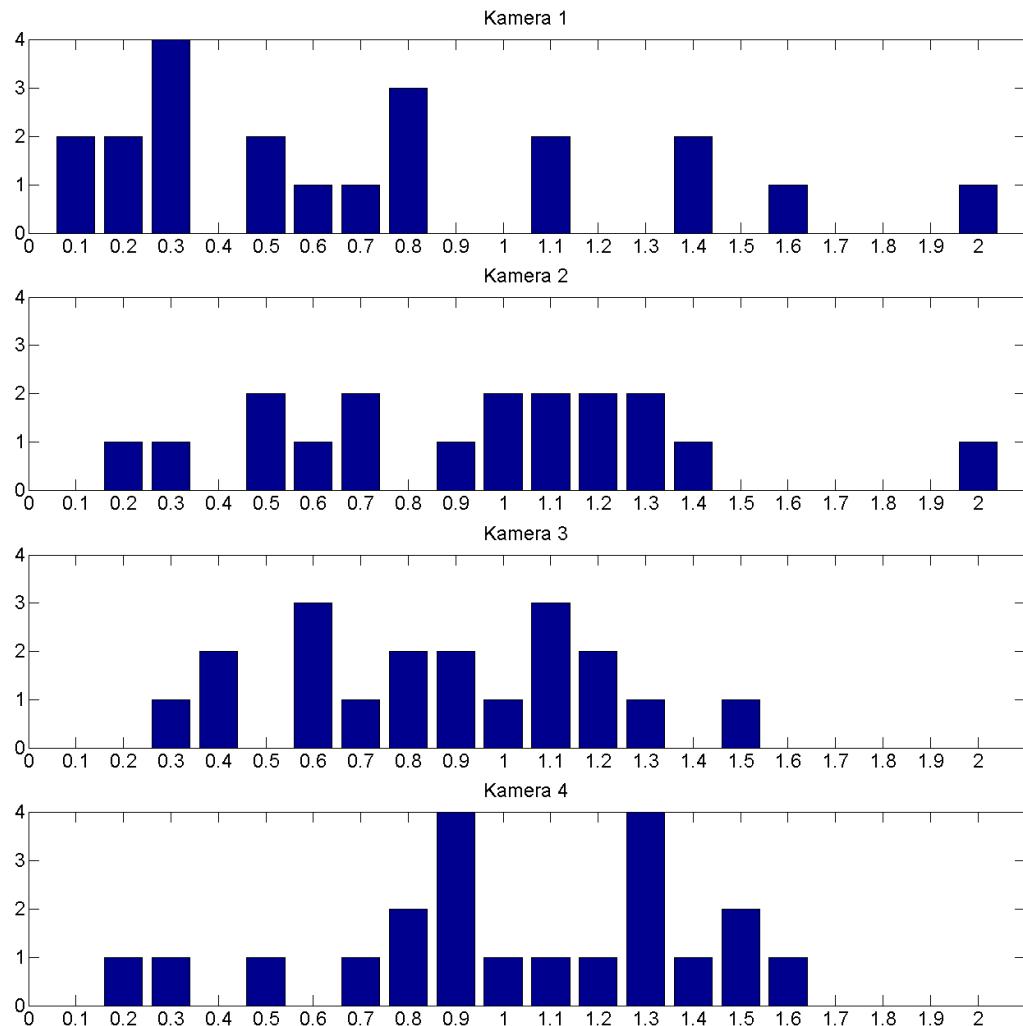


Slika 4.2: Histogrami napak zaznavanja statičnega označevalnika.

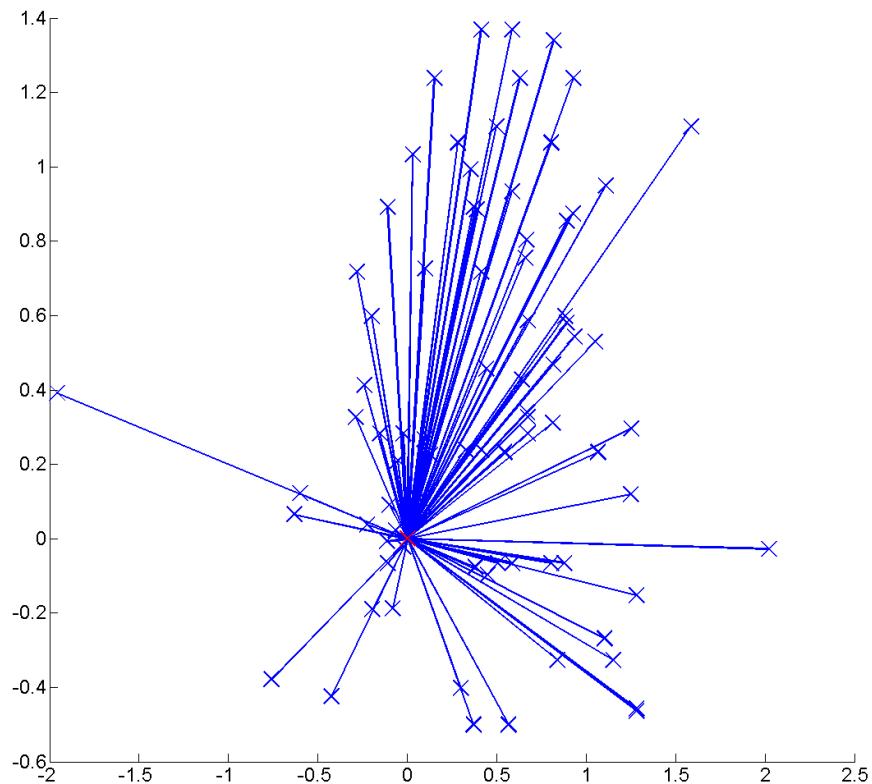


Slika 4.3: Graf vektorjev napak. Enote so slikovne pike. Zanimiva je opazka, da so vektorji usmerjeni predvsem v prvi kvadrant.

V množici 80 slik pa je označevalnik vsakič na drugem mestu v prostoru (je dinamičen). Povprečje napake je 0,852 slikovnega elementa z varianco 0,2124. Iz histogramov vidimo, da je napaka bolj enakomerno porazdeljena kot pri zaznavanju statičnega označevalnika.



Slika 4.4: Histogrami napak zaznavanja dinamičnega označevalnika.



Slika 4.5: Graf vektorjev napak. Enota je slikovni element. Gostota zaznanih točk je večja v zgornji polovici grafa, kar je lahko posledica rahle neenakomerne osvetljenosti označevalnika.

### 4.3 Natančnost pozicioniranja

Zunanji parametri kamer so bili kalibrirani s 16 koplanarnimi točkami. Za testiranje natančnosti pa v prostoru izberemo poljubno točko s poznanimi koordinatami. Izbranih je bilo 12 točk v svetu na katerih je bil postavljen označevalnik. Na eni točki je bilo opravljenih več meritev, ki so prikazane v tabeli 4.2.

št. kamer	višina točke (cm)	povprečna napaka (cm)	varianca napake
2	2	2,3594	0,5225
3	2	1,2737	0,1988
4	2	1,2146	0,6575
2	62,6	3,4443	5,1579
3	62,6	3,8677	2,3627
4	62,6	8,7621	21,4037

Tabela 4.2:

## Poglavlje 5

### Zaključek



# Literatura

- [1] V Auer, MM Maes, MJC Bonfim, MM Wanderley, and MV Lamar. 3d positioning acquisition system with application in real-time processing. *Accessed: October, 1, 2003.*
- [2] S Chan, G Sohn, L Wang, and W Lee. Dynamic wifi-based indoor positioning in 3d virtual world. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(4):1–6, 2013.
- [3] Raffaello D’Andrea. Flying machine arena. <http://raffaello.name/projects/flying-machine-arena/>, 2015. [Online; Dostopano 6. 9. 2015].
- [4] Berthold KP Horn. Tsai’s camera calibration method revisited. *Online: http://people.csail.mit.edu/bkph/articles/Tsai\_Revisited.pdf*, 2000.
- [5] YJ Lee and A Yilmaz. Real-time object detection, tracking, and 3d positioning in a multiple camera setup. In *The ISPRS Workshop on Image Sequence Analysis*, 2013.
- [6] Jinjun Li, Hong Zhao, Tao Jiang, and Xiang Zhou. Development of a 3d high-precise positioning system based on a planar target and two ccd cameras. In *Intelligent Robotics and Applications*, pages 475–484. Springer, 2008.
- [7] Atri Mandal, Cristina V Lopes, Tony Givargis, Amir Haghigat, Raja Jurdak, and Pierre Baldi. Beep: 3d indoor positioning using audible so-

- und. In *Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE*, pages 348–353. IEEE, 2005.
- [8] Raúl Mohedano, Carlos R Del-Blanco, Fernando Jaureguizar, Luis Salgado, and Narciso García. Robust 3d people tracking and positioning system in a semi-overlapped multi-camera environment. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2656–2659. IEEE, 2008.
- [9] Wikipedia. Charge-coupled device — wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=Charge-coupled\\_device&oldid=679480626](https://en.wikipedia.org/w/index.php?title=Charge-coupled_device&oldid=679480626), 2015. [Online; Dostopano 6. 9. 2015].
- [10] Wikipedia. Global positioning system — wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=Global\\_Positioning\\_System&oldid=679232332](https://en.wikipedia.org/w/index.php?title=Global_Positioning_System&oldid=679232332), 2015. [Online; Dostopano 6. 9. 2015].
- [11] Wikipedia. Multilateration — wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=Multilateration&oldid=678002521>, 2015. [Online; Dostopano 6. 9. 2015].
- [12] Wikipedia. Triangulation — wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=Triangulation&oldid=677068010>, 2015. [Online; Dostopano 6. 9. 2015].
- [13] Wikipedia. Trilateration — wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=Trilateration&oldid=677686385>, 2015. [Online; Dostopano 6. 9. 2015].
- [14] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.