

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^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogu:

Tematika naloge:

Lokalizacija predmeta ali osebe v prostoru je zelo pomembna naloga v veliko aplikacijah. Eden izmed pristopov k ugotavljanju pozicije predmeta v prostoru uporablja v ta namen sistem kamer. Izdelajte večkamerni sistem, ki bo sposoben detektirati okrogel predmet v prostoru ter določiti njegovo pozicijo v 3-dimenzionalnem koordinatnem sistemu, določenim s kalibracijo kamer v skupni koordinatni sistem. Tako izdelan sistem naj deluje v realnem času. Izvedite ustrezne eksperimente in evalvirajte natančnost dobljenega sistema.

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:

Rad bi se zahvalil vsem, ki ste ali pa še boste del mojega življenja. Posebna zahvala gre Veroniki, ki mi je priskrbela čas za študij, z opravljanjem nekaterih mojih obveznosti.

Najlepša hvala mentorju doc. dr. Danijelu Skočaju, ki je v svojem zasedenem urniku vedno našel čas za konzultacije in popravke diplomskega dela. Nasvete za pisanje strokovnega dela si bom zapomnil in jih s pridom uporabljal v bodočih delih.

Hvala družini (mama, oče, brat, sestra, polbrat, babi, dedi, teta, stric in ostali), ki je potihem moja velika motivacija za uspeh in pa hvala vsem prijateljem za nepozabne dogodiščine.

Posebno mesto v zahvali si zaslužijo Leneja, Veronika, Milutin, Jure, Nina, Darja, Aleksandar, Iztok, Gregor, Matic, Luka, Eva, Žiga, Simon, Daisy, Matevž, Žan, Lan, David, Nejc, Sabina, Urban, Nataša, Filip, Lara, Jasmina in Borut (vrsti red ne določa prioritete). Zaradi vas imam v življenju zabavo, smeh, motivacijo in smisel.

Zahvala za naj lektorico gre Juliji, saj je v le nekaj urah pregledala in popravila veliko slovničnih napak.

Opravičujem se tistemu, ki ga v zahvali nisem eksplicitno omenil pa meni, da bi ga moral.

Hčerki Leneji.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Motivacija	1
1.2	Pregled področja	2
1.3	Cilji	4
1.4	Struktura diplome	4
2	Teoretična podlaga	5
2.1	Parametri kamere	5
2.2	Ocenjevanje parametrov	12
2.3	Triangulacija	16
3	Implementacija	17
3.1	Strojna oprema	17
3.2	Programska oprema	21
3.3	Kalibracija kamer	21
3.4	Zaznavanje označevalnika	30
3.5	Postavitev sistema	33
4	Eksperimentalni rezultati	35
4.1	Reprojekcija kalibracijskih točk	35
4.2	Natančnost zaznavanja označevalnika	37

KAZALO

4.3 Natančnost lokalizacije	43
4.4 Natančnost trajektorije	48
5 Zaključek	53

Seznam uporabljenih kratic

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

Povzetek

Lokalizacija igra pomembno vlogo v številnih aplikacijah, na primer filmska in igračarska industrija uporabljata lokalizacijske sisteme za zajem gibanja telesa.

V diplomskem delu implementiramo postopek lokalizacije v prostoru s pomočjo kamer. Objekt, ki ga lokaliziramo, je okrogel svetleči barvni označevalnik. Kameram določimo notranje in zunanje parametre ter jih tako umestimo v prostor, pri čemer notranje parametre določimo z MATLAB-ovim orodjem za kalibracijo kamer, zunanje pa izračunamo s pomočjo metode DLT in SVD. V realnem času zaznavamo označevalnik na slikah in s postopkom triangulacije določimo njegov položaj v prostoru. Sistem tudi ustrezno evalviramo in ocenimo napake lokalizacije. Izkaže se, da so napake v sistemu večinoma sistematične in bi jih v praksi lahko popravili.

Ključne besede: lokalizacija, triangulacija, večkamerni sistem, računalniški vid, kalibracija kamer, DLT, SVD.

Abstract

Localization plays a big role in many applications for example film and game industry are using localization systems for capturing body movement.

In this thesis we implement localization procedure using cameras. Localization object is a light emitting spherical color marker. We compute cameras intrinsic and extrinsic parameters effectively fitting camera in space. Intrinsic parameters are determined using MATLAB camera calibration tool. Extrinsic parameters are calculated using DLT and SVD methods. From images we detect marker in real time and use triangulation to determine its position in space. We properly evaluate the system and measure localization accuracy. It turns out that system errors are mostly structured and in practice could be fixed.

Keywords: localization, triangulation, multi-camera system, computer vision, camera calibration, DLT, SVD.

Poglavlje 1

Uvod

1.1 Motivacija

Vsi avtonomni roboti morajo imeti neko vrsto lokalizacije. Lokacija jim pomaga pri načrtovanju in izvedbi akcij, zato je pomembna zanesljivost, hitrost in natančnost lokaliziranja. Poleg robotike še veliko drugih področij za svoje aplikacije izkorišča lokaliziranje (npr. sledenje ljudi v javnosti). Lokaliziranje je že dolgo znan problem in zanj je bilo razvitih že veliko rešitev, ki pa se razlikujejo po natančnosti, težavnosti in ceni implementacije. Najbolj poznan sistem za globalno lokaliziranje (pozicioniranje) je GPS (*angl. Global Positioning System*). Namenjen je globalnemu določanju položaja, zato se najpogosteje uporablja pri navigaciji v transportu. V uporabi je že od leta 1978 [13], zato deluje robustno, ima pa tudi pomanjkljivosti. Za največjo natančnost mora signal iz satelitov neovirano prispeti do GPS sprejemnika. Običajna natančnost lokaliziranja z GPS sistemom je okoli $5 - 20\text{ m}$. Obstajajo še drugi sistemi, ki za lokaliziranje uporabljajo ultrazvok, Bluetooth, radijske in svetlobne signale ter kamere.

Glavna motivacija za temo diplomskega dela je sistem kvadrokopterjev, ki so ga razvili na ETH Zürich pod okriljem prof. Raffaella D'Andree [4]. Za lokalizacijo kvadrokopterjev uporabljajo enega od sistemov za zaznavanje gibanja (*angl. motion capture system*), ki pa je zelo drag. Ti sistemi so

namenjeni lokalizaciji v notranjem prostoru (*angl. IPS - Indoor Positioning System*), zato morajo biti zelo natančni in hitri.



Slika 1.1: Kvadrokopter s tremi odbojnimi označevalniki. Povzeto po [4].

Narediti želimo podoben sistem za lokaliziranje objekta v notranjem prostoru, vendar z uporabo veliko cenejših kamer.

1.2 Pregled področja

Lokacija se večinoma določa s pomočjo triangulacije [15], trilateracije [16] ali multilateracije [14]. Pri triangulaciji moramo poznati kot objekta na znano lokacijo (sidra), pri trilateraciji razdaljo od objekta do sidra, pri multilateraciji pa razliko razdalje od objekta do sidra v različnih časih. Sidra so lahko ultrazvočni in radijski oddajniki ali sprejemniki, mikrofoni, označevalniki, kamere itd. Le-ti služijo kot referenčne točke v prostoru, s katerimi se izračuna lokacija objekta.

Na Univerzi v Kaliforniji so razvili sistem, ki izračuna lokacijo s pomočjo zvoka [9]. Za sidra so uporabili mikrofone, objekt pa je bil mobilni telefon. V 97 % poskusov so položaj izmerili do 50 cm natančno. Na Univerzi v Yorku

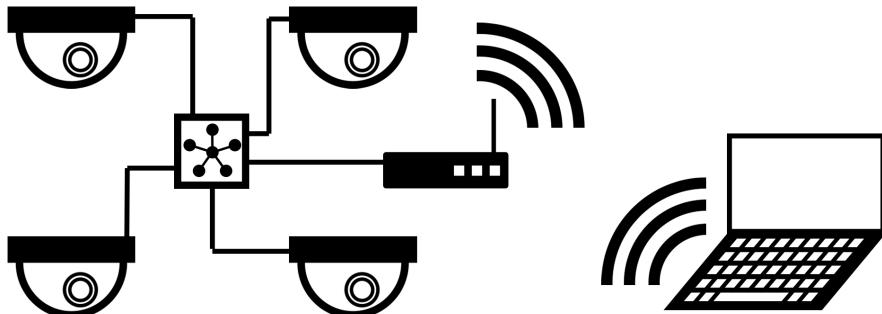
so naredili podobno z Wi-Fi signali in znamim modelom zaprtega prostora [3]. Na Univerzi v Parani pa so za merjenje razdalje uporabili ultrazvočni signal [1]. Natančnost njihovega sistema je nekaj desetink milimetra v $7 \times 7 \times 7 \text{ m}^3$ velikem prostoru.

Zgoraj opisana dela za lokalizacijo ne uporablja kamer. V nadaljevanju se bomo osredotočili na dela, ki jih. Slika iz navadne kamere sama po sebi ne pove razdalje do neke točke na sliki, ampak le smer žarka, ki pa implici-tno določa kot, ki ga uporabimo pri triangulaciji za določanje lokacije. Za triangulacijo potrebujemo vsaj dve slike, ki opazujeta isti objekt iz različnih zornih kotov.

Na Univerzi v Xi'an Jiaotongu so razvili stereo sistem za določanje položaja planarne tarče [8]. Uporabili so dve CCD (*angl. Charge-coupled Device*) [12] kamери, ki hkrati zajameta sliko ploskve in nato iz dobjenih točk trianguli-rata pozicijo ploskve v prostoru. Na Tehnični univerzi v Madridu so uporabili večkamerni sistem za določanje položaja ljudi v prostoru [11]. Najprej se-gmentirajo vse premikajoče se 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 [7], ki je od opisanih najbolj soroden diplomskemu delu. V prostor so postavili štiri visokoločljive omrežne kamere, ki so med seboj kalibrirane. Od štirih morata vsaj dve kamери zaznati objekt (v njihovem primeru vrtalnik), da se lahko določi njegovo lokacijo. Ko kamere prvič zaznajo objekt z ujemanjem pre-dlog, mu pričnejo slediti. S sledenjem oz. zaznavanjem izračunajo približno sredino objekta, ki jo z večimi kamerami izpopolnijo.

1.3 Cilji

Glavni cilj diplomskega dela je ustvariti večkamerni sistem za določanje položaja objekta v prostoru. Ker se lahko zgodi, da bo pri zaznavanju prihajalo do okluzij, morata vsaj dve kamери v nekem trenutku zaznati objekt. Za obdelavo slik se bo uporabil centralni računalnik, ki bo s kamerami povezan v lokalno omrežje. Zaradi lažjega zaznavanja, bo objekt 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. To omejitev pa se bo lahko v prihodnosti odstranilo in bo sistem zmožen hkrati določiti položaj veliko točkam v prostoru.



Slika 1.2: Oris večkamernega sistema.

1.4 Struktura diplome

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

Poglavlje 2

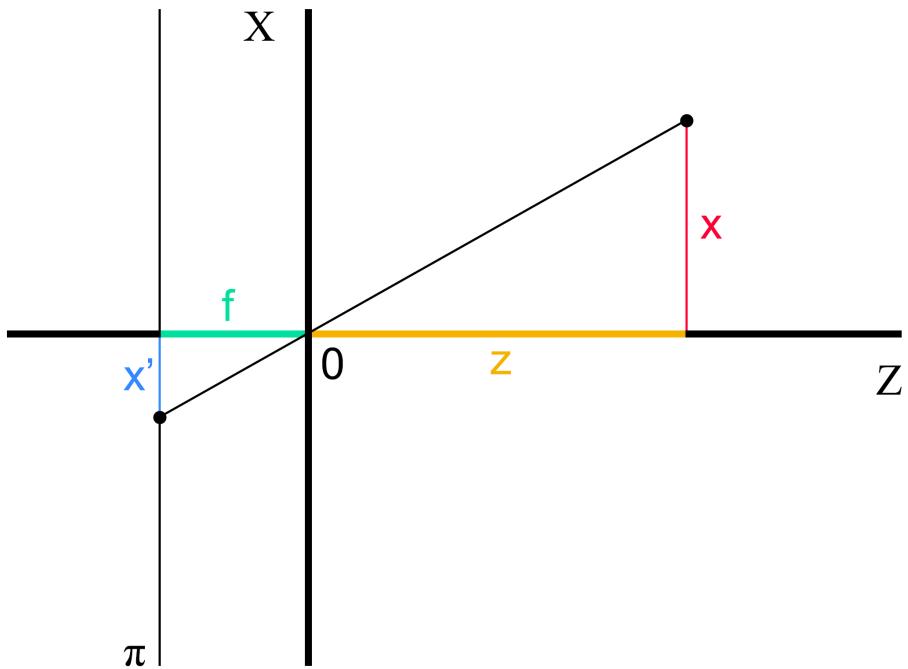
Teoretična podlaga

2.1 Parametri kamere

Ugotoviti moramo, kakšen matematični model najbolj ustreza današnjim kamерam. Večino kamer lahko modeliramo z modelom kamere z odprtino (*angl. pinhole camera*) [5, 17, 18]. Model opisuje kako se točka v svetu preslika v točko na sliki. Opišemo ga lahko s t.i. notranjimi in zunanjimi parametri kamere. V grobem, notranji parametri opisujejo interne lastnosti kamere, kot je goriščna razdalja, principalna točka, faktor poševnosti slikovnih pik itd., zunanji pa položaj kamere v prostoru.

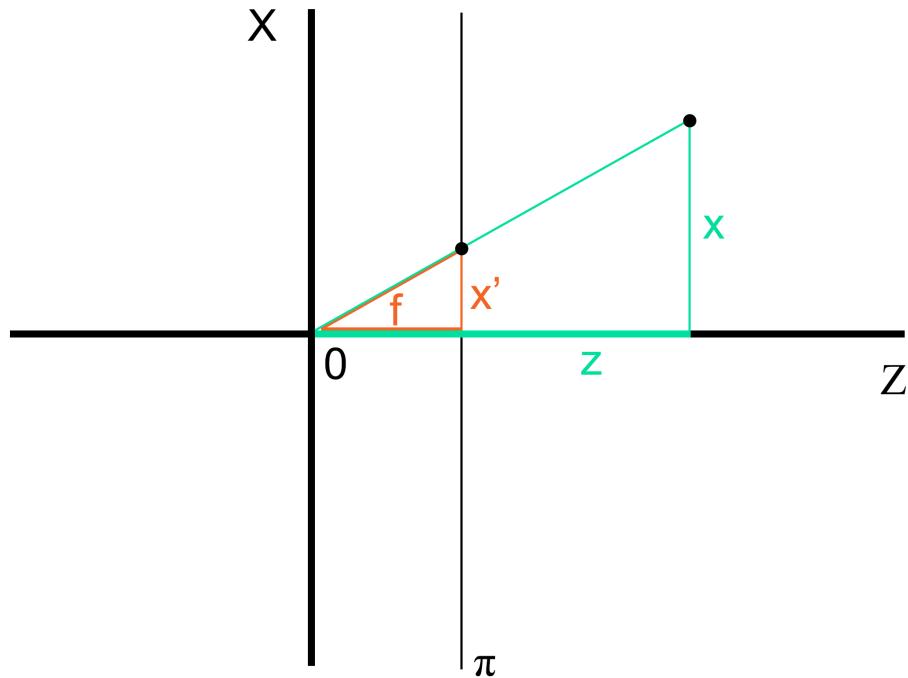
2.1.1 Notranji parametri

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



Slika 2.1: Točka v svetu se preslika na slikovno ploskev π .

Koordinatno izhodišče na sliki 2.1 predstavlja odprtino kamere. Višina točke v svetu je predstavljena z x , oddaljenost od kamere pa z z . Višina točke na sliki je označena z x' , goriščna razdalja pa s f . Točka v svetu se preko odprtine (koordinatnega izhodišča) prezrcali na slikovno ploskev π . Za lažje računanje pa lahko vpeljemo navidezno slikovno ploskev, kar je ponazorjeno na sliki 2.2.



Slika 2.2: Na sliki se jasno vidita 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 pomanjkljivosti, ki pa jih bomo postopoma odpravili. Enačbo

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

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

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 [5, 17].

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

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

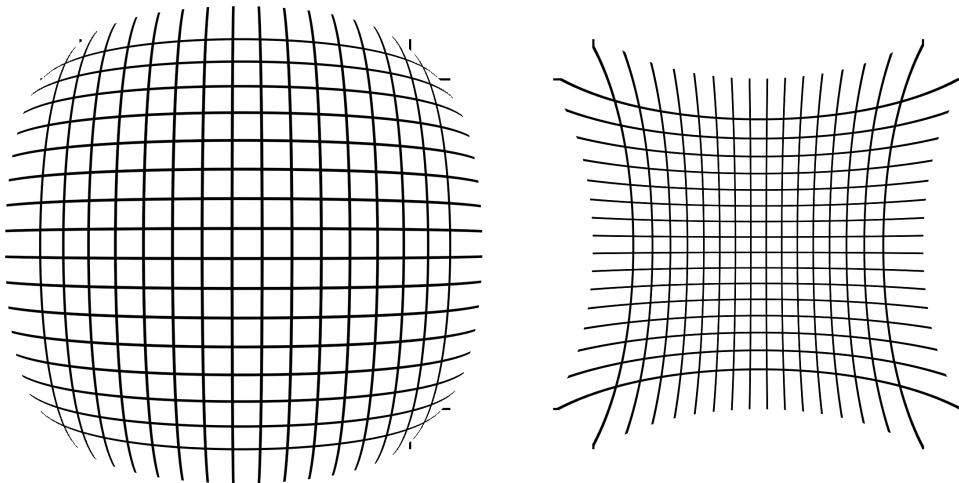
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} fm_x & s & u \\ 0 & fm_y & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.6)$$

Enačba (2.6) predstavlja notranji model kamere. Goriščna razdalja je označena s f , m_x in m_y predstavljata velikost, s 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, in sicer radialna in tangencična popačenost. Radialno popačenost povzroči oblika leče, poznamo pa dve glavni različici:

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

Obstaja še kombinacija obeh, ki se imenuje brkato popačenje (ker spominja na obliko brkov).



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

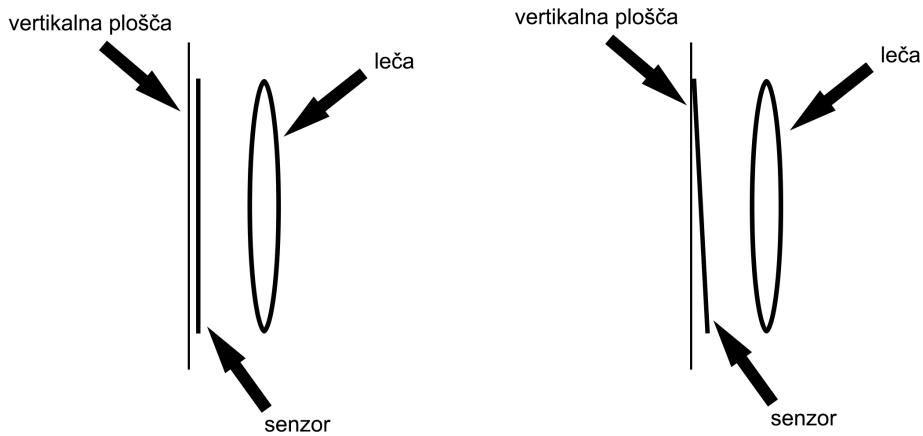
Moč radialne popačenosti slikovne točke je odvisna od razdalje do principalne točke. Modeliramo jo lahko z vsotami polinomov sodih stopenj. Za večino radialnih popačenj zadoščajo trije koeficienti [2, 5, 17].

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

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

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

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



Slika 2.4: Leva ilustracija prikazuje kamero brez tangencialne popačenosti. Na desni ilustraciji senzor ni vzporeden leči, kar povzroči tangencialno popačenost.

Modeliramo jo lahko kot [2]:

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

$$x_{popaceno} = x + (2p_1xy + p_2(r^2 + 2x^2)) \quad (2.11)$$

$$y_{popaceno} = y + (p_1(r^2 + 2y^2) + 2p_2xy). \quad (2.12)$$

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.

$$A = \begin{bmatrix} fm_x & s & u \\ 0 & fm_y & v \\ 0 & 0 & 1 \end{bmatrix}$$

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

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

2.1.2 Zunanji parametri

Do sedaj smo predpostavljali, da je koordinatno izhodišče sveta 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 napišemo malo drugače: iz enega koordinatnega sistema lahko dobimo kateri koli drug koordinatni sistem (z enakim številom dimenzij) tako, da izhodišči poravnamo s translacijo in nato rotiramo osi, da sovpadajo.

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

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

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} .

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

Zavedati se moramo, da rotacijska matrika R in translacijski vektor \vec{T} ne predstavlja rotacije in pozicije 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 in tako dobimo popoln model kamere.

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

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

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

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

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 bomo opisali, 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 se razlikujejo po hitrosti, težavnosti in natančnosti. Dve popularni metodi sta Tsaijev kalibracijski algoritem [6] in Zhangova fleksibilna tehnika za kalibracijo kamer [17]. V nalogi smo za kalibracijo notranjih parametrov uporabil MATLAB-ov kalibrator kamere [10], 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 in bi morala biti dovolj velika. Sistem mora delovati tudi, če imajo nekatere kamere med seboj minimalno ali celo prazno prekrivanje vidnega polja. V nadaljevanju bomo opisali teoretično podlago metode, ki smo jo uporabili za ocenjevanje zunanjih parametrov. Doseči želimo, da kameram določimo skupni koordinatni sistem sveta.

Metoda predpostavlja, da imamo za neko kameru že izračunane notranje parametre. Za oceno zunanjih parametrov je dovolj le ena slika iz kamere, ki pa ne sme biti popačena, zato moramo popačeno sliko najprej popraviti. Enačbi (2.9) in (2.12) določata neposredno preslikavo med popačenimi in nepopačenimi točkami.

Zunanje parametre določa rotacijska matrika R in translacijski vektor \vec{T} , kar je v enačbi (2.19) 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 (*angl. Direct Linear Transformation*) [5]. Ker računamo s homogenimi koordinatami, sta si leva in desna stran enačbe enaki do poljubnega neničelnega faktorja λ .

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

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

Z eno znano točko (na sliki in v svetu) dobimo tri enačbe, vendar pa je ena linearna kombinacija drugih dveh, zato nam pri ocenjevanju zunanjih parametrov ne pomaga. Oceniti moramo torej 12 neznank (ker je matrika $[R|\vec{T}]$ velika 3×4) z eno točko pa dobimo dve neodvisni enačbi, kar pomeni, da potrebujemo najmanj šest 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 λ , 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

(angl. *SVD - Singular Value Decomposition*). SVD razcepi matriko B na $U\Sigma V^T$, kjer je U matrika levih lastnih vektorjev, Σ 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 λ lastna vrednost. Če iz matrike V vzamemo vektor, ki ustreza najmanjši lastni vrednosti v matriki Σ , smo tako dobili netrivialno rešitev, ki najbolje zadošča pogoju $B\vec{p} = 0$. Rešitev je potrebno le še preoblikovati v 3×4 matriko zunanjih parametrov. V poglavju 3.3 naslovimo 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 štiri, saj ocenjujemo štiri parametre, ki določajo X . Ena točka na sliki nam poda dve linearne neodvisne enačbe. Potrebujemo torej vsaj dve 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.

Poglavlje 3

Implementacija

3.1 Strojna oprema

3.1.1 Računalnik

Vsa obdelava podatkov se odvija na enem prenosnem računalniku, zajemanje slik pa poteka sočasno v ločenih procesih. Računalnik, ki ima dve fizični procesni jedri in štiri niti, je 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

V nalogi uporabljamo štiri Axis 215 PTZ omrežne kamere. To so varnostne IP kamere, primarno namenjene nadzoru okolja. Kamere so povezane v

zvezdišče na omrežju ethernet, zvezdišče pa je povezano z brezžičnim usmerjevalnikom (slika 1.2). Kamere so zmožne v realnem času preko lokalnega omrežja posredovati do 30 slik na sekundo pri ločljivosti 704×576 . Vseeno pa varnostne kamere niso namenjene sinhroniziranemu zajemanju slik in ne podpirajo skupnega prožilca. Največja ločljivost je 704×576 , 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 (*angl. Common Gateway Interface*) vmesnik, ki omogoča nadzor funkcionalnosti preko protokola HTTP (*angl. 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 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 kamери.

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 }
```

3.2 Programska oprema

Delo smo začeli v programskem okolju 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 MATLAB-u smo razvili konceptno rešitev, ki jo kasneje implementiramo v Python aplikaciji.

Za zajemanje statičnih slik iz kamer smo uporabili Node.js v0.12.3. Node.js je JavaScript pogon z veliko različnimi moduli. Njegova arhitektura je po zasnovi asinhrona, kar pri vhodno/izhodnih operacijah močno pohitri sistem.

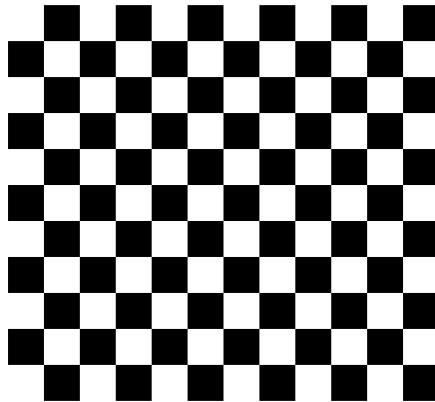
Za iskanje kamer v omrežju smo uporabili Axisov IPUtility, ki samodejno vrne IP naslove vseh kamer, ki so priključene v omrežje.

Za implementacijo glavnega sistema smo uporabili 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 aplikacij. Funkcije v knjižnicah so zaradi hitrosti implementirane v nižjenivojskih jezikih kot sta C in 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

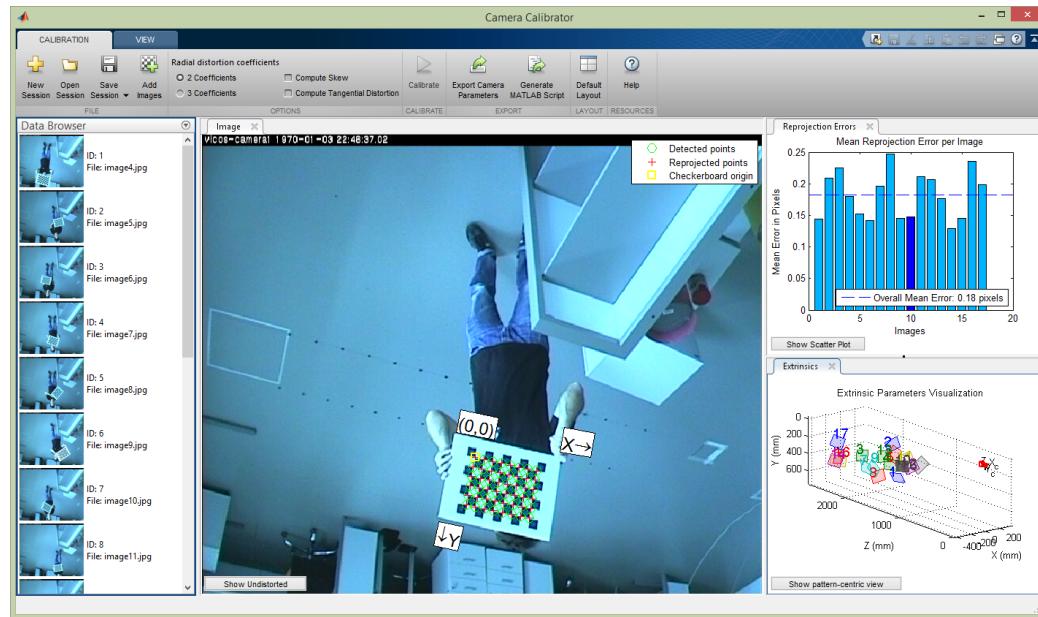
Za ocenjevanje notranjih parametrov uporabimo MATLAB-ov kalibrator kamere. Za kalibracijo mu moramo podati slike šahovnice, ki so bile zajete s kamero, ki jo želimo kalibrirati. Širina šahovnice mora biti različna od višine, da lahko enolično določimo njen orientacijo.



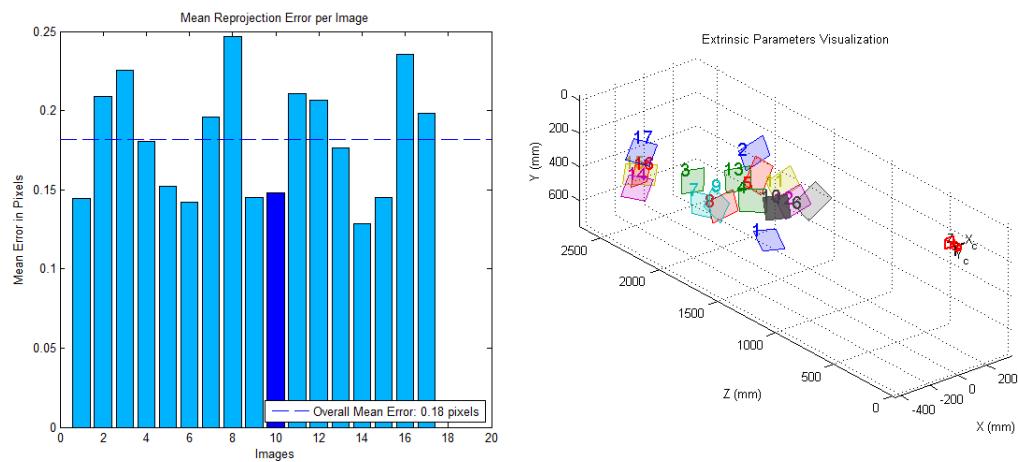
Slika 3.2: Šahovnica velikosti 11×12 .

Za učinkovito kalibracijo notranjih parametrov moramo zajeti okoli 10 - 20 slik z različnimi orientacijami in translacijami 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 dva);
4. obkljukamo ali želimo oceniti poševnost in tangencialno popačenost;
5. kalibriramo;
6. postopek lahko ponavljamo s podmnožico slik na podlagi reprojekcijske napake, in
7. ocenjene notranje parametre izvozimo za uporabo.



Slika 3.3: Na sliki se vidi zaznane robove šahovnice in njena orientacija, graf reprojekcijske napake (razlaga v 4.1) in vizualizacija zunanjih parametrov.



Slika 3.4: Graf reprojekcijske napake za vsako sliko (levo) in vizualizacija pozicij šahovnic s središčem v kameri (desno).

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

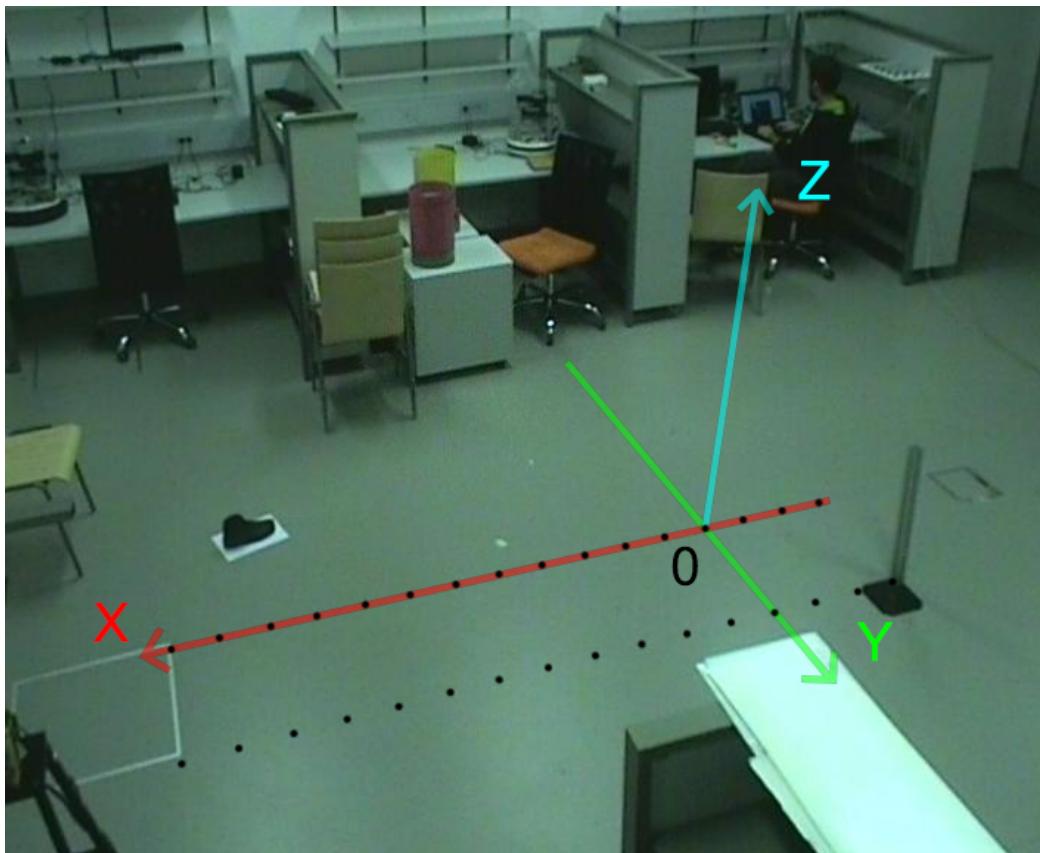
$$\vec{r} = [-0, 2934 \ 0, 1331] \quad (3.2)$$

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

Slika 3.5: 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 skupni koordinatni sistem sveta. Določiti moramo rotacijsko matriko R in translacijski vektor \vec{T} , ki predstavlja kje se nahaja koordinatni sistem relativno 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. V prostoru smo določili in označili koordinatni sistem. Med oznakami na osi x je razdalja 20 cm, na osi y pa 60 cm. Enota je 1 cm.



Slika 3.6: 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 šest parov, ker moramo oceniti 12 parametrov (devet za rotacijsko matriko in tri za translacijo), en par pa nam določi dve linearne neodvisne enačbi. Točke v svetu določimo sami, nato pa jih ročno ali avtomatsko označimo na sliki. 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 3.6.

V tabeli 3.3 se lepo vidi, da so vse točke v svetu koplanarne, saj je pri vseh $z = 0$. Sedaj lahko določimo matriko B , ki je za en 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 rešimo sistem $B\vec{p} = 0$ in dobljeno rešitev preoblikujemo v 3×4 matriko zunanjih parametrov. Rešitev je pravzaprav deveti stolpec matrike V , ki jo vrne SVD in ne dvanajsti, kot je opisano pri izpeljavi teorije. To je posledica koplanarnosti svetovnih točk. Matrika B je ranga 9, ker je komponenta z

vseh točk v svetu enaka 0. Spodnja matrika prikazuje primer rešitve za pare točk iz tabele 3.3.

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

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.

$$[R|\vec{T}] = [\vec{R}_1 \ \vec{R}_2 \ \vec{R}_3 \ \vec{T}] \sim \frac{1}{\|R_1\|} \begin{bmatrix} -0.0018 & 0.0006 & 0 & 0.1989 \\ 0.0003 & 0.0007 & 0 & 0.1607 \\ 0.0005 & 0.0016 & 0 & -0.9668 \end{bmatrix} \quad (3.5)$$

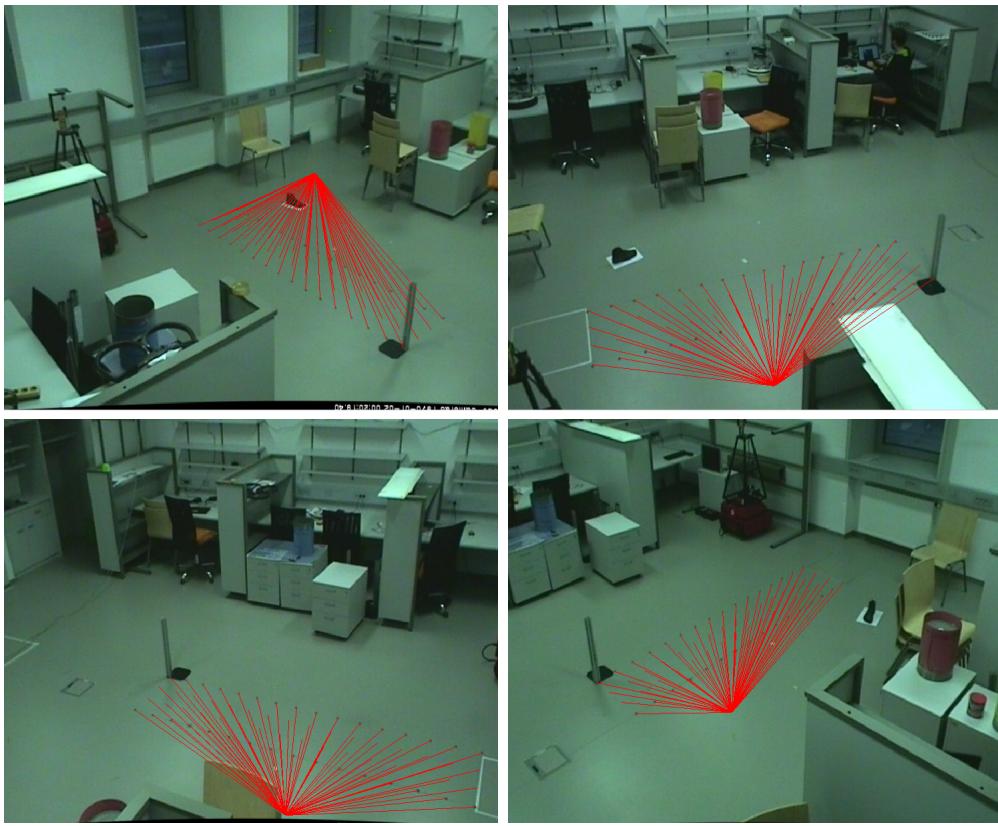
$$[R|\vec{T}] \sim \begin{bmatrix} -0.9544 & 0.3003 & 0 & 106.6226 \\ 0.1419 & 0.3877 & 0 & 86.1440 \\ 0.2627 & 0.8810 & 0 & -518.2979 \end{bmatrix} \quad (3.6)$$

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

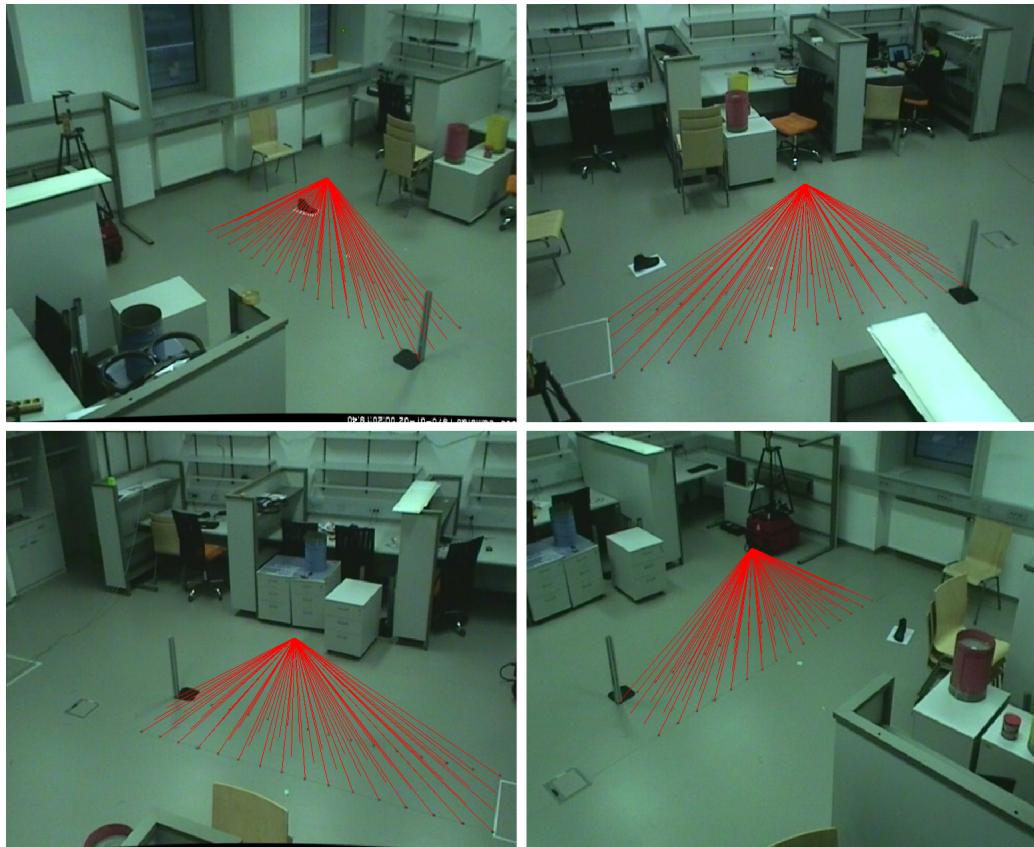
$$R_3 = R_1 \times R_2 = \begin{bmatrix} -0.9544 \\ 0.1419 \\ 0.2627 \end{bmatrix} \times \begin{bmatrix} 0.3003 \\ 0.3877 \\ 0.8810 \end{bmatrix} \quad (3.7)$$

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

Zdaj moramo le še določiti pravo smer novo izračunanega vektorja. R_3 kot tudi $-R_3$ sta legalni rešitvi in implicitno določata sučnost koordinatnega sistema. Pravilno rešitev določimo ročno na podlagi projekcije točk, kot je vidno iz slike 3.7.



Slika 3.7: Projekcija piramide na sliko pokaže, da so zadnji trije koordinatni sistemi levošučni, zato moramo obrniti smer zadnjega stolpca matrike R .



Slika 3.8: Na novo ocenjeni zunanji parametri pravilno modelirajo kamere, kar je razvidno iz projekcije piramide.

Končna matrika zunanjih parametrov za točke iz tabele 3.3 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 R prisilno ortonormirali, bi s tem vnesli dodatno napako v sistem.

3.4 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 predstavljeno 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). Vrednost, ki ločuje segmente od ozadja se imenuje prag in se določi na podlagi histograma slike.

Pri segmentiranju barvnih slik se je potrebno odločiti za primeren barvni model. Primarno je barva v slikah predstavljena z modelom RGB (*angl. Red, Green, Blue*). Običajno je bolj primeren model HSV (*angl. Hue, Saturation, Value*) ali HSL (*angl. Hue, Saturation, Lightness*), saj H (ton) 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.

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:

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y). \quad (3.10)$$

Poznamo še centralne momente, ki so definirani kot:

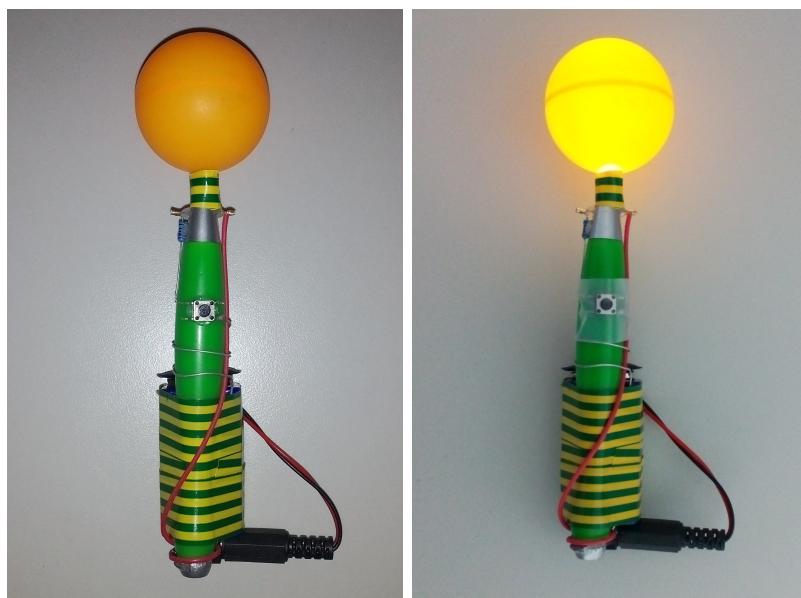
$$\bar{x} = \frac{M_{10}}{M_{00}} \quad (3.11)$$

$$\bar{y} = \frac{M_{01}}{M_{00}} \quad (3.12)$$

$$\mu_{ij} = \sum_x \sum_y (x - \bar{x})^i (y - \bar{y})^j I(x, y). \quad (3.13)$$

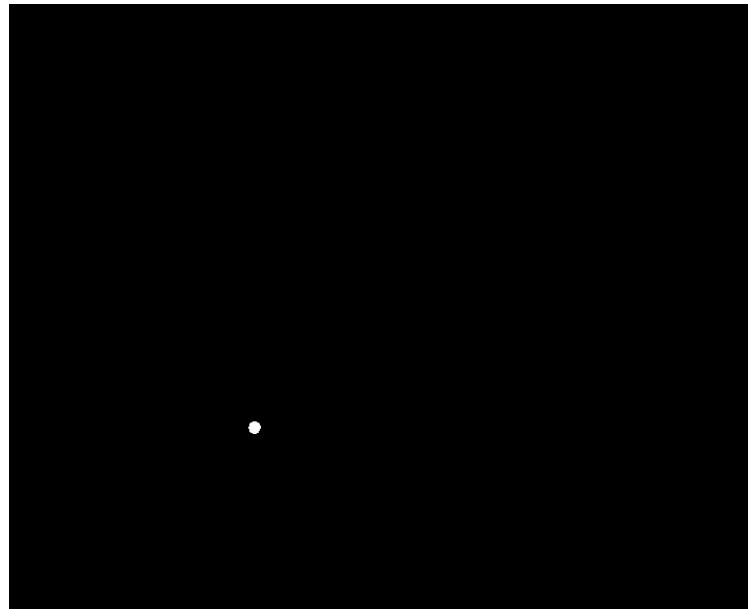
Masno središče je predstavljeno z \bar{x} in \bar{y} . Moment M_{00} je enak površini segmenta, M_{10} ter M_{01} pa sta vsoti segmenta po x in y koordinatah.

Označevalnik je v osnovi oranžna žogica za namizni tenis. Da pa bi sprememba svetlobe čim manj vplivala na natančnost zaznavanja smo v žogico vgradili modro svetlečo diodo (*angl. LED - Light Emitting Diode*) in stikalo. S pritiskom na stikalo prižgemo označevalnik, da ga kamere lahko zaznajo. Kamere zaznajo žogico kot izrazito rumene barve (slika 3.9).



Slika 3.9: Ugasnjen označevalnik je oranžne barve (levo), prižgan pa izrazito rumene barve (desno).

Prag, ki je pravzaprav pas, smo določili z analizo barve slikovnih elementov označevalnika. Za grobo oceno smo vzeli minimalne in maksimalne vrednosti izmerjenih barv. Ker je označevalnik izrazito rumene barve se model RGB izkaže za boljšega kakor model HSV. Interval smo spremenili do teme, da smo se znebili napačnih detekcij, vendar ohranili natančnost v čim večji meri. 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. Na slikah 3.10 in 3.11 sta prikazani posamezni fazи zaznavanja.



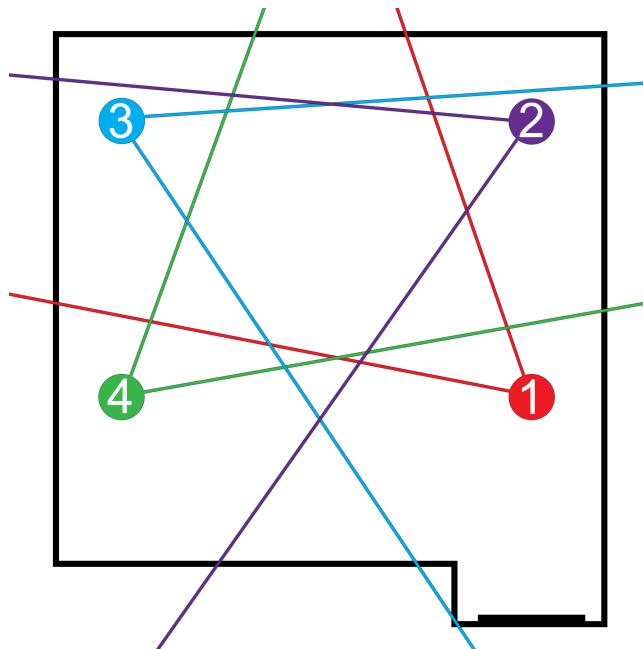
Slika 3.10: Segmentacija zajete slike z belo barvo označuje zaznan označevalnik.



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

3.5 Postavitev sistema

V prostoru z dimenrijami $7,4 \times 7,7 \times 2,97 \text{ m}^3$ so bile na strop pritrjene 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 omrežja ethernet povezane v zvezdišče in nato v brezžični usmerjevalnik. Prenosni računalnik je v omrežje povezan brezžično. Topologija povezave je predstavljena na sliki 1.2. Na računalniku se izvaja glavni program, ki komunicira s kamerami in računa položaj označevalnika v prostoru. Proses zaznavanja položaja je razdeljen v preproste korake, ki jih opisujeta psevdo kodi 2 in 1.

Data: parametri in atributi kamere

Result: pozicija označevalnika na sliki

inicializiraj kamero;

while *True* **do**

zajemi sliko iz kamere;

zaznaj označevalnik;

pošlji točko glavnemu procesu;

end

Algorithm 1: Psevdo koda podprocesov, ki zaznavanje označevalnik na slikah.

Data: število kamer n

Result: lokacija označevalnika v prostoru

zaženi n podprocesov;

while *True* **do**

foreach *podproces* **do**

sprejmi točko označevalnika;

dopolni matriko za triangulacijo;

if *vsaj dve kamери zaznata označevalnik* **then**

trianguliraj položaj označevalnika v prostoru;

izpiši položaj;

end

end

zajemi sliko iz kamere;

zaznaj označevalnik;

pošlji točko glavnemu procesu;

end

Algorithm 2: Psevdo koda glavnega procesa, ki lokalizira označevalnik v prostoru.

Poglavlje 4

Eksperimentalni rezultati

4.1 Reprojekcija kalibracijskih točk

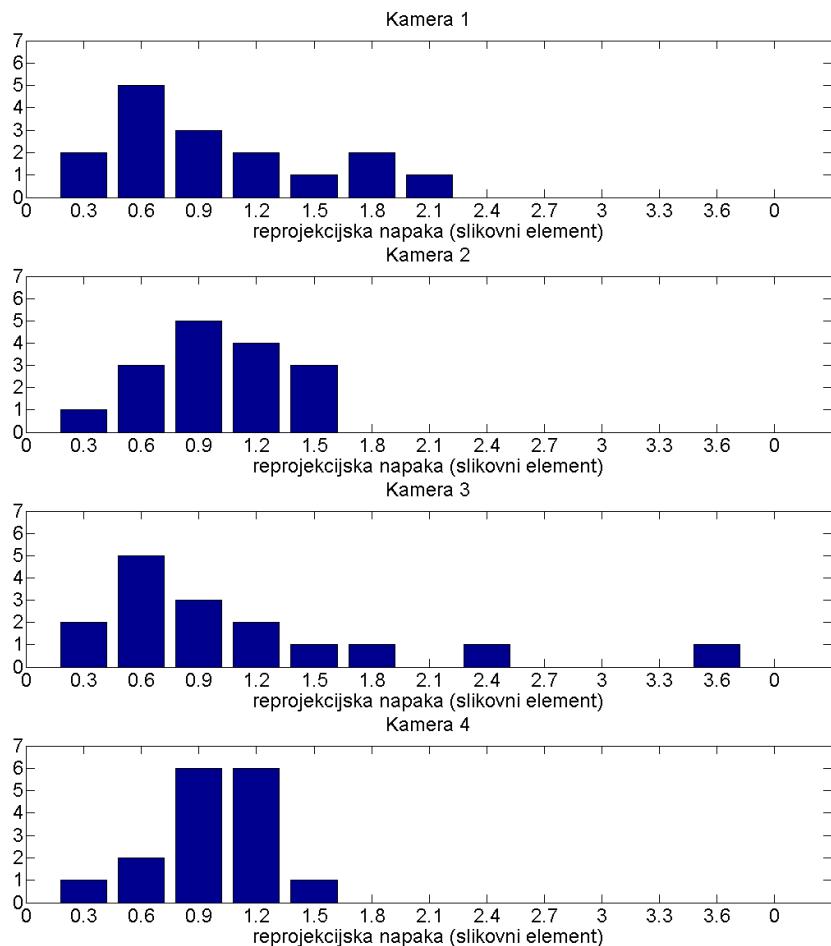
Izmeriti želimo kako dobro modeli kamer določajo kalibracijske točke iz tabele 3.3. Pri kalibraciji smo določili pare točk (X, X') , kjer je X točka v svetu, X' pa njena projekcija na sliko. S parametri kamere lahko poljubno točko v svetu preslikamo na sliko. Tako lahko točko X projeciramo na sliko in izračunamo reprojekcijsko napako \vec{e} , ki je definirana kot:

$$\vec{e} = A[R|\vec{T}]X - X'. \quad (4.1)$$

Povprečna reprojekcijska napaka pri kalibriranju notranjih parametrov je med 0.14 in 0.16 slikovnega elementa. To je posledica selektivne izbire slik s

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

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



Slika 4.1: Histogrami reprojekcijskih napak pri kalibraciji zunanjih parametrov.

katerimi načrtno zmanjšamo napako. Napaka manjša od slikovnega elementa je pri delu s slikami zelo dobra.

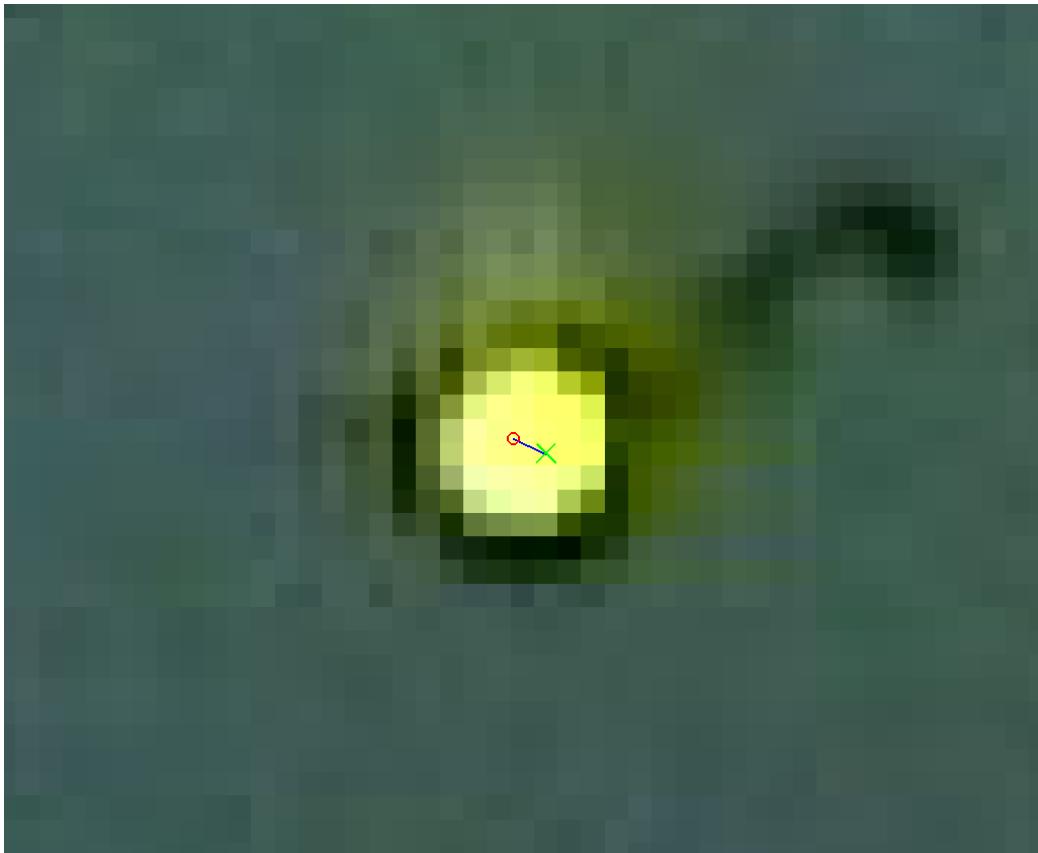
Za kalibracijo zunanjih parametrov, so bile slikovne točke v tabeli 3.3 določene ročno z ločljivostjo meritve 1 slikovnega elementa. Pomemben faktor pri tem postopku je vnos človeške napake. Sistem se je kljub napakam kalibriral tako, da je vsota razlik kvadratov najmanjša. V tabeli 4.1 in na sliki 4.1 opazimo, da so povprečne napake dokaj konsistentne pri okoli 0,9 slikovnega elementa. Pri tretji kamери izstopa varianca, kar je najverjetnejše posledica manj natančnega ročnega določanja točk.

4.2 Natančnost zaznavanja označevalnika

Natančnost zaznavanja označevalnika lahko izmerimo tako, da primerjamo zaznano točko s točno (resnično) točko označevalnika (slika 4.2). Formalno to zapišemo kot:

$$\vec{e} = \vec{x}_{zaznana} - \vec{x}_{tocna}. \quad (4.2)$$

Natančnost izmerimo na dveh vrstah množic. Prva ima na vseh slikah statično pozicijo označevalnika, saj je bil v času zajemanja slik označevalnik na istem mestu v prostoru. Zaradi šuma pri zajemu slik pa algoritem za zaznavanje označevalnika ne zazna vedno enake točke na sliki. Druga vrsta množice pa vsebuje slike, ki prikazujejo označevalnik vsakič na drugem mestu v prostoru (je dinamičen). Vsaka množica vsebuje okoli 100 slik, po 25 iz vsake kamere. Slike na katerih označevalnik ni viden so bile odstranjene iz množice (20 odstranjenih slik iz dinamične množice). Točne točke označevalnika so bile določene ročno z ločljivostjo 1 slikovnega elementa, ker MATLAB-ovo orodje za označevanje točk ne dovoljuje večje natančnosti.

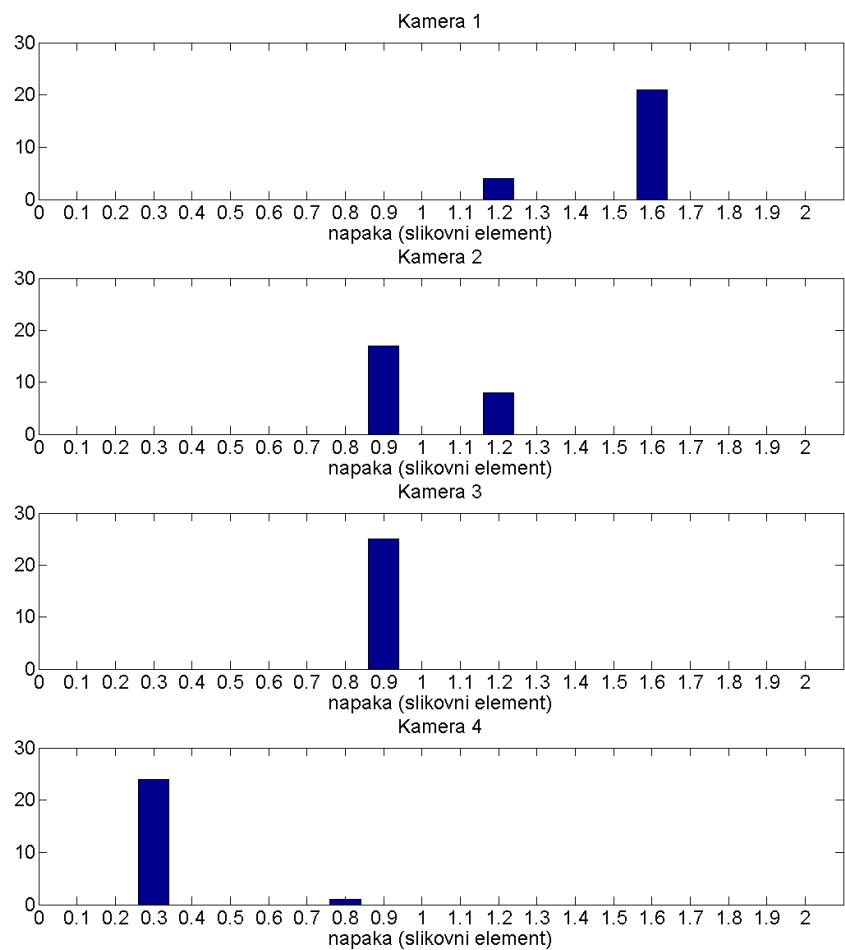


Slika 4.2: Resnična točka je označena z rdečo krožnico, zaznana pa z zelenim križem. Vektor napake je predstavljen z modro daljico.

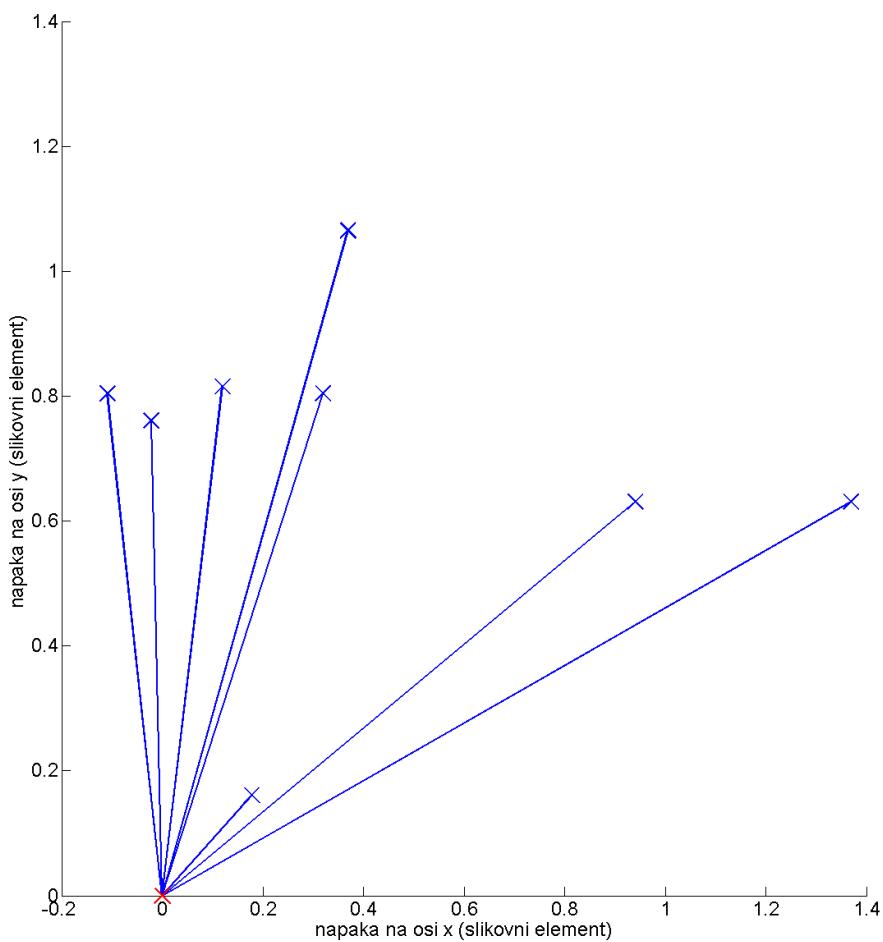
Povprečje napake v prvi množici (statični) je 0, 8614 slikovnega elementa z varianco 0, 1899. Iz histogramov na sliki 4.3 se lepo vidi, da je varianca majhna. Slike iz tretje in četrte kamere skoraj vedno določijo enako točko označevalnika. Pri slikah iz prve in druge kamere pa šum povzroči majhno spremembo intenzitet pik, kar posledično zmede algoritem za zaznavanje označevalnika.

Povprečje napake v drugi množici (dinamični) je 0, 852 slikovnega elementa z varianco 0, 2124. Iz histogramov na sliki 4.5 vidimo, da je napaka bolj enakomerno porazdeljena kot pri zaznavanju statičnega označevalnika.

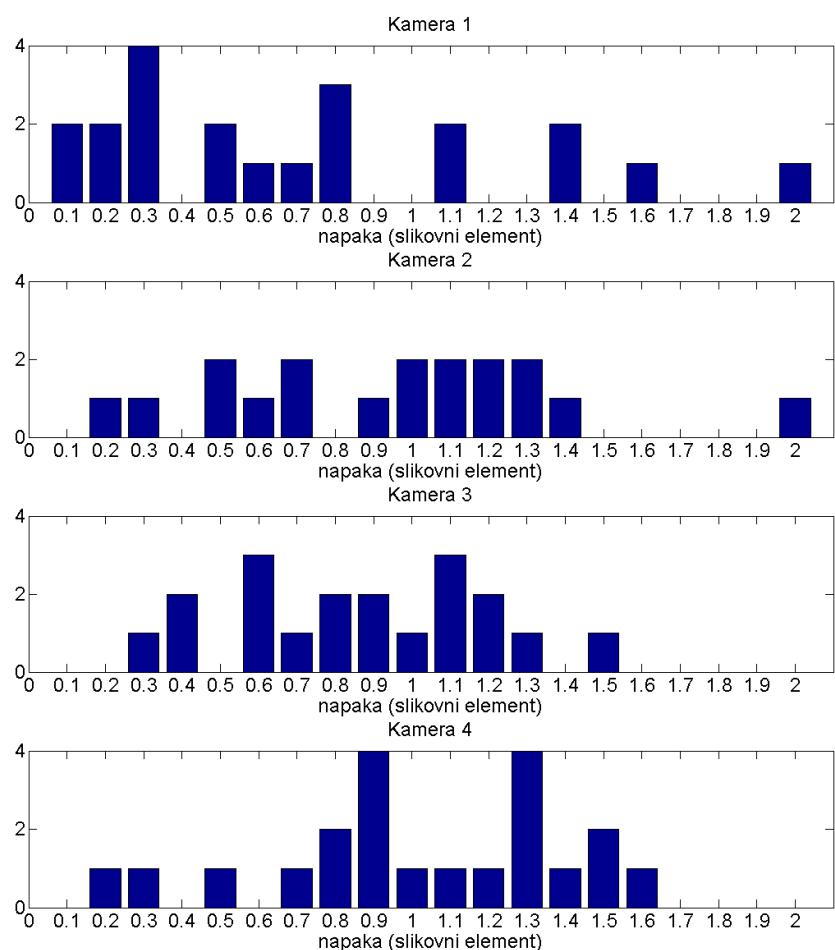
Vektorji napak na slikah 4.4 in 4.6 pa so bolj gosto porazdeljeni v prvem kvadrantu. V času meritev je bila v prostoru prisotna usmerjena svetloba, ki



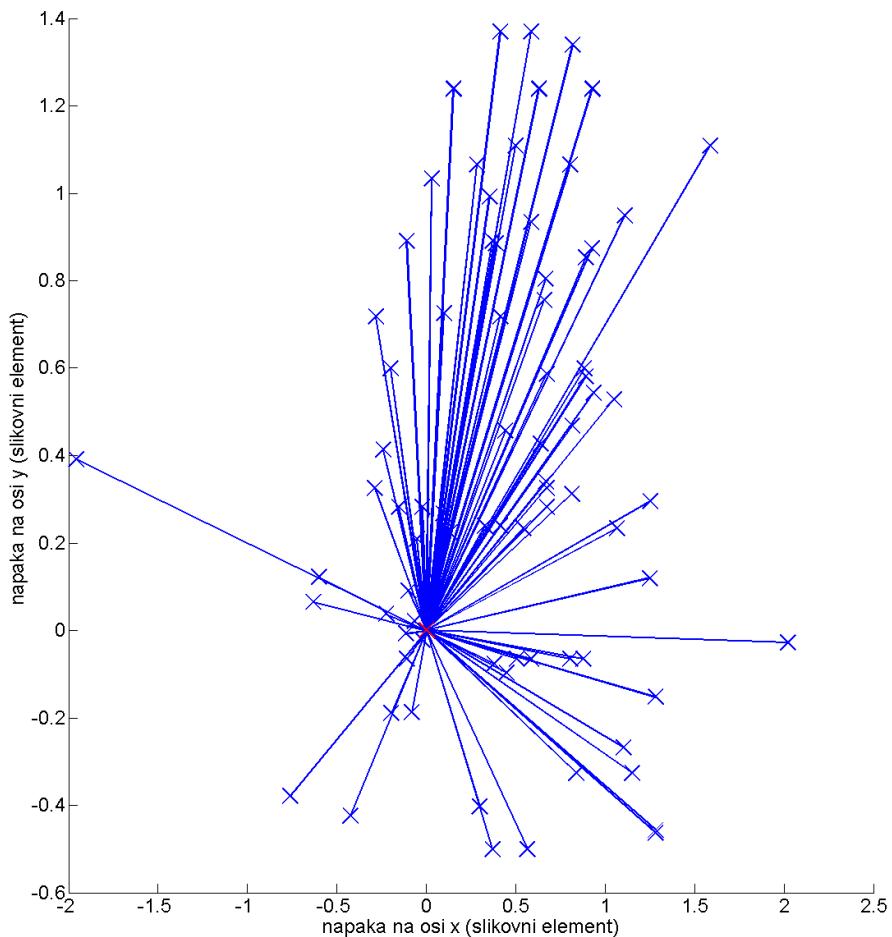
Slika 4.3: Histogrami napak zaznavanja statičnega označevalnika.



Slika 4.4: Graf vektorjev napak v statični množici. Vektorji so usmerjeni predvsem v prvi kvadrant.



Slika 4.5: Histogrami napak zaznavanja dinamičnega označevalnika.



Slika 4.6: Graf vektorjev napak v dinamični množici. Gostota zaznanih točk je večja v zgornji polovici grafa, kar je najverjetneje posledica rahle neenakomerne osvetljenosti označevalnika.

je povzročila rahlo neenakomerno osvetlitev označevalnika.

4.3 Natančnost lokalizacije

Natančnost lokalizacije določimo tako, da na izbrane poljubne točke v prostoru postavimo označevalnik in trianguliramo njegovo lokacijo. Dobljeno pozicijo primerjamo z izbrano in tako dobimo napako \vec{e} .

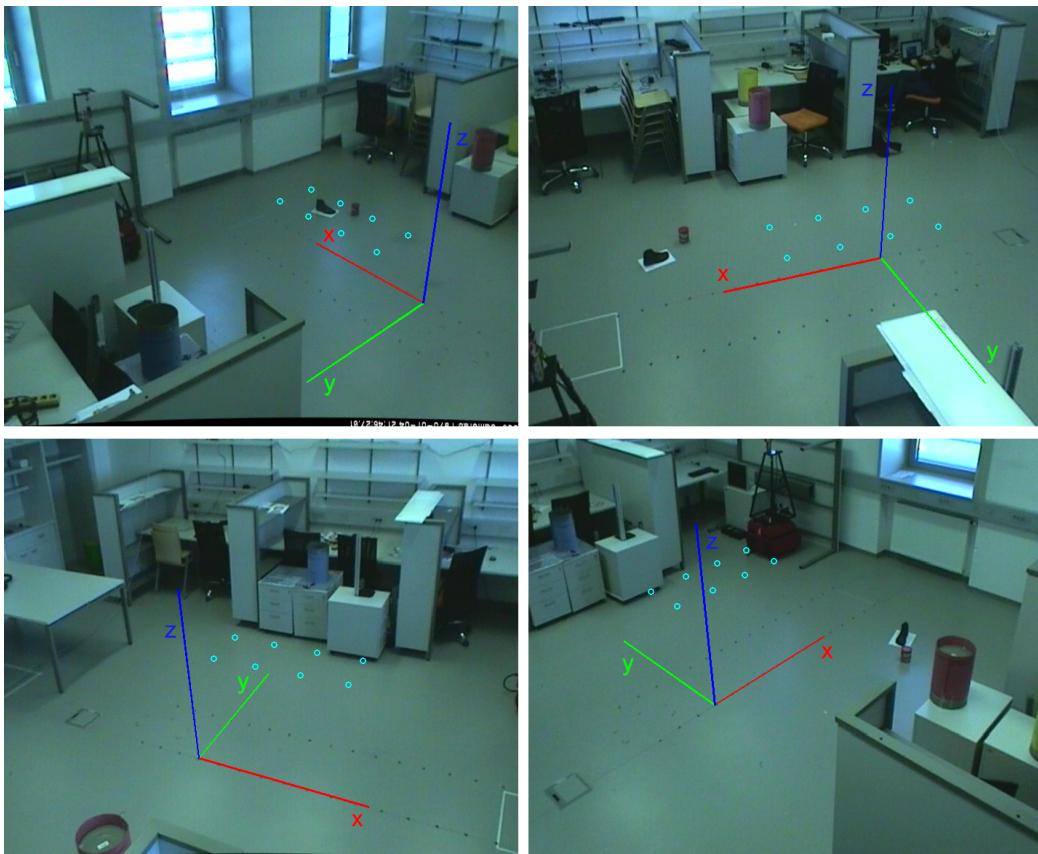
$$\vec{e} = \vec{X}_{trianguliran} - \vec{X}_{tocen} \quad (4.3)$$

Zunanji parametri kamer so bili kalibrirani s 16 koplanarnimi točkami v svetu (tabela 3.3). Za testiranje natančnosti lokaliziranja pa izberemo osem točk, prikazanih v tabeli 4.2. Napake izračunamo pri triangulaciji točk z dvemi, tremi in štirimi kamerami. Izbrane točke so vizualizirane v prostoru na sliki 4.14.

x (cm)	y (cm)	z (cm)
0	40	76
0	80	76
40	40	76
40	80	76
80	40	76
80	80	76
120	40	76
120	80	76

Tabela 4.2: Referenčne točke za meritve v prostoru.

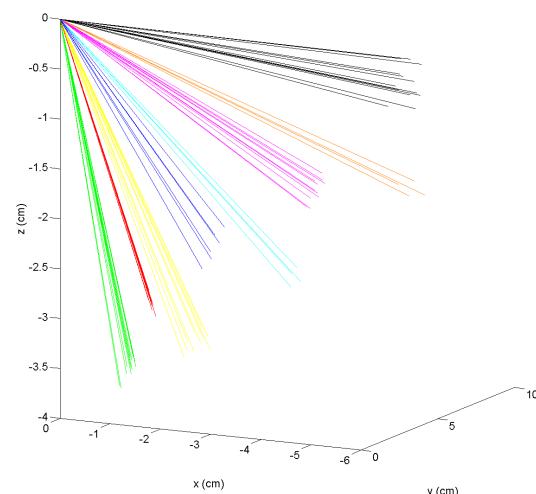
Na slikah 4.8, 4.10 in 4.12 so vizualizirani vektorji napak za meritve iz tabele 4.3. Barve vektorjev predstavljajo različne točke iz tabele 4.2. Hitro lahko opazimo, da napake niso naključne ampak sistematične. Na slikah 4.9, 4.11 in 4.13 pa so prikazani histogrami napak. Pri vseh meritvah je največja napaka na osi y . Kalibracijske točke za zunanje parametre kamer (tabela 3.3)



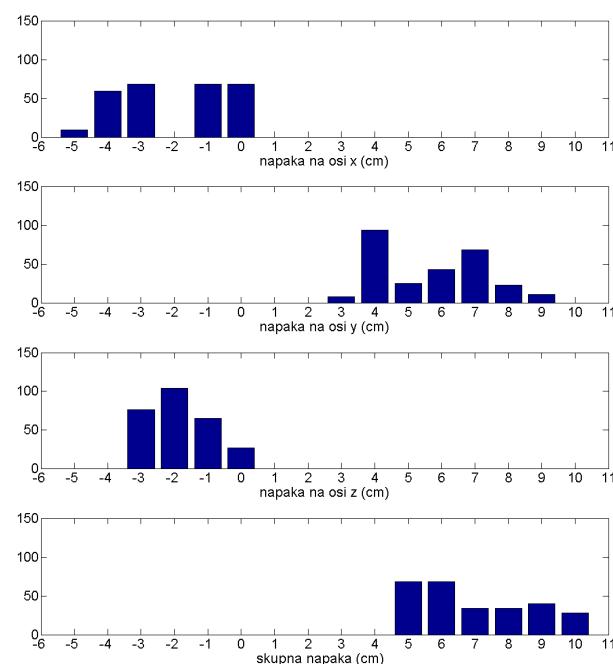
Slika 4.7: Izbrane točke so predstavljene s krogci. Vektorji x, y in z označujejo koordinatni sistem prostora.

št. kamer	št. meritev	povprečna napaka (cm)	varianca napake (cm ²)
2	272	6,4854	2,6562
3	408	2,5539	1,1009
4	712	4,4618	0,2218

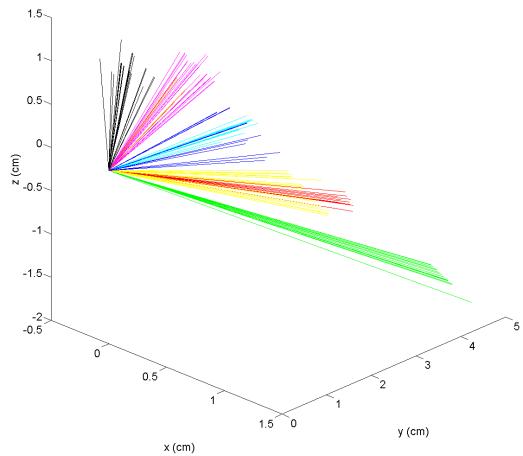
Tabela 4.3: Povprečna napaka pri pozicioniranju z dvemi, tremi in štirimi kamerami.



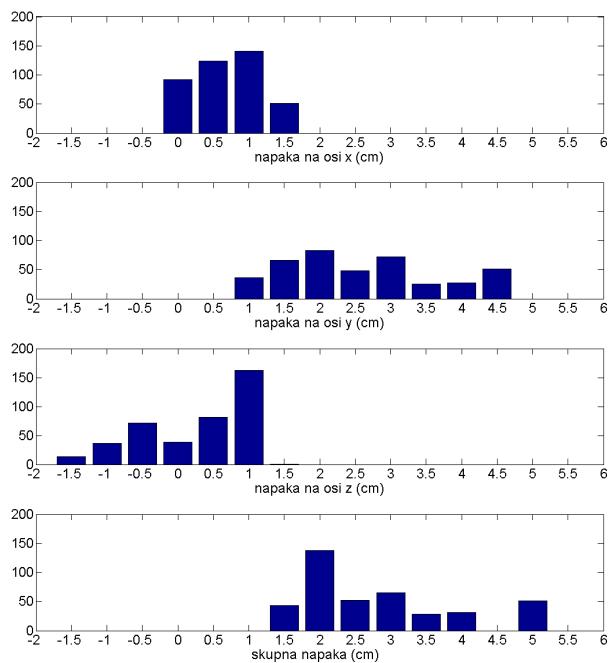
Slika 4.8: Vektorji napak pri pozicioniranju z uporabo dveh kamer.



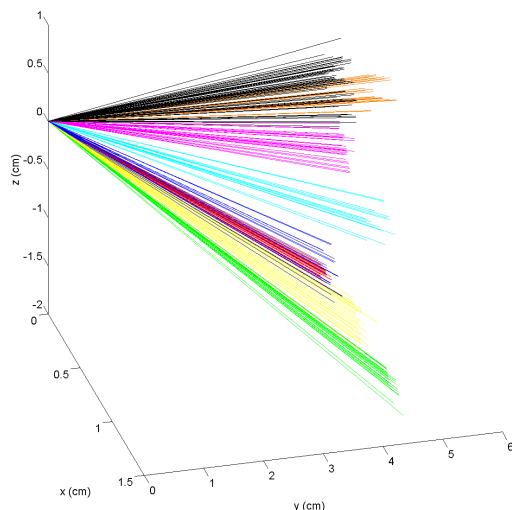
Slika 4.9: Histogrami napak pri pozicioniranju z uporabo dveh kamer.



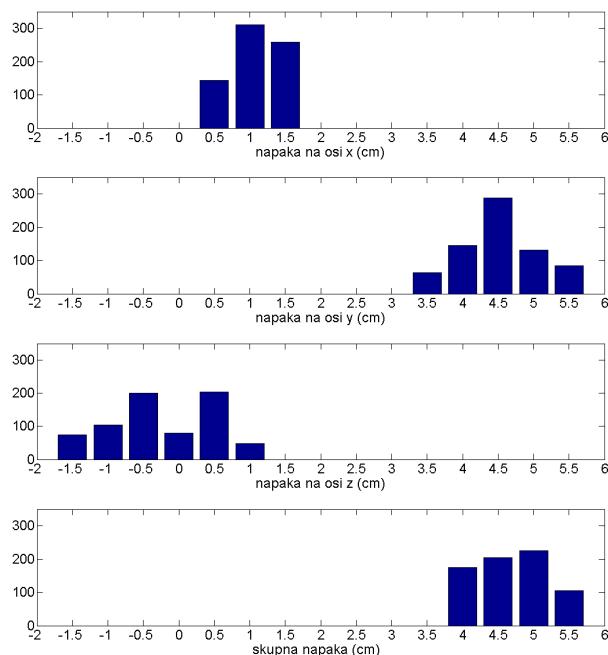
Slika 4.10: Vektorji napak pri pozicioniranju z uporabo treh kamer.



Slika 4.11: Histogrami napak pri pozicioniranju z uporabo treh kamer.



Slika 4.12: Vektorji napak pri pozicioniranju z uporabo štirih kamer.



Slika 4.13: Histogrami napak pri pozicioniranju z uporabo štirih kamer.

imajo veliko večjo varianco v smeri x , zato morda y komponenta ni dovolj natančno določena.

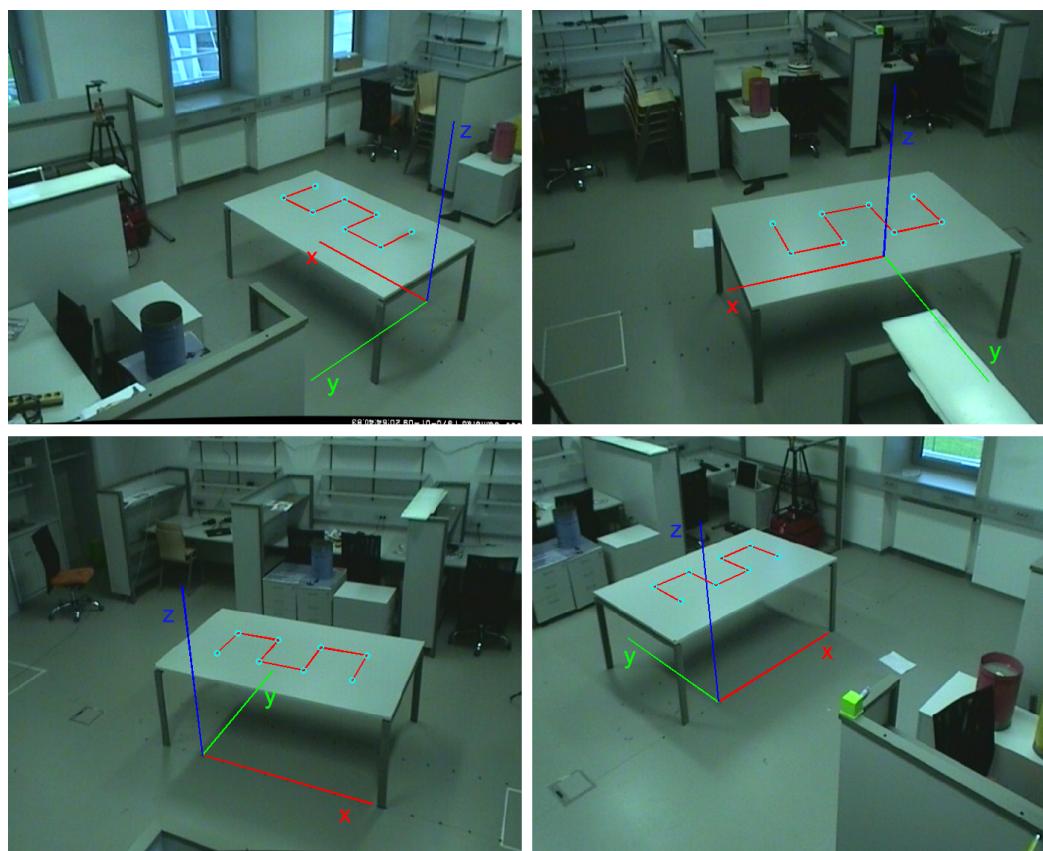
4.4 Natančnost trajektorije

Trajektorijo dobimo tako, da združimo meritve skozi čas. Slike prikazujejo zaznane trajektorije, ki so za zaznavanje lokacij uporabile različne kombinacije kamer. Z rdečo barvo je označena točna trajektorija, z zeleno pa zaznana trajektorija. Trajektorija je lomljenna, ki jo določajo točke v tabeli 4.4.

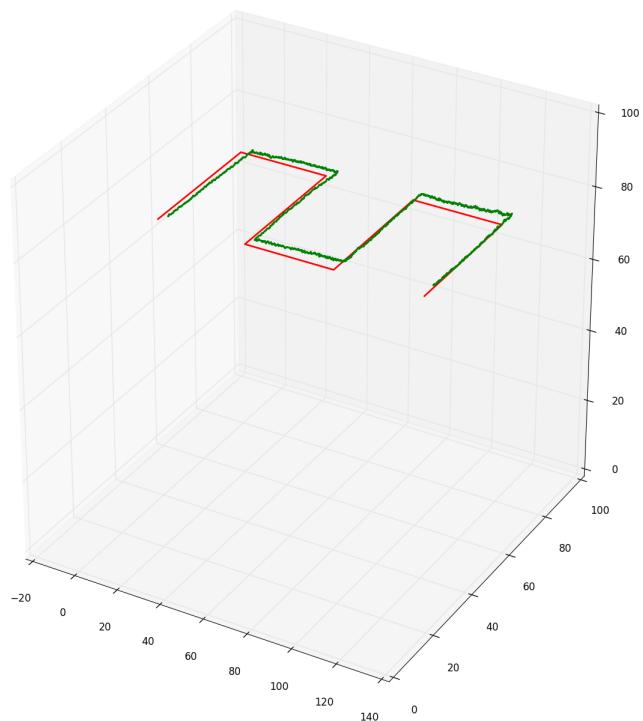
x (cm)	y (cm)	z (cm)
0	40	76
0	80	76
40	80	76
40	40	76
80	40	76
80	80	76
120	80	76
120	40	76

Tabela 4.4: Referenčne točke za meritve v prostoru.

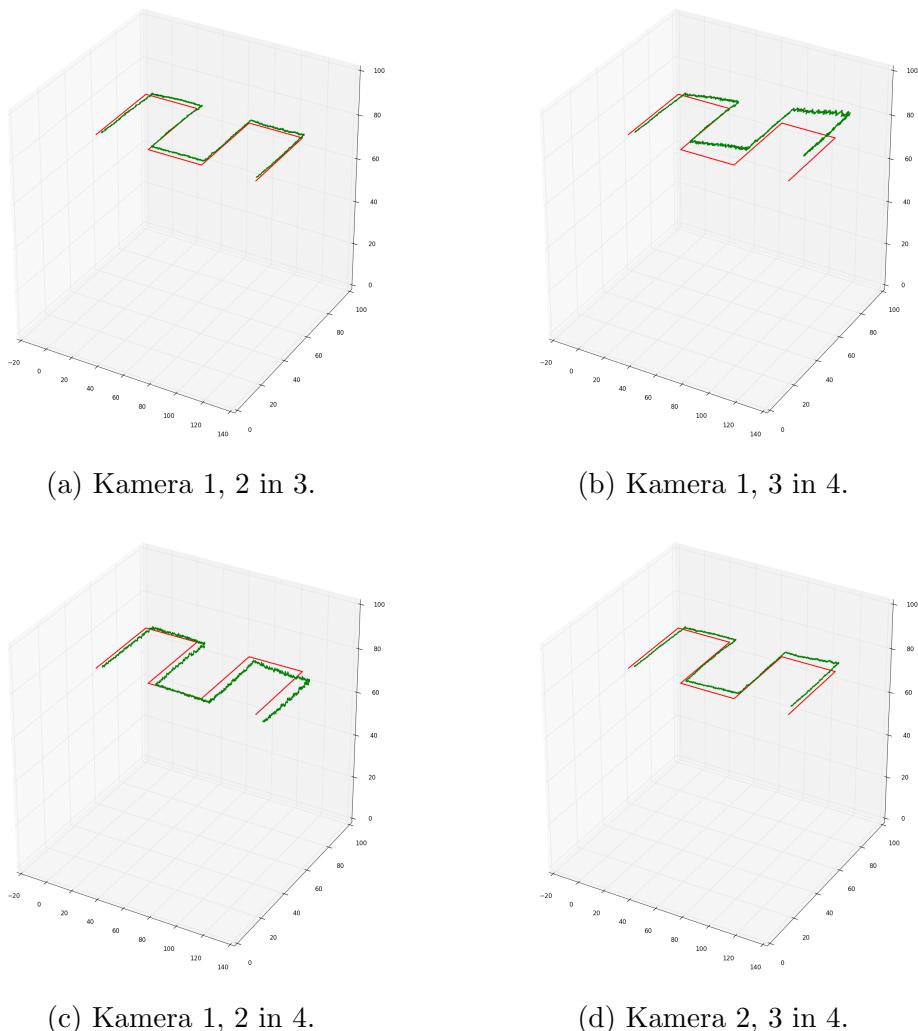
Iz slik se vidi, da je trajektorija, relativno na obliko, dobro določena, ampak narobe umeščena v prostor. Obstajata dva glavna razloga za napake v lokalizaciji: slaba kalibracija kamer ali slabo zaznavanje označevalnika. Napaka pri zaznavanju označevalnika je največ 1,5 slikovnih pik, kar je pre malo, da bi povzročilo tako drastične spremembe pri zaznavanju. Vsaka kamera je kalibrirana neodvisno od druge, zato je zelo verjetno, da je prostor za vsako kamero rahlo drugačen. To bi razložilo opazovanje obnašanje zaznavanja. Triangulacija z večjim številom kamer v povprečju vrne boljši rezultat, saj se napake med seboj skoraj izničijo. Opazimo lahko, da triangulacija s prvo in četrto kamero povzroči največjo napako.



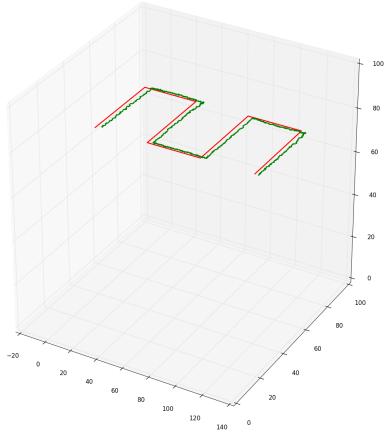
Slika 4.14: Trajektorija je lomljena označena z rdečimi daljicami.



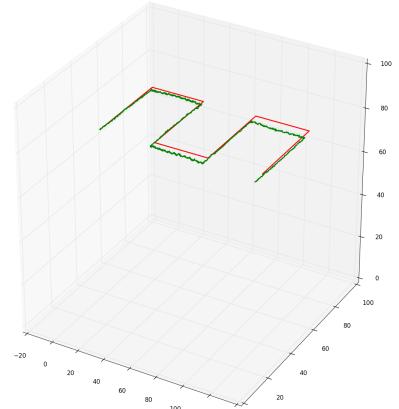
Slika 4.15: Triangulacija z vsemi kamerami.



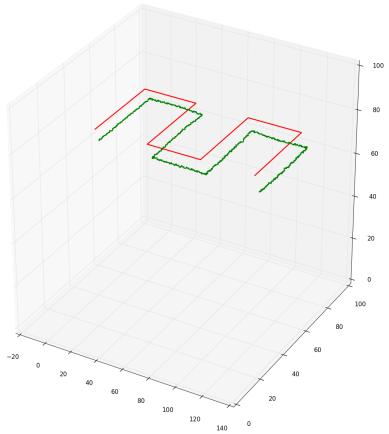
Slika 4.16: Triangulacija z vsemi trojčki kamer.



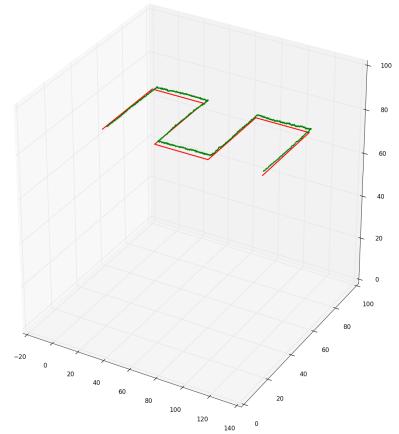
(a) Kamera 1 in 2.



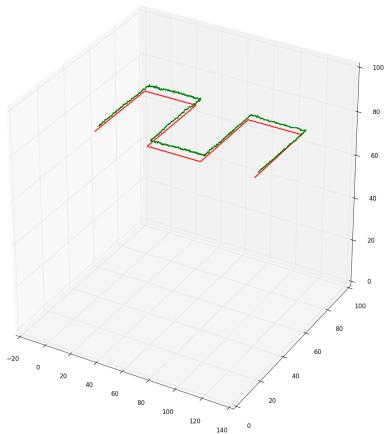
(b) Kamera 1 in 3.



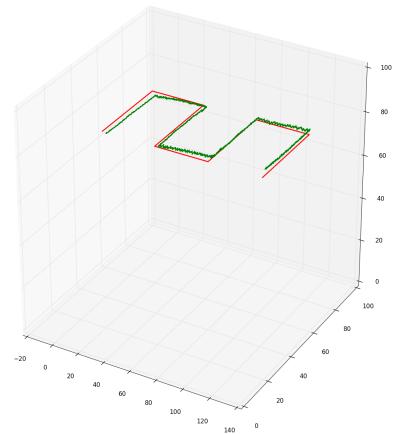
(c) Kamera 1 in 4.



(d) Kamera 2 in 3.



(e) Kamera 2 in 4.



(f) Kamera 3 in 4.

Slika 4.17: Triangulacija z vsemi pari kamer.

Poglavlje 5

Zaključek

V diplomski nalogi smo naredili večkamerni sistem za lokalizacijo objekta v prostoru. Uporabili smo štiri Axis 215 PTZ kamere in jih postavili v kote kvadratnega stropa sobe. Usmerjene so bile proti sredini tal. Z orodjem smo določili notranje parametre kamer, za izračun zunanjih parametrov pa smo uporabili metodi DLT in SVD. Zunanje parametre smo določili s ko-planarnimi točkami v prostoru, kar v sistem enačb vnese dvoumnost rešitve (sučnost koordinatnega sistema). Pravilno rešitev izberemo ročno.

Označevalnik zaznavamo s preprostim postopkom segmentacije, nato pa s slikovnimi momenti izračunamo njegovo masno središče. Sistem deluje v realnem času s 24 meritvami na sekundo. Rezultati so pokazali, da je natančnost zaznavanja označevalnika visoka (povprečno 0,85 slikovnih pik napake).

Natančnost lokalizacije je v veliki meri odvisna od natančnosti kalibracije kamer in manj od zaznavanja označevalnika. Prav tako je pomembno, katere kamere uporabimo pri triangulaciji. So pa napake pri lokalizaciji sistematične, kar pomeni, da bi jih s pravilno analizo lahko v veliki meri odpravili.

Prostora za izboljšave je še veliko. Ker kamere nimajo sočasnega prožilca za zajem slik, lahko pride do majhne nesinhroniziranosti med slikami (trianguliramo s pozicijami označevalnika v različnih časih), kar pri velikih hitrostih objekta predstavlja velik problem. Sistem bi lahko izboljšali z interpolacijo

lokacij označevalnika na slikah. Trenutno je potrebno za kalibracijo zunanjih parameterov ročno izbrati kalibracijske točke. Sistem bi lahko nadgradili tako, da kalibracija poteka avtomatizirano na podlagi vzorca v prostoru (npr. šahovnice). Nenazadnje pa je sistem zelo omejen z vrsto objekta, ki ga lokalizira (svetleč barvni označevalnik). Veliko bolj uporabno bi bilo implementirati splošni sledilnik, ki razlikuje med različnimi objekti na sliki in s tem omogoča sočasno lokalizacijo vseh objektov zanimanja.

Sistem se lahko v trenutni obliki uporablja za vizualizacijo relativne trajektorije objekta. Naredi se lahko aplikacija za navidezno risanje po prostoru. Sistematične napake so najverjetnejše posledica napak v kalibraciji, kar pomeni, da lahko z natančno kalibracijo kamer, povečamo natančnost lokalizacije. Optimalno bi bil sistem primeren za lokalizacijo mobilnega robota, ki bi se lahko avtonomno premikal po prostoru.

Literatura

- [1] V Auer, MM Maes, MJC Bonfim, MM Wanderley, and MV Lamar. 3D positioning acquisition system with application in real-time processing. 1, 2003.
- [2] Duane C Brown. Decentering distortion of lenses. *Photometric Engineering*, 32(3):444–462, 1966.
- [3] S Chan, G Sohn, L Wang, and W Lee. Dynamic Wi-Fi-based indoor positioning in 3D virtual world. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(4):1–6, 2013.
- [4] Raffaello D’Andrea. Flying machine arena. <http://raffaello.name/projects/flying-machine-arena/>, 2015. [Online; Dostopano 6. 9. 2015].
- [5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [6] Berthold KP Horn. Tsai’s camera calibration method revisited. *Online: http://people.csail.mit.edu/bkph/articles/Tsai_Revisited.pdf*, 2000.
- [7] 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.

- [8] 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.
- [9] Atri Mandal, Cristina V Lopes, Tony Givargis, Amir Haghigat, Raja Jurdak, and Pierre Baldi. Beep: 3D indoor positioning using audible sound. In *Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE*, pages 348–353. IEEE, 2005.
- [10] Mathworks. Camera calibration with matlab. <http://www.mathworks.com/videos/camera-calibration-with-matlab-81233.html>, 2015. [Online; Dostopano 14. 9. 2015].
- [11] 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.
- [12] Wikipedia. Charge-coupled device — wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Charge-coupled_device&oldid=679480626, 2015. [Online; Dostopano 6. 9. 2015].
- [13] Wikipedia. Global positioning system — wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Global_Positioning_System&oldid=679232332, 2015. [Online; Dostopano 6. 9. 2015].
- [14] Wikipedia. Multilateration — wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=Multilateration&oldid=678002521>, 2015. [Online; Dostopano 6. 9. 2015].

- [15] Wikipedia. Triangulation — wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=Triangulation&oldid=677068010>, 2015. [Online; Dostopano 6. 9. 2015].
- [16] Wikipedia. Trilateration — wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=Trilateration&oldid=677686385>, 2015. [Online; Dostopano 6. 9. 2015].
- [17] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.
- [18] Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, and Quang-Tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial intelligence*, 78(1):87–119, 1995.