

Kalibracija robotske ruke i kamere za robotsku manipulaciju zasnovanu na vizualnom servoingu

Tošeski, Marina

Master's thesis / Diplomski rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:606415>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-04-27**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Sveučilišni studij

**KALIBRACIJA ROBOTSKE RUKE I KAMERE ZA
ROBOTSKU MANIPULACIJU ZASNOVANU NA
VIZUALNOM SERVOINGU**

Diplomski rad

Marina Tošeski

Osijek, 2017

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMATIČKIH TEHNOLOGIJA OSIJEK**Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada**

Osijek, 20.09.2017.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu diplomskog rada

Ime i prezime studenta:	Marina Tošeski
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D 835 R, 09.10.2015.
OIB studenta:	43819784658
Mentor:	Prof.dr.sc. Robert Cupec
Sumentor:	Petra Đurović
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Izv. prof. dr. sc. Irena Galić
Član Povjerenstva:	Doc.dr.sc. Ratko Grbić
Naslov diplomskog rada:	Kalibracija robotske ruke i kamere za robotsku manipulaciju zasnovanu na vizualnom servoingu
Znanstvena grana rada:	Automatizacija i robotika (zn. polje elektrotehnika)
Zadatak diplomskog rada:	Izraditi računalni program za kalibraciju robotske ruke i kamere, koji na temelju niza slika snimljenih kamerom učvršćenom za robotsku ruku određuje relativni položaj koordinatnog sustava kamere i koordinatnog sustava robotske ruke. Pokusima ispitati točnost kalibracije. (Sumentor: Petra Đurović)
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	20.09.2017.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis: 
	Datum: 20.09.2017

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMATIJSKIH TEHNOLOGIJA **OSIJEK****IZJAVA O ORIGINALNOSTI RADA**

Osijek, 28.09.2017.

Ime i prezime studenta:

Marina Tošeski

Studij:

Diplomski sveučilišni studij Računarstvo

Mat. br. studenta, godina upisa:

D 835 R, 09.10.2015.

Ephorus podudaranje [%]:

1%

Ovom izjavom izjavljujem da je rad pod nazivom: **Kalibracija robotske ruke i kamere za robotsku manipulaciju zasnovanu na vizualnom servoingu**

izrađen pod vodstvom mentora Prof.dr.sc. Robert Cupec

i sumentora Petra Đurović

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Tošeski Marina

Sadržaj

1. UVOD	3
2. PRIMJENA RAČUNALNOG VIDA	4
3. KALIBRACIJA DOBIVENA RJEŠAVANJEM PROBLEMA APSOLUTNE ORIJENTACIJE	7
4. IZVEDBA KALIBRACIJE I REZULTATI	11
4.1. Korištene tehnologije	11
4.2. Implementacija	12
4.3. Testiranje dobivenih transformacija	17
4.4. Vizualni servoing	29
5. ZAKLJUČAK	34

1. UVOD

Cilj diplomskog rada je riješiti problem kalibracije robotskog manipulatora i kamere za robotsku manipulaciju zasnovanu na vizualnom servoingu. U okviru ovog rada, primjenjuje se rješenje zatvorenog oblika, koje je predložio Berthold K.P. Horn [1]. Predstavljena metoda uključuje pronalazak relativnog položaja dva koordinatna sustava mjerenjem koordinata objekta od interesa u odnosu na oba koordinatna sustava za zadani broj koordinata. U ovom radu objekt od interesa predstavlja alat robotskog manipulatora, čiji je položaj potrebno mjeriti u koordinatnom sustavu robotskog manipulatora i koordinatnom sustavu kamere. Rezultat primjene ove metode za kalibraciju robotskog manipulatora i kamere je homogena transformacijska matrica između koordinatnog sustava korištenog robotskog manipulatora i kamere.

Zadatak rada je izrada računalnog programa za kalibraciju robotske ruke i kamere, koji na temelju niza snimki dobivenih kamerom određuje odnos koordinatnog sustava kamere i koordinatnog sustava robotske ruke te ispitati točnost pokusima.

2. PRIMJENA RAČUNALNOG VIDA ZA UPRAVLJANJE ROBOTSКИM MANIPULATOROM

Robotski vid je sustav sačinjen od percepcijskog senzora, koji prikuplja informacije, odnosno snima okolinu robota, te *softwarea* za obradu dobivene snimke u cilju izdvajanja informacije potrebne za upravljanje robotom [1]. U praksi se koriste 2D percepcijski senzori, kao što je obična kamera te 3D percepcijski senzori, kao što su laserski davač udaljenosti i RGB-D kamera. U ovom se radu pod pojmom slike podrazumjeva slika snimljena običnom kamerom, kao i slika dobivena RGB-D kamerom.

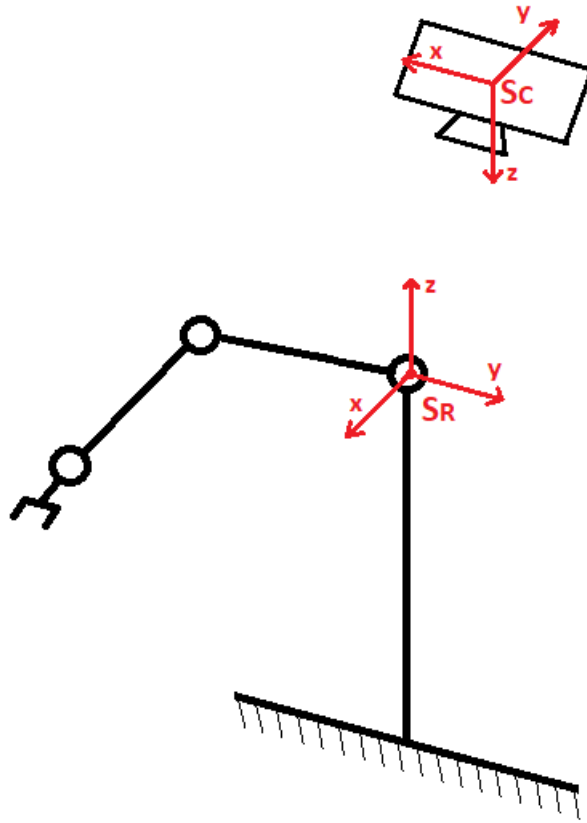
Sustav robotskog vida analizira slike i daje opis onoga što je prikazano na slici, pri čemu ti opisi moraju sadržavati informacije o objektima na slici, odnosno značajke, kako bi robot mogao izvršiti neki zadatak koji uključuje te objekte. Iz tog razloga robotski vid se može promatrati kao sučelje između stvarnog svijeta i robotskog sustava [1].

Neki od ciljeva sustava robotskog vida su prepoznavanje objekta radi robotske manipulacije istima, određivanje položaja robota u prostoru i u odnosu na objekte u prostoru, detekcija prepreka te prepoznavanje ljudi i pokreta. Da bi robot mogao na odgovarajući način pozicionirati alat u odnosu na objekt s kojim manipulira, on mora imati informaciju o položaju tog objekta u odnosu na alat [1].

Vizualni servoing, odnosno robotsko upravljanje temeljeno na viziji, tehnika je koja koristi informaciju dobivenu sensorima za upravljanje pokretima robota. Prednosti takvog upravljanja su vidljive pri radu s objektima čija je pozicija nepoznata u odnosu na robot ili pri radu manipulatora s manjom točnošću pozicioniranja. Robotski vid i robotska manipulacija u vidu vizualnog servoinga su kombinirani na principu 'gledanja' i 'kretanja', pri čemu točnost pozicioniranja ovisi o točnosti vidnog senzora i upravljačkog sustava manipulatora. Upravljanje na temelju vizualnog servoinga uključuje više područja: obradu slike, kinematiku, dinamiku, teoriju upravljanja i računarstvo u stvarnom vremenu. Pri vizualnom servoingu kamera može biti montirana na alat robotskog manipulatora, pri čemu je njihov međusobni položaj konstantan, ili fiksiran, pri čemu je potrebno riješiti problem međusobnog položaja kamere i alata robotskog manipulatora [2]. U ovom radu se razmatra slučaj u kojem je kamera fiksno postavljena iznad robotskog manipulatora.

Kalibracija robotskog manipulatora i kamere, koji se obrađuje u radu, vrši se radi koordinacije pokreta robota s obzirom na informaciju dobivenu kamerom. Iz tog razloga bitno je utvrditi odnos između koordinatnog sustava robotskog manipulatora i kamere. Da bi se provela

kalibracija robotskog manipulatora i kamere, potrebno je definirati kalibracijski objekt. U ovom radu je to marker postavljen na alat robotskog manipulatora. Shematski prikaz robotskog manipulatora i kamere prikazan je na slici 2.1.



Slika 2.1: Shematski prikaz robotskog manipulatora i kamere s pripadajućim koordinatnim sustavima

Kao rezultat kalibracije potrebno je dobiti homogenu transformacijsku matricu ${}^R T_C$, koja predstavlja položaj koordinatnog sustava S_C u koordinatnom sustavu S_R . Homogenu transformacijsku matricu potrebno je formirati na sljedeći način:

$$T_{RC} = \begin{bmatrix} R_{RC} & t_{RC} \\ 0 & 1 \end{bmatrix}, \quad (1-1)$$

gdje je R_{RC} rotacijska matrica iz koordinatnog sustava S_C u koordinatni sustav S_R , a t_{RC} , translacijski vektor iz koordinatnog sustava S_C u koordinatni sustav S_R .

U radu se razmatra metoda kalibracije koja, za izračun homogene transformacijske matrice, koristi nasumično zadane pozicije alata robotskog manipulatora u koordinatnom sustavu S_R te

njemu pripadajuće pozicije markera, koji je postavljen na alat robotskog manipulatora, u koordinatnom sustavu S_C . Problem se može formulirati na sljedeći način:

Za dva skupa točaka

$$\{R_i\}, \{C_i\},$$

gdje R_i predstavlja poziciju u koordinatnom sustavu S_R , a C_i poziciju u koordinatnom sustavu S_C , potrebno je odrediti rotaciju i translaciju te pomoću njih formirati homogenu transformacijsku matricu T_{RC} tako da suma kvadrata razlike između pozicija točaka R_i i točaka C_i transformiranih pomoću T_{RC} bude minimalna.

3. KALIBRACIJA DOBIVENA RJEŠAVANJEM PROBLEMA APSOLUTNE ORIJENTACIJE

Rješavanje problema apsolutne orijentacije vrši se radi kalibracije robotskog manipulatora i kamere za manipulaciju zasnovanu na vizualnom servoingu. Kalibracija se provodi tako da se robotskom manipulatoru zadaje određeni broj nasumičnih pozicija u zadanom radnom prostoru na koje mora pozicionirati alat. Zadani radni prostor robotskog manipulatora je sljedeći:

$$x \in [150 \text{ mm}, 250 \text{ mm}] \quad (3-1)$$

$$y \in [-150 \text{ mm}, 150 \text{ mm}] \quad (3-2)$$

$$z \in [100 \text{ mm}, 200 \text{ mm}] \quad (3-3)$$

Kamerom je potrebno snimiti položaj alata na tim pozicijama. Na taj način se dobiju parovi koordinata alata robota u koordinatnom sustavu robotskog manipulatora i u koordinatnom sustavu kamere. U nastavku je matrica, koja sadrži sve koordinate alata zadane robotu, označena sa pR , dok je matrica, koja sadrži koordinate alata koje je usnimila kamera, označena s pC [3]. Stupci tih matrica označavaju poziciju na koju je alat manipulatora doveden, dok redci predstavljaju x , y i z koordinatu te pozicije.

Navedene koordinate korisno je predstaviti u odnosu na središnju vrijednost točaka svakog koordinatnog sustava, odnosno pomoću centroida, koji su definirani na sljedeći način:

$$\overline{pR} = \frac{1}{n} \sum_{i=1}^n pR_i, \quad \overline{pC} = \frac{1}{n} \sum_{i=1}^n pC_i, \quad (3-4)$$

gdje je $\overline{pR}$ centroid koordinata u odnosu na koordinatni sustav robotskog manipulatora, $\overline{pC}$ centroid koordinata u odnosu na koordinatni sustav kamere, dok pR_i i pC_i označavaju svaku poziciju zadanu manipulatoru, odnosno poziciju snimljenu kamerom. Broj zadanih pozicija je označen sa n . Dobivene se koordinate obilježavaju kako je navedeno:

$$pR'_i = pR_i - \overline{pR} \ , \ pC'_i = pC_i - \overline{pC}, \quad (3-5)$$

gdje je pR'_i matrica koja sadrži koordinate robota predstavljene u odnosu na centroid robota, a pC'_i matrica koja sadrži koordinate u koordinatnom sustavu kamere u odnosu na centroid kamere [3].

Transformacija između ovih koordinatnih sustava zahtjeva izračun skalarnog faktora rotacijske matrice, koji je u nastavku označen sa s . Skalarni faktor daje odnos dimenzija objekta na slici i fizičkih dimenzija objekta, odnosno u ovom slučaju udaljenosti od ishodišta koordinatnog sustava robota. Izračun skalarnog faktora je obavljen na sljedeći način:

$$s = \sqrt{\frac{\sum_{i=1}^n \|pC'_i\|^2}{\sum_{i=1}^n \|pR'_i\|^2}} \quad (3-6)$$

Gdje su $\|pC'_i\|^2$ i $\|pR'_i\|^2$ kvadrati duljina vektora formiranih od pojedinih stupaca matrica. Za dobivanje skale na način predstavljen u članku, nije potrebno poznavati rotaciju između koordinatnih sustava [3].

Rotacija između koordinatnih sustava predstavljena je pomoću jediničnih kvaterniona. Prednost ovakve notacije je u tome što su jedinični kvaternioni usko povezani s notacijom osi i kutova. Kvaternioni mogu biti prikazani kao vektori s četiri komponente, od kojih je prva vrijednost realna komponenta kvaterniona, a ostale tri su imaginarne komponente. Kvaternion $\dot{q}$ se može zapisati na sljedeći način:

$$\dot{q} = q_0 + iq_x + jq_y + kq_z, \quad (3-7)$$

gdje je q_0 realni dio, a q_x , q_y i q_z su tri imaginarna dijela kvaterniona [3].

Kako bi se odredio kvaternion, pomoću kojeg je predstavljena rotacija, potrebno je izračunati matricu N dimenzija 4×4 . Matrica N ima sljedeći oblik:

$$N = \begin{bmatrix} S_{xx} + S_{yy} + S_{zz} & S_{yz} - S_{zy} & S_{zx} - S_{xz} & S_{xy} - S_{yx} \\ S_{yz} - S_{zy} & S_{xx} - S_{yy} - S_{zz} & S_{xy} + S_{yx} & S_{zx} + S_{xz} \\ S_{zx} - S_{xz} & S_{xy} + S_{yx} & -S_{xx} + S_{yy} - S_{zz} & S_{yz} + S_{zy} \\ S_{xy} - S_{yx} & S_{zx} + S_{xz} & S_{yz} + S_{zy} & -S_{xx} - S_{yy} + S_{zz} \end{bmatrix}, \quad (3-8)$$

gdje je

$$S_{xx} = \sum_{i=0}^n x'_{pRi} x'_{pCi}, \quad S_{xy} = \sum_{i=0}^n x'_{pRi} y'_{pCi}, \quad (3-9)$$

pri čemu x'_{pRi} predstavlja x komponentu matrice pR_i . Ostale veličine: S_{yy} , S_{xz} , S_{yz} , S_{zz} računaju se analogno za odgovarajuće koordinate x , y i z . Kvaternion, kojim je predstavljena rotacija, može se izračunati kao svojstveni vektor koji odgovara najvećoj svojstvenoj vrijednosti matrice N . Taj kvaternion daje najveću vrijednost sljedećeg izraza:

$$\dot{q}^T N \dot{q}, \quad (3-10)$$

gdje je $\dot{q}$ jedinični kvaternion.

Rotacijsku matricu je moguće formirati s realnim i imaginarnim komponentama dobivenog kvaterniona, na način:

$$R_{RC} = \begin{bmatrix} 1 - 2q_j^2 - 2q_k^2 & 2(q_i q_j - q_k q_r) & 2(q_i q_k + q_j q_r) \\ 2(q_i q_j + q_k q_r) & 1 - 2q_i^2 - 2q_k^2 & 2(q_j q_k - q_i q_r) \\ 2(q_i q_k - q_j q_r) & 2(q_j q_k + q_i q_r) & 1 - 2q_i^2 - 2q_j^2 \end{bmatrix} \quad (3-11)$$

gdje je R_{RC} rotacijska matrica iz koordinatnog sustava robota u koordinatni sustav kamere [3].

Translacija između dvaju koordinatnih sustava se može izračunati kao razlika između centroida koordinatnog sustava kamere te skaliranog i rotiranog centroida koordinatnog sustava robotskog manipulatora, odnosno:

$$t_{RC} = \overline{pC} - sR_{RC}(\overline{pR}), \quad (3-12)$$

gdje je t_{RC} translacijski vektor između koordinatnih sustava [3].

Rotacijska matrica i translacijski vektor elementi su koji formiraju matricu homogene transformacije iz koordinatnog sustava robotskog manipulatora u koordinatni sustav kamere T_{RC} , koja je oblika:

$$T_{RC} = \begin{bmatrix} R_{RC} & t_{RC} \\ 0 & 1 \end{bmatrix} \quad (3-13)$$

Za primjenu i ispitivanje rezultata kalibracije, potrebno je izračunati transformaciju koordinata iz koordinatnog sustava kamere u koordinatni sustav robotskog manipulatora, odnosno T_{CR} . Matrica T_{CR} se dobije inverznom funkcijom matrice T_{RC} , odnosno:

$$T_{CR} = T_{RC}^{-1} \quad (3-14)$$

Ovaj postupak se u radu izvršava za sljedeće zadane brojeve točaka : 10, 20, 30, 40, 50. Različit broj točaka se uzima u obzir kako bi se testiranjem utvrdilo na koji način točnost kalibracije ovisi o broju točaka s kojima je kalibracija izvršena.

4. IZVEDBA KALIBRACIJE I REZULTATI

4.1. Korištene tehnologije

Korištena je RGB-D *Orbbec Astra S* kamera. Kamera je kompatibilna s *OpenNI* aplikacijama te omogućuje 3D viziju. *Orbbec Astra S* kamera je namijenjena za korištenje pri kratkom dometu, npr. 3D skeniranje, raspoznavanje lica, itd. Domet kamere je između 0.4 m i 2 m [4]. Robotski manipulator korišten pri radu je *Dobot Arm V1.0*. Pri tome su kamera i robot montirani na postolje na način prikazan na slici 4.1.1. [5]



Slika 4.1.1: Dobot Arm V1.0

Rješenje problema apsolutne orijentacije je implementirano u *MS Visual Studio 2013* razvojnom okruženju. Za izradu računalne aplikacije koja vrši kalibraciju, korišten je C++

¹ P. Đurović, R. Grbić, R. Cupec, D. Filko, Low Cost Robot Arm with Visual Guided Positioning, 40th Jubilee International Convention on Information and Communication Technology, Electronics and Microelectronics, Opatija, Hrvatska, 22.-26. 05. 2017.

programski jezik, dok je Python skriptni jezik korišten za kontrolu robotskog manipulatora. Pri implementaciji algoritma uključena je programska biblioteka *OpenCV*(*Open Source Computer Vision Library*)[6], koja omogućuje infrastrukturu za aplikacije računalnog vida te ubrzava korištenje strojne percepcije. *OpenCV* biblioteka se koristi za olakšani rad sa slikama dobivenim kamerom u matričnom zapisu te olakšava proces izračuna naveden u prethodnom poglavlju [7]. Uključuje se i *ArUco* biblioteka za aplikacije proširene stvarnosti, koja se temelji na *OpenCV* biblioteci. *ArUco* biblioteka korištena je za detekciju markera postavljenog na alat, čime se omogućava raspoznavanje alata robotskog manipulatora na slici te određuje njegova pozicija u odnosu na kameru [8].

4.2.Implementacija

Kako je navedeno u prethodnom poglavlju, robotu se zadaje broj nasumično generiranih točaka u definiranom radnom prostoru. Definicija broja točaka i krajnjih koordinata radnog prostora je prikazan na primjeru 4.2.1.

```
float x, y, z;
int pointN = 50; // Best calibration results for N = 50;
int x_min = 150;
int x_max = 250;
int y_min = -150;
int y_max = 150;
int z_min = 100;
int z_max = 200;
```

Primjer 4.2.1: Definicija broja točaka i krajnjih koordinata radnog prostora

Za pohranu matrica navedenih u prethodnom poglavlju, koriste se objekti klase *cv::Mat* definirane u *OpenCV* knjižnici (Pr.4.2.2.) [9].

```
cv::Mat pointsRobot(3, pointN, CV_32F);
cv::Mat pointsCamera(3, pointN, CV_32F);
cv::Mat robotCentroid(3, 1, CV_32F);
cv::Mat cameraCentroid(3, 1, CV_32F);
//coordinates according to centroids
cv::Mat cpointsRobot(3, pointN, CV_32F);
cv::Mat cpointsCamera(3, pointN, CV_32F);
//Finding scale
float sumRobot = 0;
float sumCamera = 0;
cv::Mat N(4, 4, CV_32F);
cv::Mat lambda(4, 1, CV_32F);
cv::Mat eigenVec(4, 4, CV_32F);
```

Primjer 4.2.2: Definicija objekta klase *cv::Mat*, matrice se koriste u daljem provođenju algoritma

Za početak kalibracije, potrebno je detektirati marker postavljen na vrh alata robotskog manipulatora. Pri tome kamera uzima sliku sve dok marker nije detektiran (Pr.4.2.3.). Za snimanje slike RGB-D kamerom i detekciju markera koristi se funkcija *GetImage*, koja predstavlja sastavni dio programskog modula *Robot Manipulation Module (RMM)* razvijenog na FERIT Osijek u okviru projekta Napredna 3D percepcija za mobilne robotske manipulatore [10].

```
//Searching for a marker
do GetImage();
while (Markers.size() == 0);
```

Primjer 4.2.3: Pronalazak markera na slici

Primjer 4.2.4. prikazuje generiranje nasumičnih *x*, *y* i *z* koordinata te pohranjivanje istih u matricu *pointsRobot*. Na navedeni način dobivaju se koordinate u zadanom intervalu prikazanom na primjeru 4.2.1. Svaki puta kad se generira koordinata, alat robotskog manipulatora se pozicionira na tu koordinatu kako bi kamera mogla zabilježiti poziciju manipulatora pomoću funkcije *void RMM::GetPoint(bool bObject)*. Funkcija *GetPoint* uzima snimku pomoću kamere te provjerava nalazi li se marker postavljen na alat manipulatora na slici. Ako je marker detektiran, njegova se pozicija u koordinatnom sustavu kamere pohranjuje u matricu *points*. Koordinata dobivena pomoću kamere se tada sprema u matricu *pointsCamera* (Pr.4.2.5.). Za pomicanje robotske ruke na zadanu poziciju koristi se funkcija *MoveToCoordinates* opisana u [10].

```
x = rand() % (x_max - x_min) + x_min;
pointsRobot.at<float>(0, i) = x;
y = rand() % (y_max - y_min) + y_min;
pointsRobot.at<float>(1, i) = y;
z = rand() % (z_max - z_min) + z_min;
pointsRobot.at<float>(2, i) = z;

MoveToCoordinates(x, y, z);
```

Primjer 4.2.4: Generiranje nasumičnih koordinata i pozicioniranje robotskog manipulatora pomoću funkcije *void RMM::MoveToCoordinates(int x, int y, int z)*

```
GetPoint();
pointsCamera.at<float>(0, i) = point.at<float>(0, 0) * 1000;
pointsCamera.at<float>(1, i) = point.at<float>(1, 0) * 1000;
pointsCamera.at<float>(2, i) = point.at<float>(2, 0) * 1000;
```

Primjer 4.2.5. Pohranjivanje koordinata dobivenih kamerom pomoću funkcije *void RMM::GetPoint(bool bObject)*

Centroid je srednja vrijednost svih generiranih koordinata zadanih robotskom manipulatoru, odnosno koordinata dobivenih kamerom. Za dobivanje centroida koordinata robotskog manipulatora, zbrajaju se posebno *x*, *y* i *z* koordinate svih generiranih točaka, kako je prikazano na

primjeru 4.2.6. Isto vrijedi i za centroid koordinata dobivenih kamerom. Centroid koordinata se tada dobiva dijeljenjem sume koordinata sa zadanim brojem točaka *pointsN* (Pr.4.2.6.).

```
for (i = 0; i < pointN; i++)
{
    x = rand() % (x_max - x_min) + x_min;
    pointsRobot.at<float>(0, i) = x;
    y = rand() % (y_max - y_min) + y_min;
    pointsRobot.at<float>(1, i) = y;
    z = rand() % (z_max - z_min) + z_min;
    pointsRobot.at<float>(2, i) = z;

    MoveToCoordinates(x, y, z);

    GetPoint();
    pointsCamera.at<float>(0, i) = point.at<float>(0, 0) * 1000;
    pointsCamera.at<float>(1, i) = point.at<float>(1, 0) * 1000;
    pointsCamera.at<float>(2, i) = point.at<float>(2, 0) * 1000;

    //suming columns in mat (centroid)
    //robot
    robotCentroid.at<float>(0, 0) += pointsRobot.at<float>(0, i);
    robotCentroid.at<float>(1, 0) += pointsRobot.at<float>(1, i);
    robotCentroid.at<float>(2, 0) += pointsRobot.at<float>(2, i);
    //camera
    cameraCentroid.at<float>(0, 0) += pointsCamera.at<float>(0, i);
    cameraCentroid.at<float>(1, 0) += pointsCamera.at<float>(1, i);
    cameraCentroid.at<float>(2, 0) += pointsCamera.at<float>(2, i);
}

//get centroid
robotCentroid /= pointN;
cameraCentroid /= pointN;
```

Primjer 4.2.6: Petlja u kojoj se vrši pozicioniranje alata robotskog manipulatora, snimanje položaja kamerom te zbrajanje x, y i z koordinata, kako bi se kako bi se dobio centroid robotskog manipulatora i centroid kamere.

Primjer 4.2.7. prikazuje predstavljanje svih koordinata u odnosu na centroide, kao što je navedeno u formuli (3-4). Svaki stupac predstavlja jednu koordinatu kako kod robotskog manipulatora, tako i kod kamere. Stupcima zasebno se pristupa pomoću metode *cv::col(int x)*; [9].

```
// refer all measurements to the centroids
for (i = 0; i < pointN; i++)
{
    cpointsRobot.col(i) = pointsRobot.col(i) - robotCentroid;
    cpointsCamera.col(i) = pointsCamera.col(i) - cameraCentroid;
}
```

Primjer 4.2.7: Koordinate zadane robotskom manipulatoru i koordinate snimljene kamerom predstavljaju se u odnosu na pripadajući centroid

Za izračun skalarnog faktora *scale*, potrebno je implementirati formulu (3-6), što je prikazano na primjeru 4.2.8. Metoda *double norm(InputArray src1, int normType=NORM_L2)* kao rezultat daje kvadrat duljine proslijeđenog vektora [11].

```
// find scale
for (i = 0; i < pointN; i++)
{
    //normtype = L2
    sumRobot += cv::norm(cpointsRobot.col(i), 2) * cv::norm(cpointsRobot.col(i), 2);
    sumCamera += cv::norm(cpointsCamera.col(i), 2) * cv::norm(cpointsCamera.col(i),
2);
}
scale = std::sqrt(sumCamera / sumRobot);
```

Primjer 4.2.8: Izračun skalarnog faktora

Formula (3-9) je implementirana na način prikazan na primjeru 4.2.9., koji prikazuje izračun elemenata koji se koriste za formiranje matrice *N* (Pr.4.2.10.). Kao što je navedeno u prethodnom poglavlju, potrebno je pronaći svojstveni vektor koji odgovara najvećoj svojstvenoj vrijednosti matrice *N*. Taj svojstveni vektor je tada kvaternion, čiji se elementi koriste za formiranje rotacijske matrice. Elegantno rješenje za taj problem je korištenje *bool eigen(InputArray src, OutputArray eigenvalues, OutputArray eigenvectors)* metode, koja vraća svojstvene vrijednosti *lambda* u silaznom poretku i njemu pripadajuće svojstvene vektore (Pr.4.2.11). S obzirom na silazni poredak, svojstveni vektor koji odgovara najvećoj svojstvenoj vrijednosti je prvi redak matrice *eigenVec*, dok su elementi kvaterniona raspoređeni po stupcima (Pr.4.2.12) [11].

```
for (i = 0; i < pointN; i++)
{
    Sxx += cpointsRobot.at<float>(0, i) * cpointsCamera.at<float>(0, i);
    Sxy += cpointsRobot.at<float>(0, i) * cpointsCamera.at<float>(1, i);
    Sxz += cpointsRobot.at<float>(0, i) * cpointsCamera.at<float>(2, i);
    Syx += cpointsRobot.at<float>(1, i) * cpointsCamera.at<float>(0, i);
    Syy += cpointsRobot.at<float>(1, i) * cpointsCamera.at<float>(1, i);
    Syz += cpointsRobot.at<float>(1, i) * cpointsCamera.at<float>(2, i);
    Szx += cpointsRobot.at<float>(2, i) * cpointsCamera.at<float>(0, i);
    Szy += cpointsRobot.at<float>(2, i) * cpointsCamera.at<float>(1, i);
    Szz += cpointsRobot.at<float>(2, i) * cpointsCamera.at<float>(2, i);
}
```

Primjer 4.2.9: Dobivanje S_{ij} ($i,j = \{x,y,z\}$)

```
N.at<float>(0, 0) = Sxx + Syy + Szz;
N.at<float>(0, 1) = Syz - Szy;
N.at<float>(0, 2) = Szx - Sxz;
N.at<float>(0, 3) = Sxy - Syx;
N.at<float>(1, 0) = Syz - Szy;
N.at<float>(1, 1) = Sxx - Syy - Szz;
N.at<float>(1, 2) = Sxy + Syx;
N.at<float>(1, 3) = Szx + Sxz;
```

```

N.at<float>(2, 0) = Sxz - Sxz;
N.at<float>(2, 1) = Sxy + Syx;
N.at<float>(2, 2) = -Sxx + Syy - Szz;
N.at<float>(2, 3) = Syz + Szy;
N.at<float>(3, 0) = Sxy - Syx;
N.at<float>(3, 1) = Sxz + Sxz;
N.at<float>(3, 2) = Syz + Szy;
N.at<float>(3, 3) = -Sxx - Syy + Szz;

```

Primjer 4.2.10: Dobivanje matrice N

```

//eigenvalues and eigenvector
cv::eigen(N, lambda, eigenVec);

```

Primjer 4.2.11: Dobivanje svojstvenih vektora i svojstvenih vrijednosti matrice N

```

//get rotation
float qr = eigenVec.at<float>(0, 0); //quaternion coef
float qi = eigenVec.at<float>(0, 1);
float qj = eigenVec.at<float>(0, 2);
float qk = eigenVec.at<float>(0, 3);

```

Primjer 4.2.12: Dobivanje elemenata jediničnog kvaterniona, pomoću kojeg se računaju elementri rotacijske matrice

Rotacijska matrica dobivena je na način naveden u formuli (3-11), što je prikazano na primjeru 4.2.13., dok je izračun translacijskog pomaka t_{RC} (3-12) prikazan na primjeru 4.2.14. Homogena transformacijska matrica se tada formira kako je navedeno na primjeru 4.2.15. Objasnjenim postupkom dobiva se homogena transformacijska matrica iz koordinatnog sustava robota u koordinatni sustav kamere T_{CR} , čime se dobiva veza između dva koordinatna sustava. Međutim, za potrebe testiranja, potrebno je znati obrnuto, odnosno homogenu transformacijsku matricu iz koordinatnog sustava kamere u koordinatni sustav robota T_{CR} , odnosno inverznu matricu T_{RC} (Pr.4.2.16).

```

RRC.at<float>(0, 0) = 1 - 2 * qj*qj - 2 * qk*qk;
RRC.at<float>(0, 1) = 2 * (qi*qj - qk*qr);
RRC.at<float>(0, 2) = 2 * (qi*qk + qj*qr);
RRC.at<float>(1, 0) = 2 * (qi*qj + qk*qr);
RRC.at<float>(1, 1) = 1 - 2 * qi*qi - 2 * qk*qk;
RRC.at<float>(1, 2) = 2 * (qj*qk - qi*qr);
RRC.at<float>(2, 0) = 2 * (qi*qk - qj*qr);
RRC.at<float>(2, 1) = 2 * (qj*qk + qi*qr);
RRC.at<float>(2, 2) = 1 - 2 * qi*qi - 2 * qj*qj;

```

Primjer 4.2.13: Formiranje rotacijske matrice RRC pomoću elemenata jediničnog kvaterniona

```

tRC = cameraCentroid - scale * RRC * robotCentroid;

```

Primjer 4.2.14: Izračun translacijskog pomaka t_{RC}

```

//transformation matrix
TRC.at<float>(0, 0) = RRC.at<float>(0, 0);
TRC.at<float>(0, 1) = RRC.at<float>(0, 1);
TRC.at<float>(0, 2) = RRC.at<float>(0, 2);
TRC.at<float>(0, 3) = tRC.at<float>(0);
TRC.at<float>(1, 0) = RRC.at<float>(1, 0);
TRC.at<float>(1, 1) = RRC.at<float>(1, 1);
TRC.at<float>(1, 2) = RRC.at<float>(1, 2);
TRC.at<float>(1, 3) = tRC.at<float>(1);
TRC.at<float>(2, 0) = RRC.at<float>(2, 0);
TRC.at<float>(2, 1) = RRC.at<float>(2, 1);
TRC.at<float>(2, 2) = RRC.at<float>(2, 2);
TRC.at<float>(2, 3) = tRC.at<float>(2);
TRC.at<float>(3, 0) = 0;
TRC.at<float>(3, 1) = 0;
TRC.at<float>(3, 2) = 0;
TRC.at<float>(3, 3) = 1;

```

Primjer 4.2.15: Formiranje transformacijske matrice od elemenata rotacijske matrice RRC i translacijskog pomaka tRC

```

//Testing
RRC_ = RRC.t();
TCR = TRC.inv();
MoveToCoordinates(170, 0, 90);

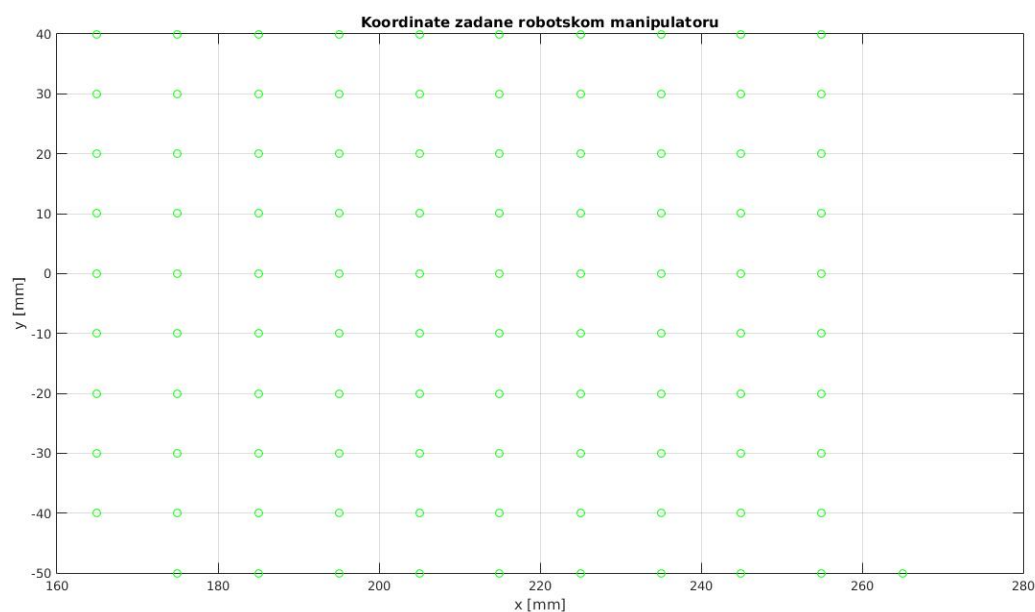
```

Primjer 4.2.16: Priprema za testiranje

4.3. Testiranje dobivenih transformacija

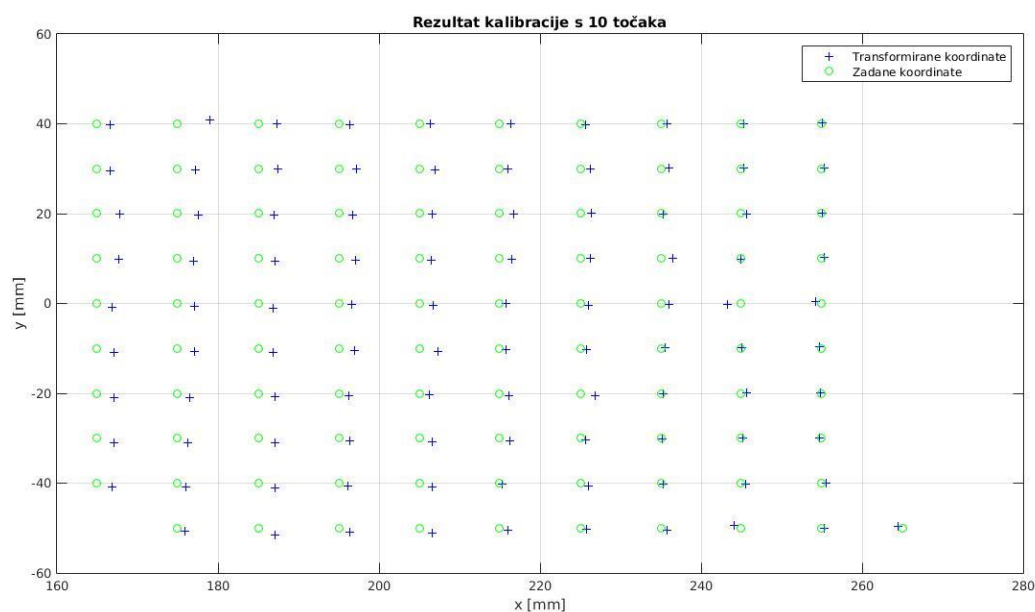
Testiranje dobivene transformacije između koordinatnih sustava izvršeno je pozicioniranjem alata robotskog manipulatora na sto zadanih koordinata unutar radnog prostora, uzimanjem snimke na toj poziciji, te transformacijom koordinata, na kojima se alat nalazi, dobivenih u sustavu kamere u sustav robotskog manipulatora. Zadana z koordinata je 150, dok su zadane x i y koordinate prikazane na slici 4.3.1. Testiranje je rađeno uz pretpostavku da će veći broj pozicija korištenih u postupku kalibracije dati manju razliku između koordinata zadanih

robotskom manipulatoru i transformiranih koordinata dobivenih kamerom. Sva mjerenja navedena u daljnjem tekstu iskazana su u milimetrima.

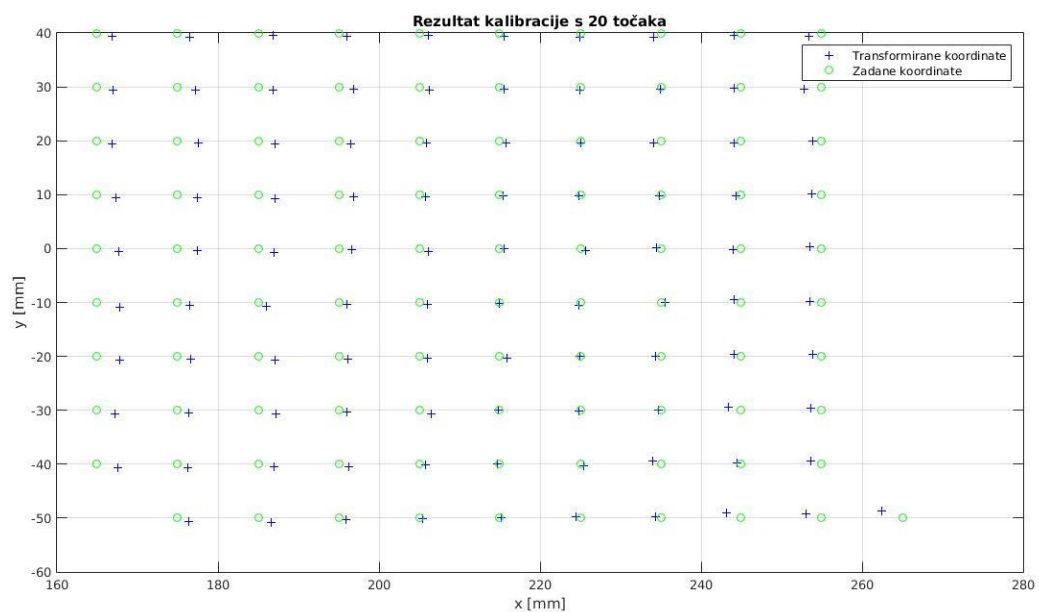


Slika 4.3.1: X i y koordinate zadane robotskom manipulatoru. Z koordinata je nepromjenjiva i iznosi 150mm

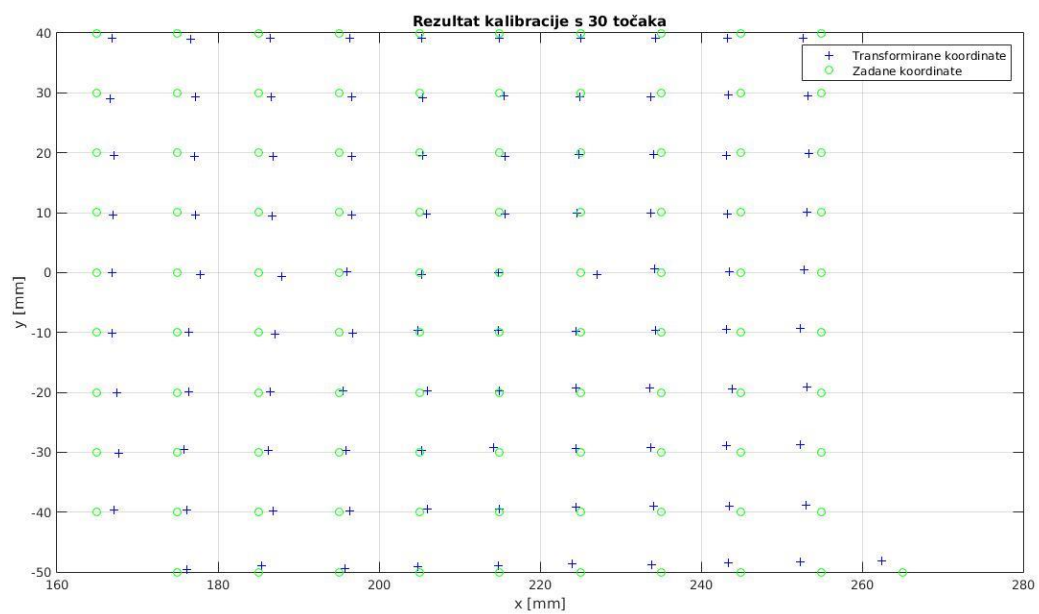
U nastavku slijede prikazi koordinata zadanih robotu i transformiranih koordinata dobivenih kamerom za 10 (Sl. 4.3.2), 20 (Sl. 4.3.3), 30 (Sl. 4.3.4), 40 (Sl. 4.3.5) i 50 (Sl. 4.3.6) zadanih točaka.



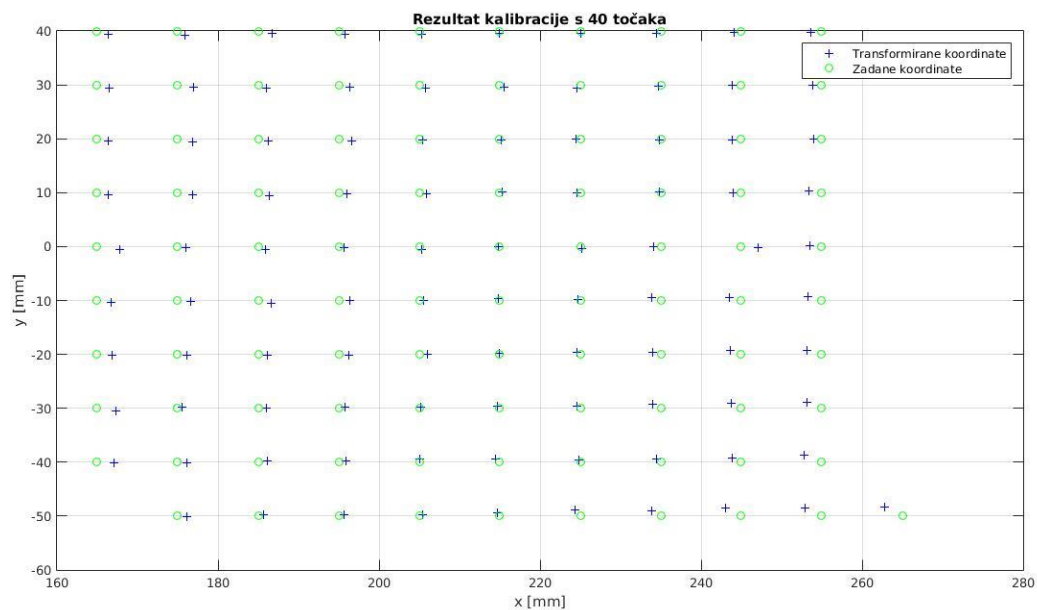
Slika 4.3.2: Prikaz koordinata zadanih robotu i koordinata dobivenih transformacijom kao rezultat kalibracije s 10 točaka



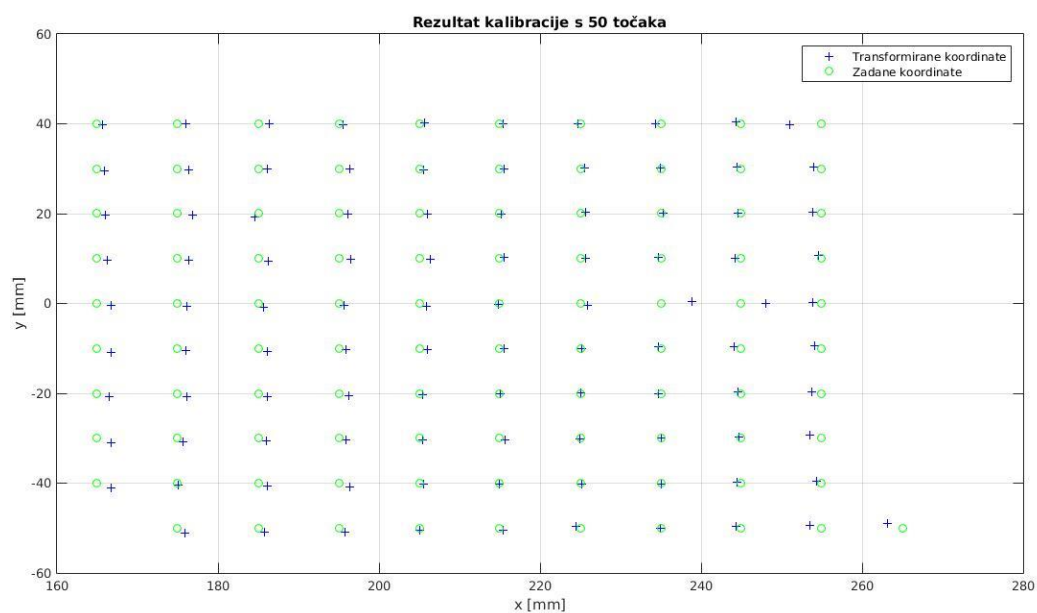
Slika 4.3.3: Prikaz koordinata zadanih robotu i koordinata dobivenih transformacijom kao rezultat kalibracije s 20 točaka



Slika 4.3.4: Prikaz koordinata zadanih robotu i koordinata dobivenih transformacijom kao rezultat kalibracije s 30 točaka



Slika 4.3.5: Prikaz koordinata zadanih robotu i koordinata dobivenih transformacijom kao rezultat kalibracije s 40 točaka



Slika 4.3.6: Prikaz koordinata zadanih robotu i koordinata dobivenih transformacijom kao rezultat kalibracije s 50 točaka

Za svaki par koordinata, odnosno za svaku zadanu koordinatu i njoj pripadajuću transformiranu koordinatu, tablica 4.3.1 prikazuje sljedeće vrijednosti:

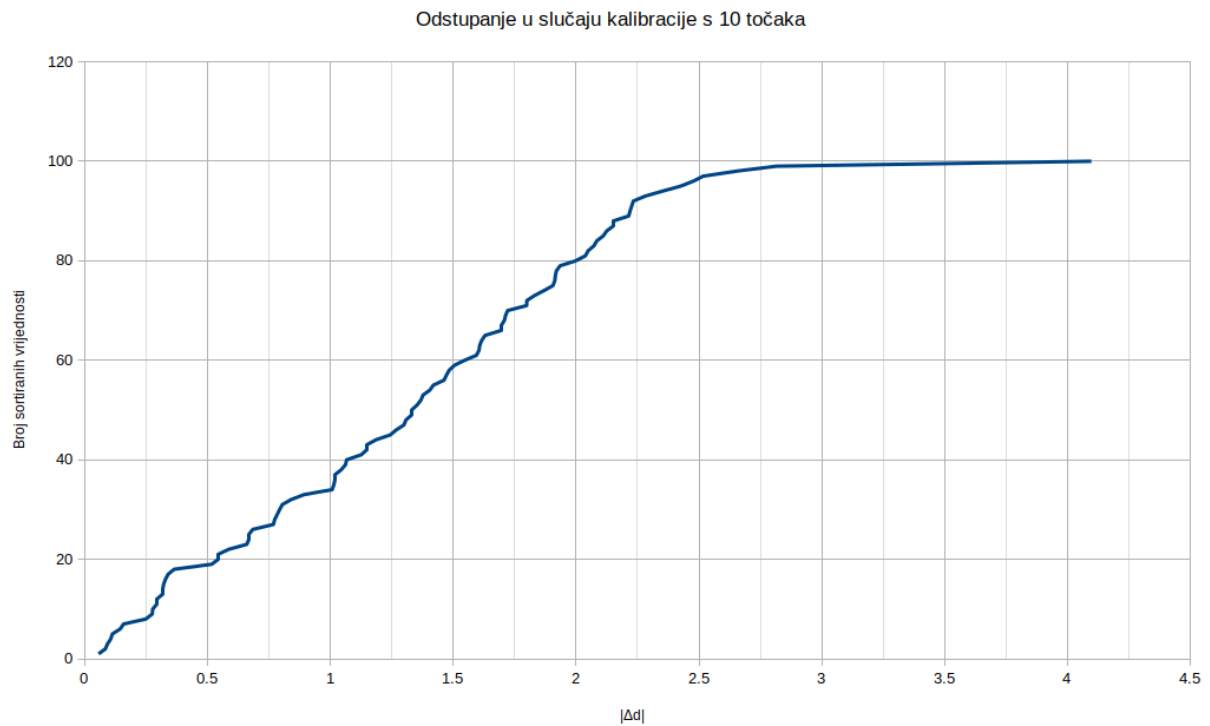
- srednja vrijednost razlike parova koordinata po x – osi ($\Delta x'$)
- srednja vrijednost razlike parova koordinata po y – osi ($\Delta y'$)

- srednja vrijednost apsolutne razlike parova koordinata po x – osi ($|\Delta x|'$)
- srednja vrijednost apsolutne razlike parova koordinata po y – osi ($|\Delta y|'$)
- srednja vrijednost razlike udaljenosti između koordinata zadanih robotu i koordinata snimljenih kamerom transformiranih u koordinatni sustav robota ($\Delta d'$)
- srednja vrijednost apsolutne razlike udaljenosti između koordinata zadanih robotu i koordinata snimljenih kamerom transformiranih u koordinatni sustav robota ($|\Delta d|'$)
- najmanja vrijednost apsolutne razlike parova koordinata po x – osi ($\min |\Delta x|$)
- najveća vrijednost apsolutne razlike parova koordinata po x – osi ($\max |\Delta x|$)
- najmanja vrijednost apsolutne razlike parova koordinata po y – osi ($\min |\Delta y|$)
- najveća vrijednost apsolutne razlike parova koordinata po y – osi ($\max |\Delta y|$)
- najmanja vrijednost apsolutne razlike udaljenosti između koordinata zadanih robotu i koordinata snimljenih kamerom transformiranih u koordinatni sustav robota ($\min |\Delta d|$)
- najveća vrijednost apsolutne razlike udaljenosti između koordinata zadanih robotu i koordinata snimljenih kamerom transformiranih u koordinatni sustav robota ($\max |\Delta d|$)

Broj točaka	10	20	30	40	50
$\Delta x'$ [mm]	1.1876792	0.5093069	0.1643888	0.1853727899	0.39825453
$\Delta y'$ [mm]	-0.317725	-0.279556	0.06028113	0.03892894	-0.15177944
$ \Delta x' $ [mm]	1.2858097	1.2098395	1.28203259	1.01514051	0.87211167
$ \Delta y' $ [mm]	0.4080006	0.44107042	0.56175609	0.4049647	0.35376584
$\Delta d'$ [mm]	1.2119490985	0.4870686	0.0830288	0.134309	0.41438032
$ \Delta d' $ [mm]	1.31417172	1.2179777	1.285894	1.02322056	0.8946868
min $ \Delta x $ [mm]	0.037735	0.029953	0.0021209	0.0092009	0.012344
max $ \Delta x $ [mm]	4.010071	2.794083	2.888336	2.8281859	3.98979
min $ \Delta y $ [mm]	0.007895	0.013898	0.009313	0.031861	0.0063609
max $ \Delta y $ [mm]	1.612289	1.30432	1.859501	1.600521	1.150444
min $ \Delta d $ [mm]	0.056370	0.037656	0.02923724	0.0227603847	0.0174127680
max $ \Delta d $ [mm]	4.097275	2.844098	2.94093523	2.828858	3.96980036

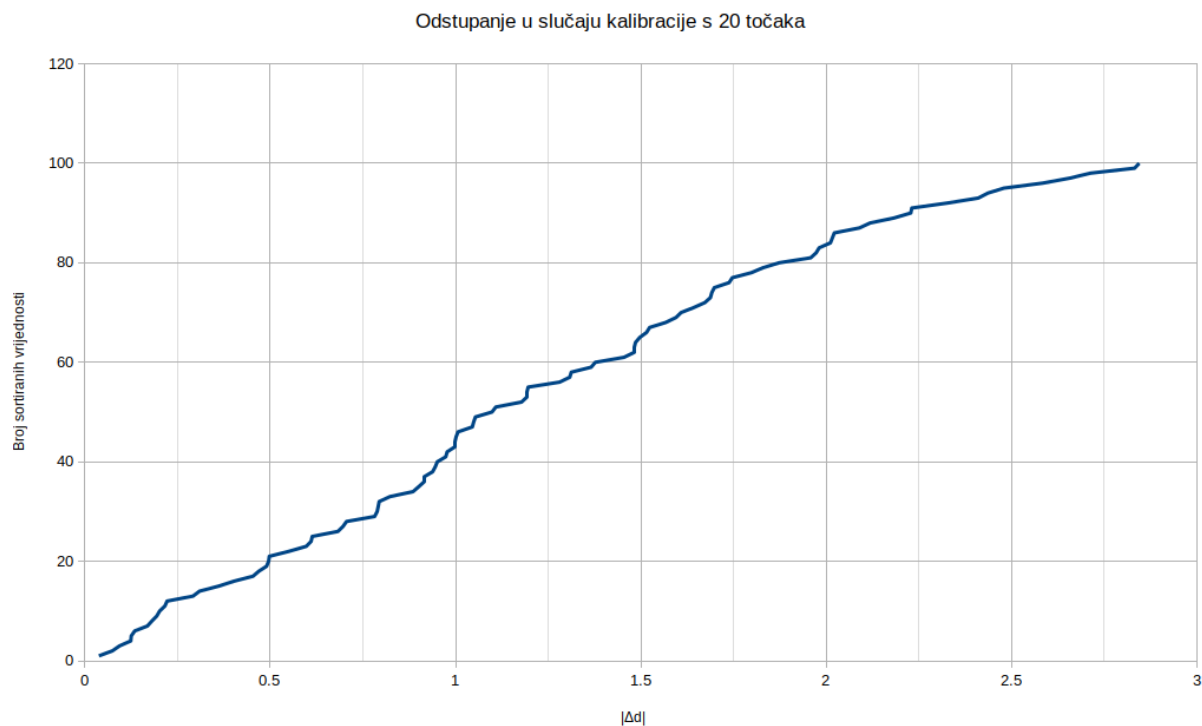
Tablica 4.3.1: Rezultat statističke analize dobivenih rezultata

Razlika je veća po x-osi nego po y-osi, što utječe i na razliku udaljenosti od središta koordinatnog sustava robota. Nadalje, bitno je uočiti najveću vrijednost apsolutne razlike udaljenosti parova koordinata od ishodišta koordinatnog sustava robota te usporediti te vrijednosti sa pripadajućom srednjom vrijednosti razlike udaljenosti. Srednja vrijednost razlike udaljenosti je najmanja kod testiranja kalibracije s 50 točaka, dok najveća vrijednost apsolutne razlike udaljenosti nije najmanja u tom slučaju. Iz toga se zaključuje da je transformacija dobivena kalibracijom s 50 točaka najtočnija, ali da ima određeni broj ekstremnih vrijednosti veći nego što je to slučaj kod kalibracije s 20, 30 i 40 točaka. Navedeno je uočljivo prikazom kumulativnih funkcija distribucije, odnosno prikazom sortiranih apsolutnih razlika udaljenosti parova koordinata od ishodišta koordinatnog sustava robota. Na sljedećim slikama nalazi se prikaz kumulativnih funkcija distribucije za slučaj svake kalibracije.



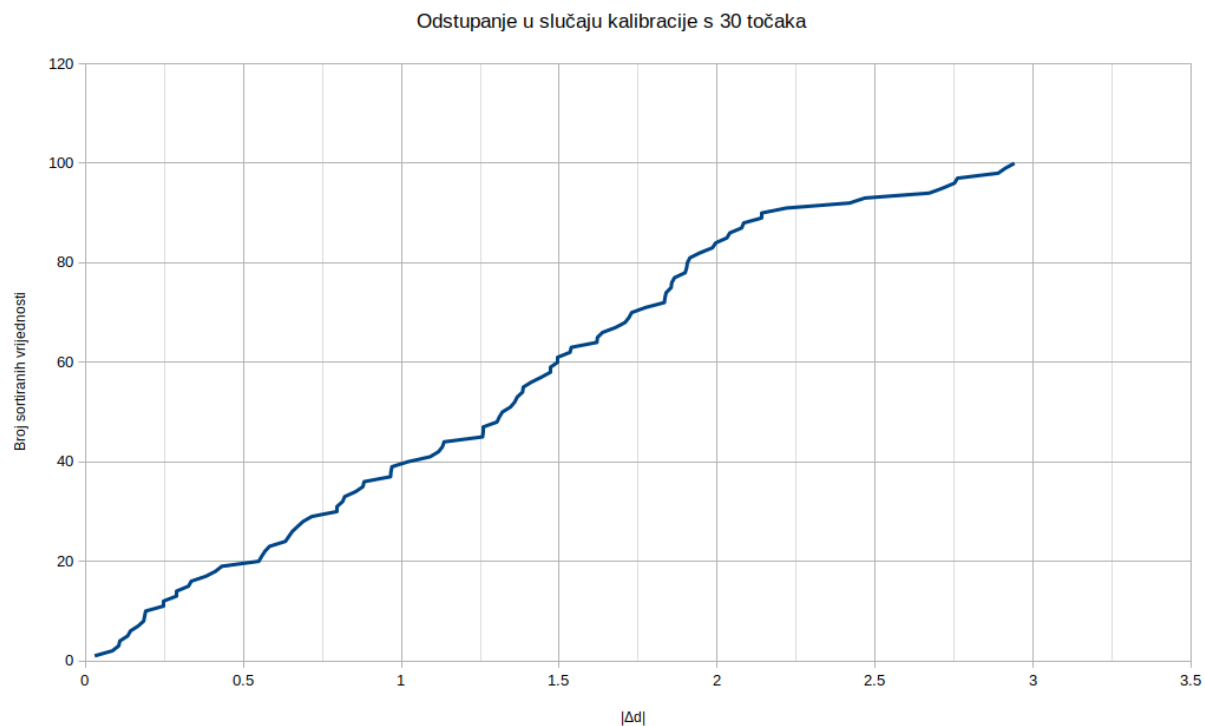
Slika 4.3.7: Prikaz sortiranih apsolutnih razlika udaljenosti parova koordinata od ishodišta u slučaju kalibracije s 10 koordinata

Na slici 4.3.7 prikazana je razlika u mjerenju u slučaju kalibracije s 10 koordinata. Pritom je 97% vrijednosti unutar 2.5 milimetra razlike te je raspoređenost istih gotovo linearna. Razlika u ostalih 3% mjerenja naglo raste.



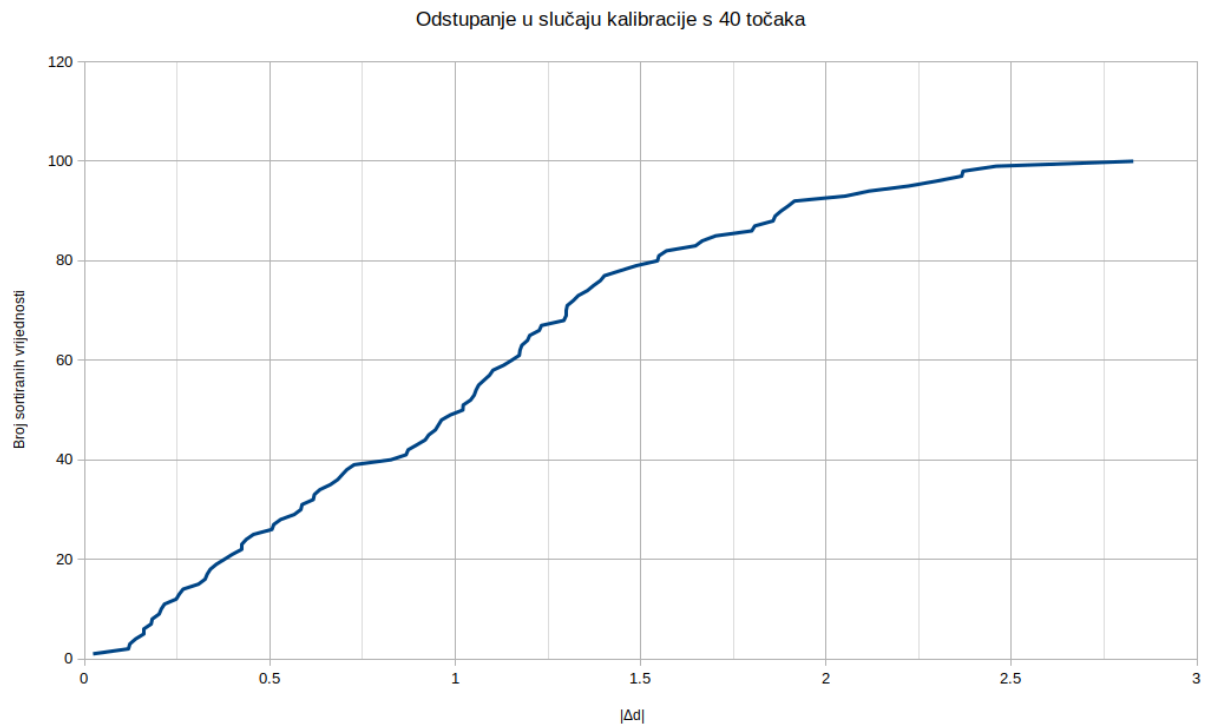
Slika 4.3.8: Prikaz sortiranih apsolutnih razlika udaljenosti parova koordinata od ishodišta u slučaju kalibracije s 20 koordinata

Rast razlike u mjerenju u slučaju kalibracije s 20 koordinata, prikazan na slici 4.3.8., gotovo je linearan, a razlika je za sva mjerenja manje od 3 milimetra. Osim manjeg ekstrema mjerenja, graf prikazuje manju razliku udaljenosti parova koordinata od ishodišta, nego što je to slučaj kod kalibracije s 10 točaka.



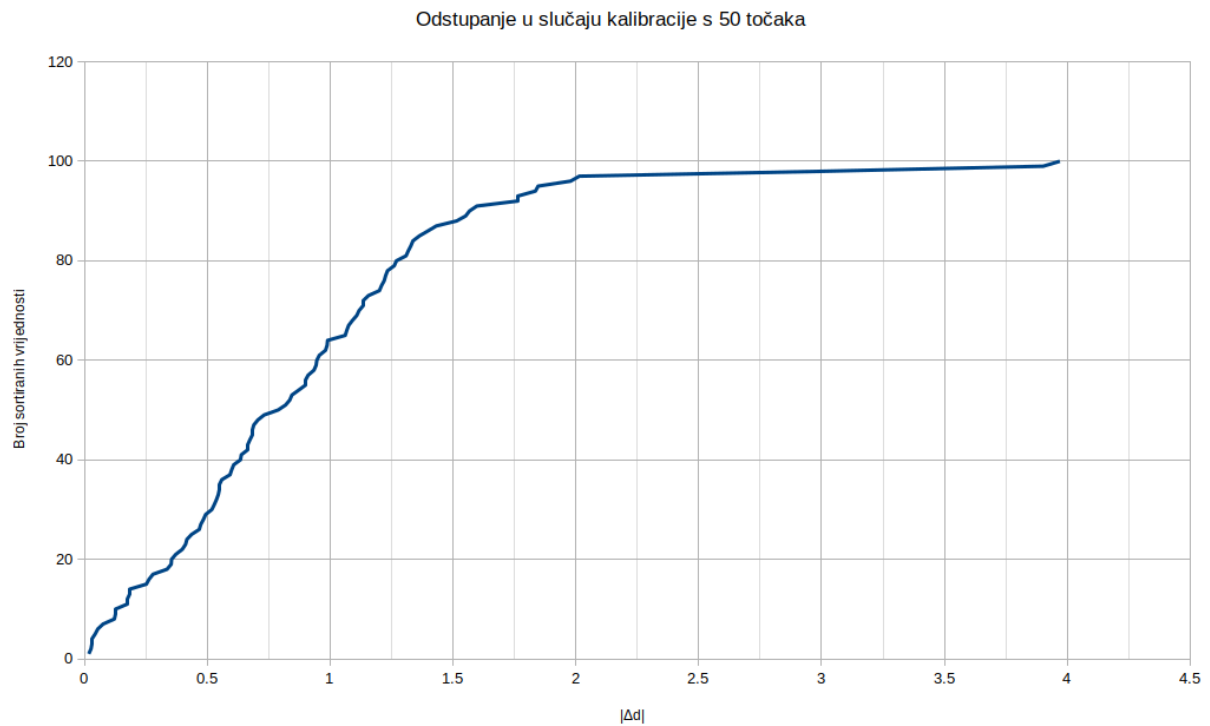
Slika 4.3.9: Prikaz sortiranih apsolutnih razlika udaljenosti parova koordinata od ishodišta u slučaju kalibracije s 30 koordinata

Odstupanje u mjerenju u slučaju kalibracije s 30 koordinata, prikazan na slici 4.3.9., ne odudara značajno od predhodna dva rezultata, kao što je prikazano i u tablici 4.3.1.



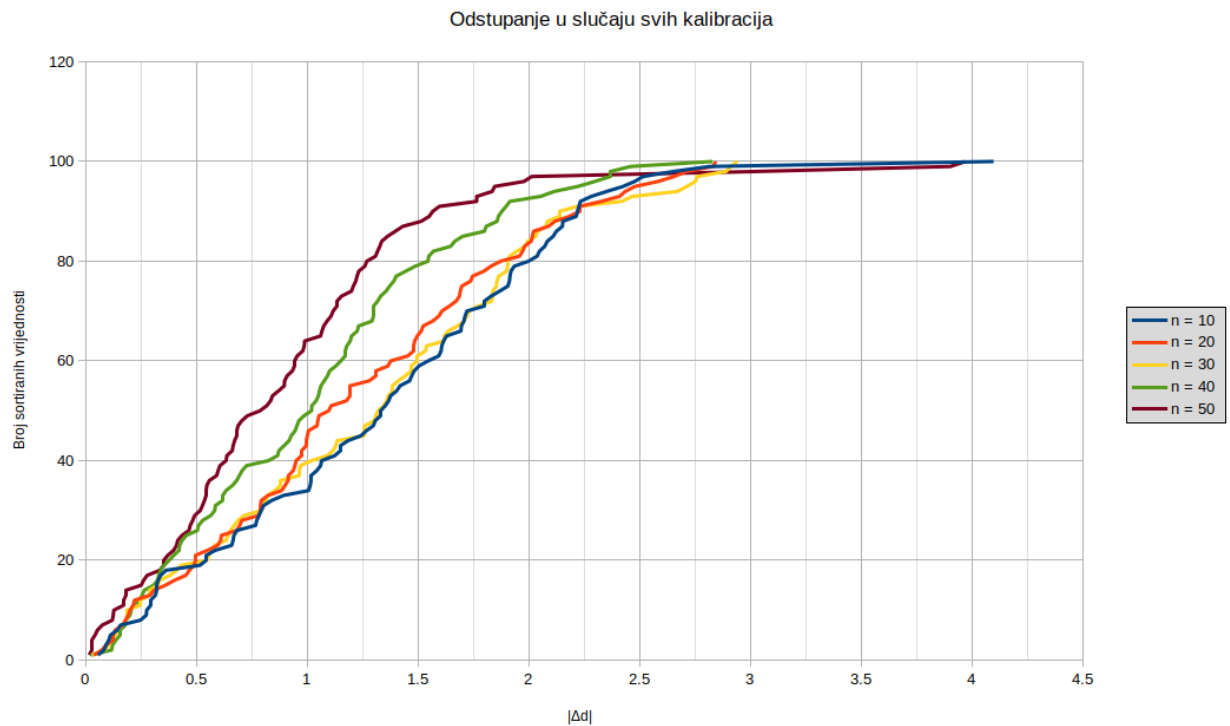
Slika 4.3.10: Prikaz sortiranih apsolutnih razlika udaljenosti parova koordinata od ishodišta u slučaju kalibracije s 40 koordinata

Za slučaj kalibracije s 40 nasumičnih točaka, čiji je rezultat prikazan na slici 4.3.10., razlika udaljenosti se ponovno smanjuje. Nadalje, 99% mjerenja je unutar razlike od 2.5 milimetra, za razliku od kalibracije s 10 nasumičnih točaka, gdje je to bio 97% mjerenja.



Slika 4.3.11: Prikaz sortiranih apsolutnih razlika udaljenosti parova koordinata od ishodišta u slučaju kalibracije s 50 koordinata

Kalibracija s 50 nasumično zadanih točaka, prikazana na slici 4.3.11., daje najmanje razlike udaljenosti para koordinata od ishodišta koordinatnog sustava. Međutim, za razliku od kalibracije s 20, 30 i 40 točaka, ekstremne vrijednosti su iznad 3 milimetra. No, bez obzira na ekstremne vrijednost, 97% mjerenja daje razliku udaljenosti para koordinata od ishodišta unutar 2 milimetra.



Slika 4.3.12: Prikaz sortiranih apsolutnih razlika udaljenosti parova koordinata od ishodišta koordinatnog sustava robota za sve kalibracije

Slika 4.3.12. prikazuje rezultate svih kalibracija. Na ovom prikazu uočljivo je kako broj zadanih nasumičnih vrijednosti utječe na rezultat kalibracije, te da kalibracija s pedeset zadanih točaka u većini slučajeva ima manju razliku udaljenosti parova koordinata od ishodišta.

4.4. Vizualni servoing

Vizualni servoing se u ovom testiranju provodi kako bi se procijenila točnost dobivene transformacijske matrice u slučaju kada robotu nisu zadane koordinate, već se pozicioniranje vrši na temelju prepoznatog objekta. U ovom slučaju taj objekt je marker s pripadajućim identifikacijskim brojem prepoznat od strane kamere. Markeri su ručno postavljeni na nasumične pozicije na radnoj površini. Postupak vizualnog servoinga se izvršava na način da se robotski manipulator pokušava pozicionirati na sredinu svakog od markera i to četiri puta. Kako bi se robotski manipulator uspio pozicionirati, marker mora biti u vidnom polju robota i ništa ga ne smije zaklanjati. Pozicioniranje alata se izvodi iterativno, na temelju snimke dobivene kamerom robotski manipulator dobiva informaciju o poziciji markera te se pokušava pozicionirati na tu poziciju. Tada se ponovo uzima slika i uspoređuje se nalazi li se alat na poziciji gdje se prvotno nalazio postavljeni marker. Ako nije na toj poziciji, robot korigira poziciju. Postupak korekcije pozicije alata robotskog manipulatora se ponavlja sve dok se alat ne nalazi unutar zadanog praga od 1 mm.

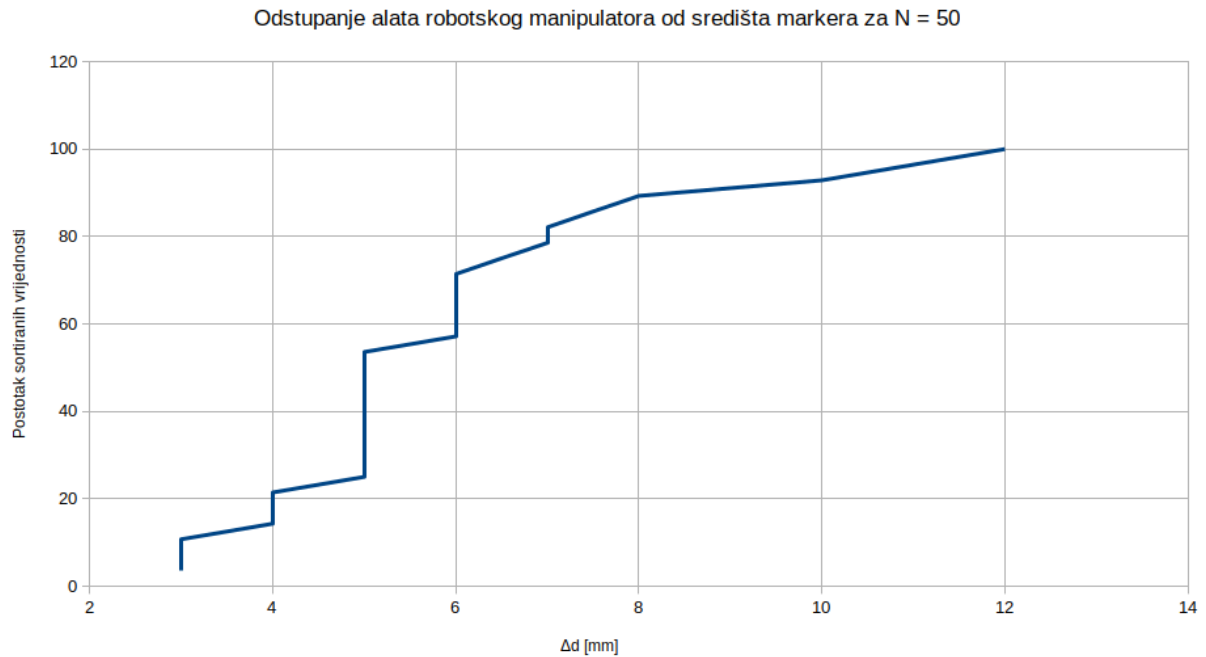
Testiranje pozicioniranja je izvršeno za kalibraciju s 50 nasumično zadanih pozicija jer je takva kalibracija dala najbolje rezultate testiranja u prethodnom poglavlju. tablica 4.4.1. prikazuje izmjerene udaljenosti od središta markera za svaki marker, kao i standardnu devijaciju mjerenja te najveću, srednju i najmanju razliku.

Rezultati vizualnog servoinga, prikazani u tablici 4.4.1., daju prilično široki spektar odstupanja položaja robotskog manipulatora od središta markera. Naime, srednje odstupanje mjerenja je 5,89 mm, te odstupanje poprima vrijednosti i od 12 mm. Udaljenost alata robotskog manipulatora od središta markera je također prikazana na slici 4.4.1.

Točnost pozicioniranja se može poboljšati uključivanjem informacije o dubini snimke te skaliranjem svake pozicije na temelju te informacije tokom postupka vizualnog servoinga. Pritom se u svakoj iteraciji pozicioniranja uzima snimka te se svaka pozicija dobivena kamerom množi s odgovarajućim skalarnim faktorom [10] (Pr.4.4.1).

Marker ID	d[mm]	σ_d
151	5	2,220932
	5	max_d
	7,5	12
	11	min_d
152	3	3
	5	avg_d
	8	5,892857
	12	
153	3	
	4	
	6	
	10	
155	5	
	5	
	5	
	7	
156	3	
	4	
	4	
	5	
157	5	
	6	
	6	
	6	
158	4	
	6	
	6,5	
	7	

Tablica 4.4.1: Izmjerene udaljenosti od središta markera, standardna devijacija te najveća udaljenost, najmanja udaljenost i srednja vrijednost udaljenosti od središta markera



Slika 4.4.1: Odstupanje alata robotskog manipulatora od središta markera za kalibraciju s 50 točaka

```
//Searching depth based on camera
for (int i = -2; i <= 2; i++)
{
    for (int j = -2; j <= 2; j++)
    {
        if ((float)depthF.at<uint16_t>(imgPxY + j, imgPxX + i) != 0)
        {
            zDepth += depthF.at<uint16_t>(imgPxY+j, imgPxX+i);
            br++;
        }
    }
}

float zAruco = 1000 * Markers.at(ID).Tvec.at<float>(2, 0);

if (br == 0) zDepth = zAruco;
else zDepth = zDepth / br;
int16_t zDepth = depthF.at<uint16_t>(imgPxY, imgPxX);

//Point
float scaleDepth = (float)zDepth / zAruco;
float scaleDepth = 1;
point.at<float>(0, 0) = Markers.at(ID).Tvec.at<float>(0, 0)*scaleDepth;
point.at<float>(1, 0) = Markers.at(ID).Tvec.at<float>(1, 0)*scaleDepth;
point.at<float>(2, 0) = Markers.at(ID).Tvec.at<float>(2, 0)*scaleDepth;
```

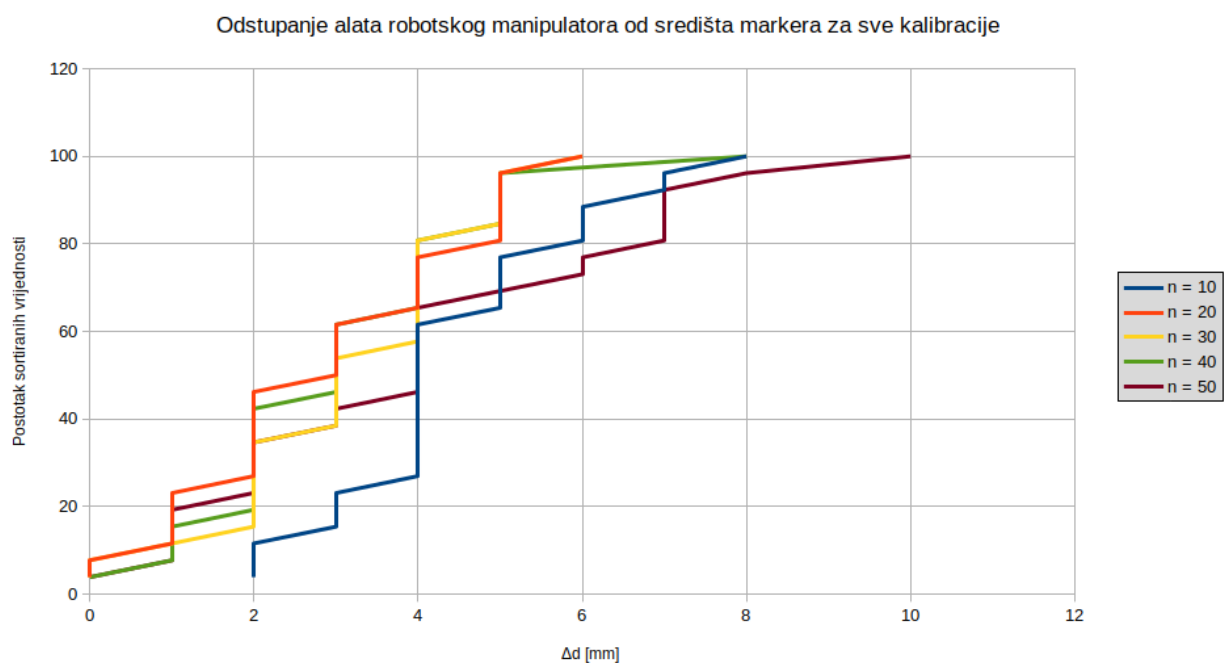
Primjer 4.4.1: Dobivanje informacije o dubini snimke te skaliranje pozicije na temelju te informacije[10]

Broj zadanih točaka pri kalibraciji	10	20	30	40	50
	d[mm]				
1	7	3	1	2	7
2	7	5	4	2	4
3	4	3	3	4	7
4	5	1	5	5	6
5	4	2	3	4	8
6	8	5	2	1	5
7	4	2	5	8	6
8	4	0	2	5	7
9	6	1	3	2	3
10	5	5	2	3	4
11	5	4	3	4	1
12	6	0	0	2	4
13	5	2	4	1	4
14	3	5	2	5	2
15	4	1	3	3	2
16	4	3	4	3	2
17	4	2	0	0	1
18	6	2	5	1	4
19	2	6	6	2	1
20	4	5	2	3	0
21	2	3	4	2	1
22	4	4	4	4	3
23	3	1	4	3	4
24	2	2	5	2	2
25	3	4	2	4	7
26	4	4	4	5	10
σ_d	1.553656	1.704745	1.541228	1.718676	2.584272

max_d	8	6	6	8	10
min_d	2	0	0	0	0
avg_d	4.423077	2.884615	3.153846	3.076923	4.038462

Tablica 4.4.2: Izmjerene udaljenosti od središta markera za kalibracije sa sljedećim brojem nasumično zadanih pozicija: 10, 20, 30, 40, 50. U tablici su prikazani i standardna devijacija, kao i najveća udaljenost, najmanja udaljenost i srednja vrijednost udaljenosti za svaku kalibraciju

Vizualni servoing u slučaju uključene informacije o dubini slike daje najbolje rezultate za kalibraciju s dvadeset i trideset nasumično zadanih pozicija. Daljnjim povećavanjem broja nasumično zadanih pozicija pri kalibraciji daje sve lošije rezultate. Isto je prikazano na slici 4.4.2.



Slika 4.4.2: Odstupanje alata robotskog manipulatora od središta markera za sve kalibracije

5. ZAKLJUČAK

Rezultat ovog rada je računalni program koji obavlja postupak kalibracije robotskog manipulatora i kamere. Nadalje, program obavlja postupak testiranja dobivenih kalibracija tako da se alat manipulatora postavlja na sto zadanih pozicija, te se uzima snimka i određuje na kojoj poziciji u koordinatnom sustavu robotskog manipulatora se alat nalazi. Odnosno, računalni program obavlja transformaciju koordinata dobivenih kamerom u koordinatni sustav robota. Rad također uključuje i postupak pozicioniranja na principu vizualnog servoinga.

Pri rješavanju problema apsolutne orijentacije potrebno je voditi računa o tome da kamera ima dovoljno vremena za uzeti snimku prije nego što se robotski manipulator pomakne sa zadane pozicije na drugu. Treba uzeti u obzir da pomicanje alata robotskog manipulatora između dviju točaka nije uvijek istog trajanja. Ako se snimka dobije kada robotski manipulator još nije stigao na zadanu koordinatu, izračun ne može biti točan. Iz tog razloga, produljuje se vrijeme kalibracije jer robotski manipulator mora imati dovoljno vremena za pozicioniranje s jedne točke na drugu za svaku razliku između nasumično zadanih točaka. Tu ostaje neriješen problem vremenske optimizacije gibanja robotskog manipulatora između dvije točke.

Osim činjenice da bez riješenog spomenutog problema optimizacije kalibracija traje dulje za veći broj točaka, rezultati kalibracije se poboljšavaju. Tako je za slučaj kalibracije s pedeset nasumičnih točaka greška između zadanih koordinata i transformiranih koordinata manja u odnosu na rezultate testova ostalih kalibracija. Nadalje, sva su mjerenja dobivena transformacijom pozicija iz koordinatnog sustava kamere u koordinatni sustav robota vrijednosti unutar 5mm od pozicija zadanih robotu.

Na temelju dobivenih rezultata provođenjem pozicioniranja na temelju vizualnog servoinga, vidljivo je da povećanje broja nasumično zadanih pozicija pri kalibraciji ne rezultira boljom točnošću sustava. Mogući razlog jeste da se povećanjem broja pozicija povećava vjerojatnost pojavljivanja većih outliera, na koje je primjenjena metoda kalibracije zasnovana na kvadratnom odstupanju osjetljiva. Kako bi se dalje smanjilo vrijeme kalibracije, potrebno je smanjiti trajanje pozicioniranja između dvije točke. To se može izvesti na način da kamera uzima snimku dok traje kalibracija, te da, kada se alat nalazi na istom mjestu na više slika, sprema tu poziciju kao koordinatu dobivenu kamerom, nakon čega se robotski manipulator pozicionira na iduću poziciju. S tako dobivenim koordinatama bi se trebao dalje raditi izračun kojim se dobiva homogena transformacijska matrica.

LITERATURA

- [1] B.K.P. Horn, Robot Vision, The MIT Press, Cambridge, Massachusetts, 1986.
- [2] K. Hashimoto, Visual Servoing – Real-Time Control od Robot Manipulators Based on Visual Sensory Feedback, World Scientific Publishing Co. Pte. Ltd., Singapur.
- [3] B. K. P. Horn, Closed-form solution of absolute orientation using unit quaternions, No. 4, Vol. 4, str. 629 – 642, Honolulu, Hawaii, 1986.
- [4] Astra - Orbbec, <https://orbbec3d.com/product-astra/>, Veljača 2017.
- [5] P. Đurović, R. Grbić, R. Cupec, D. Filko, Low Cost Robot Arm with Visual Guided Positioning, 40th Jubilee International Convention on Information and Communication Technology, Electronics and Microelectronics, Opatija, Hrvatska, 22.-26. 05. 2017.
- [6] A. Kaehler and G. Bradski, *Learning OpenCV 3*, Sebastopol, CA: O'Reilly, 2017.
- [7] OpenCV library, <http://opencv.org/>, Veljača 2017.
- [8] ArUco: a minimal library for Augmented Reality applications based on OpenCV, <https://www.uco.es/investiga/grupos/ava/node/26>, Veljača 2017.
- [9] OpenCV: cv::Mat Class Reference - OpenCV Documentation, http://docs.opencv.org/3.1.0/d3/d63/classcv_1_1Mat.html, Veljača 2017.
- [10] Petra Đurović, Robert Cupec, Ratko Grbić, Damir Filko, Robot Manipulation Module – Documentation, FERIT Osijek, 2017.
- [11] Operations on Arrays — OpenCV 2.4.13.2 documentation, http://docs.opencv.org/2.4/modules/core/doc/operations_on_arrays.html, Veljača 2017.

SAŽETAK

Razvijen je računalni program za kalibraciju robotskog manipulatora i kamere, koji na temelju niza slika dobivenih kamerom određuje odnos koordinatnog sustava kamere i koordinatnog sustava robotskog manipulatora. Točnost kalibracije ispitana je pokusima s robotskim manipulatorom. Kalibracija se vrši nad *Dobot ARM V1.0* te RGB-D *Orbbec Astra S* kamerom

U radu je obrađena primjena računalnog vida, kao i teorijsko rješenje problema kalibracije primjenom rješenja zatvorenog oblika. Navedeno rješenje kao rezultat daje homogenu transformacijsku matricu između koordinatnog sustava korištenog robotskog manipulatora i kamere.

Točnost kalibracije, koja se postiže razvijenim programom, eksperimentalno je ispitana mjerenjem odstupanja pozicija alata izmjerenih sustavom računalnog vida od pozicija zadanih robotskom manipulatoru. Također je ispitana i primjena razmatrane kalibracije za vizualni servoing.

Ključne riječi: robotski vid, kalibracija, robotski manipulator, RGB-D kamera, vizualni servoing

ABSTRACT

A computer program for calibration of a robotic arm and a camera has been developed, which, based on series of images obtained by the camera, determines the relative pose between the camera coordinate systems and the robotic arm coordinate system. Calibration accuracy has been tested experimentally. Calibration is performed on Dobot ARM V1.0 and RGB-D *Orbbec Astra S* camera.

Application of computer vision in robotics is discussed. A teoretical closed-form solution to the considered calibration problem is described, which gives a homogenous transformation matix between the camera coordinate system and the robotic arm coordinate system as the result.

Calibration accuracy, achieved by the developed program, is tested experimentally by measuring the difference between the tool positions measured by computer vision and the positions given to the robotic arm. Furthermore, application of the considered calibration for visual servoing was tested.

Keywords: robotic vision, calibration, robotic arm, RGB-D camera, visual servoing

ŽIVOTOPIS

Marina Tošeski rođena je 26. listopada 1993. godine u Osijeku, Hrvatska. Pohađanje Osnovne škole Ivana Kukuljevića Belišće započinje 2000., a završava 2008. godine. Srednjoškolsko obrazovanje započinje 2008. godine upisom u Opću Gimnaziju Srednje škole Valpovo. Srednju školu završava 2012. godine polaganjem državne mature, nakon čega iste godine upisuje Preddiplomski sveučilišni studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek Sveučilišta Josipa Jurja Strossmeyera u Osijeku. Godine 2015. završava Preddiplomski sveučilišni studij računarstva te upisuje Diplomski sveučilišni studij računarstva, elektivni blok Procesno računarstvo, na istom fakultetu.