

Markerska lokalizacija i registracija temeljena na medicinskom stereovizijskom sustavu

Gugo, Bruno

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:882919>

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

Download date / Datum preuzimanja: **2024-12-18**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Bruno Gugo

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:

Doc. dr. sc. Marko Švaco, mag. ing. mech.

Student:

Bruno Gugo

Zagreb, 2024.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru doc. dr. sc. Marku Švaci, asistentu Branimiru Čaranu, mag. ing. i asistentici Tari Knežević, mag. ing. mech. na pruženoj pomoći i korisnim savjetima prilikom izrade ovog završnog rada.

Posebnu zahvalu želim dati svojim roditeljima na beskrajnoj ljubavi, potpori i bodrenju koje su mi pružili tijekom preddiplomskog studija. Hvala svim mojim prijateljima i kolegama te prvenstveno mojoj djevojci, na svakom izlasku, kavi, piću i svemu ostalom što je ispunilo moju studentsku svakodnevicu trenutcima koje ću pamtiti do kraja života.

Bruno Gugo



Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 – 04 / 24 – 06 / 1	
Ur.broj: 15 – 24 –	

ZAVRŠNI ZADATAK

Student: **Bruno Gugo**

JMBAG: 0035239554

Naslov rada na hrvatskom jeziku: **Markerska lokalizacija i registracija temeljena na medicinskom stereovizijskom sustavu**

Naslov rada na engleskom jeziku: **Marker based localization and registration with a medical stereovision system**

Opis zadatka:

Proces lokalizacije pacijenata i ljudske anatomije u računalom navodenim operacijama (engl. *image guided surgery – IGS*) temelji se na više različitih principa koji se u načelu mogu podijeliti u markerske (engl. *marker based*) i bezmarkerske metode (engl. *markerless*). Cilj procesa lokalizacije je ostvariti generiranje koordinatnog sustava (matrice transformacije) povezane uz objekt od interesa u definiranom radnom prostoru te zadanom referentnom koordinatnom sustavu. Proces registracije označava povezivanje dva koordinatna sustava i izračun njihove međusobne transformacije. Registracija je potrebna kako bi se planirane operativne točke iz prostora slike (engl. *image space*) transformirale u fizički prostor (engl. *physical space*).

U ovom završnom radu potrebno koristeći marker povezan uz 3D printanu koštanu strukturu (kost lubanje, femura ili tibije) ostvariti digitalizaciju površine koristeći ručno vođeni alat (engl. *stylus*).

U sklopu rada potrebno je:

- upoznati se s radom medicinskog stereovizijskog sustava za praćenje krutih (engl. *rigid*) markera
- povezati računalo i medicinski stereovizijski sustav za praćenje krutih markera te ostvariti komunikaciju putem TCP/IP ili serijskog protokola
- upoznati se s karakteristikama zapisa u obliku oblaka točaka (engl. *point cloud*)
- digitalizirati i pretvoriti u datoteku oblaka točaka nekoliko objekata od interesa u fizičkom prostoru
- proučiti metode za krutu registraciju (engl. *rigid registration*) dva oblaka točaka
- primijeniti metodu kao što je PCA (engl. *principal component analysis*) za grubu te ICP (engl. *iterative closest point*) za preciznu registraciju
- na eksperimentalnom laboratorijskom postavu demonstrirati točnost i ponovljivost implementiranog procesa lokalizacije i registracije između objekta u prostoru slike (CT snimka ili 3D model) i objekta u fizičkom prostoru (stvari 3D printani objekt).

Funkcionalnosti iz završnog rada potrebno je demonstrirati na računalnoj opremi te na stereovizijskom sustavu za praćenje u Regionalnom centru izvrsnosti za robotske tehnologije.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

Datum predaje rada:

Predviđeni datumi obrane:

24. 4. 2024.

2. rok (izvanredni): 11. 7. 2024.
3. rok: 19. i 20. 9. 2024.

2. rok (izvanredni): 15. 7. 2024.
3. rok: 23. 9. – 27. 9. 2024.

Zadatak zadao:

doc. dr. sc. Marko Švaco

Predsjednik Povjerenstva:

izv. prof. dr. sc. Petar Čurković

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	III
POPIS TABLICA.....	V
POPIS OZNAKA	VI
SAŽETAK.....	VII
SUMMARY	VIII
1. UVOD.....	1
2. OPTIČKO PRAĆENJE	2
2.1. Princip rada	2
2.1.1. Aktivni i pasivni markeri	3
2.2. Northern Digital Inc. (NDI)	4
2.2.1. Polaris Spectra	4
3. OBLAK TOČAKA.....	7
3.1. Karakteristike oblaka točaka	8
3.1.1. Statističko uklanjanje šuma (eng. Statistical outlier removal).....	8
3.1.2. Voxel – grid filtriranje	9
3.2. Kruta registracija oblaka točaka.....	10
3.2.1. Normal distributions transform (NDT).....	11
3.2.2. Singular value decomposition (SVD).....	12
3.2.3. Principal component analysis (PCA).....	13
3.2.4. Iterative closest point (ICP)	15
3.2.4.1. Globalno optimalni ICP (Go-ICP).....	17
4. DIGITALIZACIJA POVRŠINE	19
4.1. Rješavanje matrica homogenih transformacija	19
4.2. Pivot kalibracija	21
4.2.1. Rasipanje prilikom pivot kalibracije	22
4.3. Stvaranje oblaka točaka	23
5. TESTIRANJE ALGORITAMA ZA REGISTRACIJU.....	26
5.1. Dobivanje oblaka točaka iz CAD modela.....	26
5.2. PCA za grubu registraciju	27
5.3. Go- ICP za preciznu registraciju	29
6. 3D SLICER.....	31
6.1. PLUS.....	31
6.2. Moduli <i>Transforms</i> i <i>Markups</i>	34
6.2.1. Orijentiranje modela stylusa i kosti tibije	35
7. VALIDACIJA I PROCJENA GREŠKE	37
8. ZAKLJUČAK.....	46
LITERATURA.....	47

PRILOZI..... 48

POPIS SLIKA

Slika 1. Stereovizija kod optičke kamere [1]	2
Slika 2. Pasivni markeri [2]	3
Slika 3. Aktivni markeri [2]	3
Slika 4. Polaris Spectra [3]	4
Slika 5. Mjerni volumen- Polaris Spectra [4]	5
Slika 6. Upravljačka jedinica sustava	6
Slika 7. Oblak točaka dobiven LiDAR tehnologijom [5]	7
Slika 8. Statističko uklanjanje šuma [6]	9
Slika 9. Voxel-grid filter	10
Slika 10. Princip rada 2D NDT algoritma [8]	12
Slika 11. Utjecaj šuma na PCA registraciju [9]	14
Slika 12. Problem lokalnog minimuma [11]	15
Slika 13. Point-to-point ICP [12]	16
Slika 14. Point-to-plane ICP [12]	16
Slika 15. Ilustracija branch-and-bound algoritma [13]	17
Slika 16. Pojednostavljeni prikaz Go-ICP algoritma	18
Slika 17. Ručno vođeni alat s pasivnim markerima - stylus	19
Slika 18. Shema matrica transformacija	20
Slika 19. Pivot kalibracija [15]	22
Slika 20. <i>Lever-arm</i> efekt [15]	23
Slika 21. Digitalizacija površine miša te prikaz oblaka u Matlabu	24
Slika 22. 3D printana kost tibia	25
Slika 23. Šumovi u oblaku točaka	25
Slika 24. 3D printana testna pločica	26
Slika 25. Dobivanje oblaka točaka iz CAD modela pomoću MeshLab-a	27
Slika 26. Početni raspored oblaka točaka	28
Slika 27. Loše preklapanje PCA metodom	28
Slika 28. Preklapanje nakon Go-ICP algoritma	30
Slika 29. Rad algoritma pri polovičnoj digitalizaciji objekta	30
Slika 30. Korisničko sučelje 3D Slicer-a	31
Slika 31. Potrebne ekstenzije unutar 3D slicer-a [18]	32
Slika 32. PLUS server launcher	33
Slika 33. Izgled sučelja IGT modula	34
Slika 34. PLUS server nakon uspješnog povezivanja s kamerom	34
Slika 35. <i>Transforms</i> modul te <i>Python</i> konzola	35
Slika 36. <i>Markups</i> modul	35
Slika 37. Hijerarhija transformacija	36
Slika 38. 3D printana kost tibije sa statičnim markerom	37
Slika 39. Početni raspored oblaka točaka kosti tibije	38
Slika 40. Svojstveni vektori digitaliziranog oblaka točaka (plavom - svojstveni vektor najvećeg smjera rasipanja, zelenom - vektor drugog najvećeg smjera rasipanja, crnom - vektor najmanjeg smjera rasipanja)	38
Slika 41. Svojstveni vektori oblaka točaka CAD modela (plavom - svojstveni vektor najvećeg smjera rasipanja, zelenom - vektor drugog najvećeg smjera rasipanja, crnom - vektor najmanjeg smjera rasipanja)	39
Slika 42. Oblaci točaka nakon PCA registracije	39
Slika 43. Oblaci točaka nakon Go-ICP registracije	40
Slika 44. Rezultati Go-ICP algoritma kod oblaka s najvećim brojem točaka	40

Slika 45. Dijagram toka.....	41
Slika 46. Konačan rezultat	42
Slika 47. 3D pogled u Slicer-u	42
Slika 48. Dovođenje stylusa u definirane točke	43
Slika 49. Prikaz mjerenja u 3D Slicer-u	44
Slika 50. Greška kod nagiba stylusa.....	45

POPIS TABLICA

Tablica 1. Tehničke specifikacije NDI Polaris Spectra..... 5
Tablica 2. Rasipanje prilikom pivot kalibracije 23
Tablica 3. Rezultati nakon Go-ICP algoritma 29
Tablica 4. Rezultati mjerenja (mm)..... 44

POPIS KRATICA

Oznaka	Jedinica	Opis
RMS		root mean square
SVD		singular value decomposition
PCA		principal component analysis
ICP		iterative closest point
NDI		Northern Digital Incorporated
Go -ICP		globally optimal iterative closest point
TCP		transmission control protocol
IP		internet protocol
CAD		computer aided design
NDT		normal distributions transform

SAŽETAK

Procesi lokalizacije i registracije predstavljaju temelj današnjih računalno navođenih operacija. Zadatak ovog završnog rada bio je uz pomoć medicinskog stereovizijskog sustava koji koristi krute markere najprije napraviti proces registracije prelaženjem ručno vođenog alata preko površine objekta, prilikom čega je generiran oblak točaka vezan uz koordinatni sustav od interesa, a zatim obaviti proces krute registracije, to jest pronalaženja matrice transformacije između koordinatnog sustava generiranog oblaka točaka te oblaka točaka objekta u prostoru slike (3D model). U početnim poglavljima objašnjen je princip rada stereovizijskog sustava te je napravljen pregled dostupnih metoda i algoritama za preklapanje dva oblaka točaka kao što su PCA za grubu registraciju te ICP za preciznu registraciju, ali i poboljšana verzija ICP metode Go-ICP algoritam. Na kraju je napravljena integracija svega te je provedena procjena točnosti sustava.

Ključne riječi: računalno navođene operacije, registracija, lokalizacija, oblak točaka, PCA, ICP

SUMMARY

Processes of localization and registration represent the foundation of today's image-guided surgeries - IGS. The objective of this paper was to use a medical stereovision system that utilizes rigid markers to first perform the registration process by manually guiding a stylus over the surface of an object, during which a point cloud associated to the coordinate system of interest was generated. This was followed by a rigid registration process, i.e., finding the transformation matrix between the coordinate system of the generated point cloud and the point cloud of the object in the image space (3D model). The initial chapters explain the operating principle of the stereovision system and provide an overview of available methods and algorithms for overlapping two point clouds, such as PCA (Principal Component Analysis) for rough registration and ICP (Iterative Closest Point) for precise registration, along with the improved version of the ICP method, the Go-ICP algorithm (Globally Optimal – ICP). Finally, an integration of all the components was performed, and the accuracy of the system was evaluated.

Key words: image-guided surgeries (IGS), registration, localization, point cloud, PCA, ICP

1. UVOD

Područje medicinske robotike svakim danom sve više napreduje. Računalom navođene operacije sve se učestalije primjenjuju zbog karakteristika kao što su visoka preciznost, učinkovitost i sigurnost. Ovakve operacije uključuju upotrebu hardverskih i softverskih alata, 3D modela anatomije pacijenta, navigacijskih sustava te robotske asistencije pri čemu se kirurzima olakšava planiranje te sama izvedba operacije, a pacijentu omogućuje brži oporavak. Dvije glavne metode koje se koriste za navigaciju u većini današnjih računalno navođenih operacija jesu optičko i elektromagnetsko praćenje. Optičko praćenje se temelji na stereovizijskim kamerama koje prate poziciju i orijentaciju reflektirajućih markera u prostoru. S druge strane, elektromagnetsko praćenje se temelji na registriranju promjena unutar elektromagnetskog polja od strane mikrosenzora ugrađenih na medicinske instrumente te je glavna prednost ovakve tehnologije naspram optičkog praćenja što omogućuje praćenje bez potrebe za izravnom vidljivošću između senzora i izvora. Jedan od glavnih problema računalno navođenih operacija je kako odrediti i povezati koordinate pacijenta u stvarnom svijetu s točkama CT snimke ili 3D modela. Lokalizacija označava proces određivanja egzaktna pozicije pacijenta u fizičkom prostoru ili prostoru slike, a pojam registracije veže se uz određivanje prostorne transformacije između pacijenta u prostoru slike i stvarnom svijetu te je najutjecajniji čimbenik na točnost ovakvog sustava.

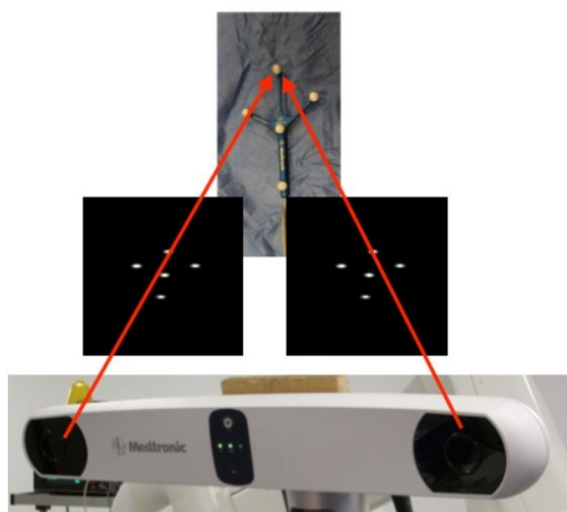
S obzirom na sve veći broj algoritama za preklapanje oblaka točaka te obećavajuću budućnost tehnologije oblaka točaka, ovaj završni rad nudi integraciju ove rastuće grane znanosti u računalno navođene kirurške zahvate.

2. OPTIČKO PRAĆENJE

Prve računalno potpomognute operacije s optičkim praćenjem značile su veliki iskorak na području medicinske tehnologije, prvenstveno u kirurgiji. Jedna od prvih takvih operacija obavljena je 1989. godine na Sveučilištu u Jeni, u Njemačkoj, gdje je optički sustav služio za navigaciju prilikom ortopedskog zahvata. Posljednjih 20-ak godina optičko praćenje doživljava ubrzan rast i razvoj te postaje neizostavna komponenta medicinske robotike. Moderni stereovizijski sustavi za optičko praćenje osiguravaju visoku preciznost i pouzdanost praćenja medicinskog alata u velikom volumenu mjerenja uz greške manje od pola milimetra. Svoju primjenu stereovizijske kamere našle su na raznim područjima u medicini, pa se tako koriste u neurokirurgiji, dentalnoj medicini, ortopedskoj kirurgiji, oftalmologiji i mnogim drugim.

2.1. Princip rada

Optičko praćenje radi na principu detektiranja infracrvenog svjetla koje markeri ili odašilju (aktivni markeri) ili reflektiraju (pasivni markeri), prilikom kojeg stereovizijski sustav ima ugrađene izvore infracrvenog svjetla kojeg emitiraju na mjereni volumen. Ovo infracrveno svjetlo detektiraju senzori dviju kamera koje generiraju 2D slike. Softver sustava obrađuje 2D slike te na principu računalne stereovizije izračunava 3D koordinate položaja. Računalna stereovizija radi na sličan način kao i ljudske oči percipiraju dubinu. Dvije ili više kamera snimaju isti objekt iz različitih kutova te na temelju razlike u perspektivama i principu triangulacije računalni algoritmi izračunavaju točne 3D koordinate položaja.

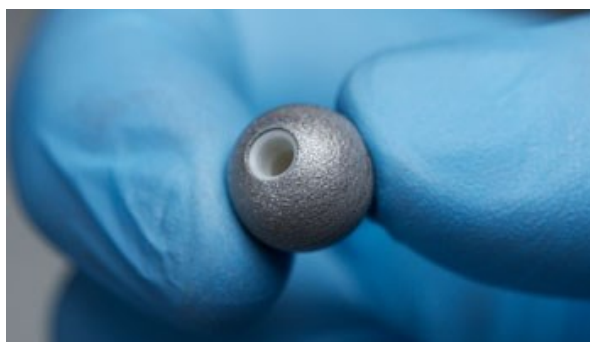


Slika 1. Stereovizija kod optičke kamere [1]

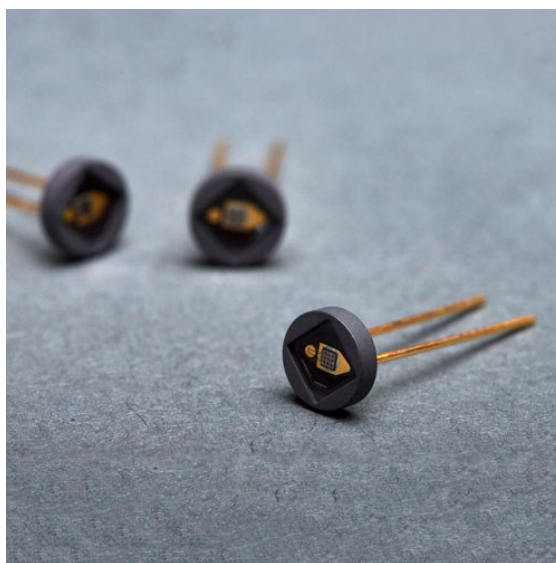
2.1.1. Aktivni i pasivni markeri

S obzirom na to emitiraju li ili reflektiraju infracrvenu svjetlost razlikujemo aktivne i pasivne markere. Pasivni markeri su sferična tijela reflektirajuće površine koji se pričvršćuju na krute objekte već predodređene geometrije. Bitno je naglasiti da se međusobna udaljenost markera na ovom krutom tijelu ne mijenja. Reflektirajuće površine pasivnih markera se vrlo lako izlizuju i troše što za posljedicu može imati vrlo slabo odbijanje infracrvenog svjetla, pa je ove markere potrebno često mijenjati. Aktivni markeri su infracrvene LED diode koje emitiraju infracrvenu svjetlost, a napajaju se iz dodatne upravljačke jedinice. Iako su dugotrajniji od pasivnih markera, znatno su skuplji.

Markeri se učvršćuju na medicinske alate ili anatomske strukture (npr. lubanju, kosti i sl.). S obzirom na to da je geometrija krutog tijela poznata, sustav može pratiti ne samo položaj nego i orijentaciju markera u prostoru. Orijentaciju krutog tijela moguće je odrediti uz uvjet da se na njemu nalaze najmanje tri markera koji formiraju ravninu u prostoru, dok je uz pomoć dva markera moguće odrediti samo smjer (dvije točke određuju pravac).



Slika 2. Pasivni markeri [2]



Slika 3. Aktivni markeri [2]

2.2. Northern Digital Inc. (NDI)

Northern Digital Incorporated (NDI) najveći je svjetski proizvođač medicinskih sustava za navigaciju. Ovu kanadsku tvrtku pokrenuli su 1981. godine diplomanti Sveučilišta Waterloo, koji su u to vrijeme dizajnirali znanstvene instrumente za industrijska mjerenja. Shvaćajući kakav potencijal njihova tehnologija ima, ubrzo su dizajnirali prvi optički sustav za praćenje pokreta. Početkom 90-ih tvrtka razvija *Optotrak*, stereovizijski sustav za praćenje infracrvenih markera, koji brzo postaje tržišni lider u svijetu 3D optičke navigacije. NDI predstavlja liniju *Polaris* 1996. godine te prvu malu, laganu i isplativu stereovizijsku kameru za potrebe lokalizacije. Do danas je NDI razvio nekoliko *Polaris* kamera:

- *Polaris Spectra*,
- *Polaris Vicra*,
- *Polaris Vega*,
- *Polaris Hybrid*.

Ovi modeli razlikuju se po točnosti, brzini praćenja te području mjernog volumena. U sklopu ovog završnog rada, koristila se *Polaris Spectra* stereovizijska kamera.



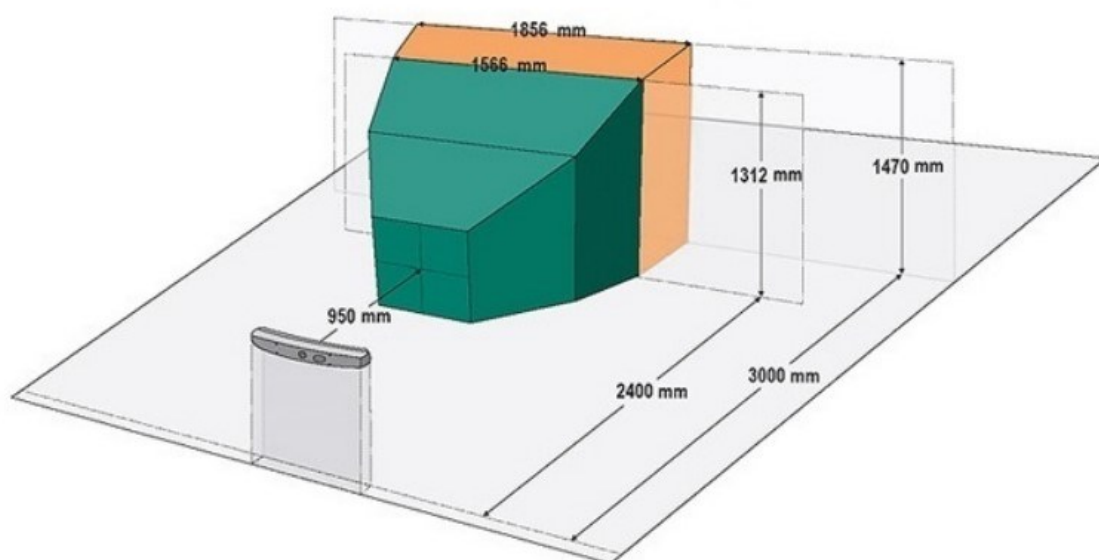
Slika 4. *Polaris Spectra* [3]

2.2.1. *Polaris Spectra*

Polaris Spectra jedan je od starijih modela serije *Polaris* koji je na tržište izašao 2008. godine. Na tržištu je bio do 2021. godine, kada ga je zamijenio novi model *Polaris Vega*, koji je sa sobom donio neka poboljšanja u smislu smanjenja greške prilikom dinamičkog praćenja te brzini odziva sustava. Iako bi se već mogao smatrati „zastarjelim“ uređajem, *Polaris Spectra* i dalje je sustav visoke preciznosti i pouzdanosti sa zadovoljavajućim performansama.

Tablica 1. Tehničke specifikacije NDI Polaris Spectra

Statička točnost	0,25 mm RMS
Dinamička točnost	0,5 mm RMS
Latencija	18 ms
Brzina ažuriranja	60 Hz
Domet mjerenja	3 m



Slika 5. Mjerni volumen- Polaris Spectra [4]

Stereovizijski sustav *Polaris Spectra* s računalom se povezuje i komunikaciju ostvaruje putem serijskog protokola. Serijski protokol je standard za komunikaciju između dva uređaja, prilikom čega podaci putuju jedan za drugim, bit po bit, preko standardnih međusklopova kao što su RS-232, RS-485 te USB. Serijska komunikacija koristi se za jednostavan i učinkovit prijenos podataka između računala i vanjskih perifernih uređaja (kamera, senzora, mikrokontrolera itd.). Treba dodati kako noviji modeli serije *Polaris* koriste TCP/IP protokol za komunikaciju putem interneta s računalom.

Dakle, podaci iz kamere prvo putuju u upravljačku jedinicu sustava (engl. *system unit*), u kojoj se obavlja obrada 3D slike te izračunavanje pozicije markera. Ove informacije serijskom vezom

putem USB priključka dalje idu u računalo. Iz upravljačke jedinice također se proteže i kabel za napajanje.

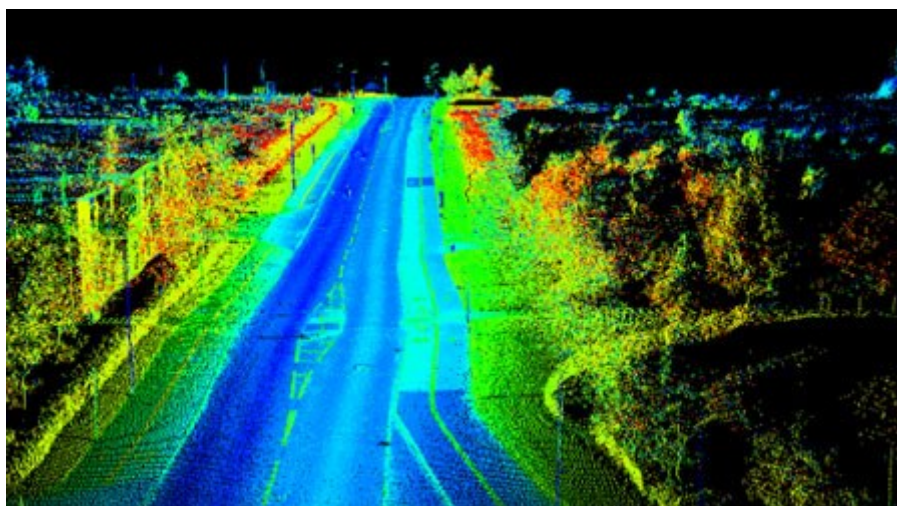


Slika 6. Upravljačka jedinica sustava

3. OBLAK TOČAKA

Termin oblak točaka (eng. *point cloud*) često je definiran kao skup podatkovnih točaka u trodimenzionalnom koordinatnom sustavu. On se prvi put počinje upotrebljavati sredinom 1960-ih kada se lasersko skeniranje počinje upotrebljavati za snimanje 3D podataka, tehnika koja je tada još bila vrlo skupa i komplicirana. Napretkom na području tehnologije LiDAR (engl. *Light Detection and Ranging*) 1990-ih prikupljanje i rad s oblacima točaka napravili su značajan korak naprijed. Ranih 2000-tih svijetu je predstavljena metoda terestričkog laserskog skeniranja (prikupljanja 3D prostornih podataka kao oblaka točaka), što je učinilo tehnologiju oblaka točaka široko dostupnom i jeftinom. Posljednjih desetljeća pojam oblaka točaka posebice se istaknuo u disciplinama kao što su arhitektura i građevinarstvo, gdje se veliki skupovi točaka koriste za rekonstrukciju zgrada, izgradnju novih, dobivanje podataka o terenima i objektima te mnogočemu drugom.

Tehnologija oblaka točaka revolucionizirala je način na koji prikupljamo, analiziramo i koristimo 3D podatke. Zadnjih par godina svoju primjenu pronalazi i na području automatizacije, strojnog učenja i umjetne inteligencije gdje, za sada, upotreba oblaka točaka izgleda jako obećavajuće.



Slika 7. Oblak točaka dobiven LiDAR tehnologijom [5]

3.1. Karakteristike oblaka točaka

Oblak točaka može nositi puno više informacija od samo 3D koordinata položaja svake točke. Tako svaka točka može imati i informaciju kao što je:

- boja – ako je oblak točaka snimljen RGB-D kamerom (eng. *Red-Green-Blue and Depth camera*),
- intenzitet – ako je oblak točaka snimljen LiDAR tehnologijom,
- reflektivnost,
- vektor normale.

Oblaci točaka često znaju sadržavati šumove, odnosno točke koje nisu dio pravog objekta te su se greškom našle u tom skupu podataka. Nadalje, mogu sadržavati područja praznina te područja iznimne gustoće točaka. U nekim slučajevima oblaci točaka mogu sadržavati milijune ili čak milijarde točaka te je s ovolikom količinom podataka jako teško baratati. Iz navedenih razloga oblake točaka potrebno je prvo filtrirati i klasificirati kako bi bili prikladni za daljnju upotrebu i analizu. Neke od često korištenih metoda u praksi su statističko filtriranje, filtriranje na temelju broja susjeda, projekcijsko filtriranje te voxel-grid filtriranje. S obzirom na to da se u sklopu ovog rada koristilo statističko te voxel-grid filtriranje, o njima će biti nešto više riječi u narednim poglavljima.

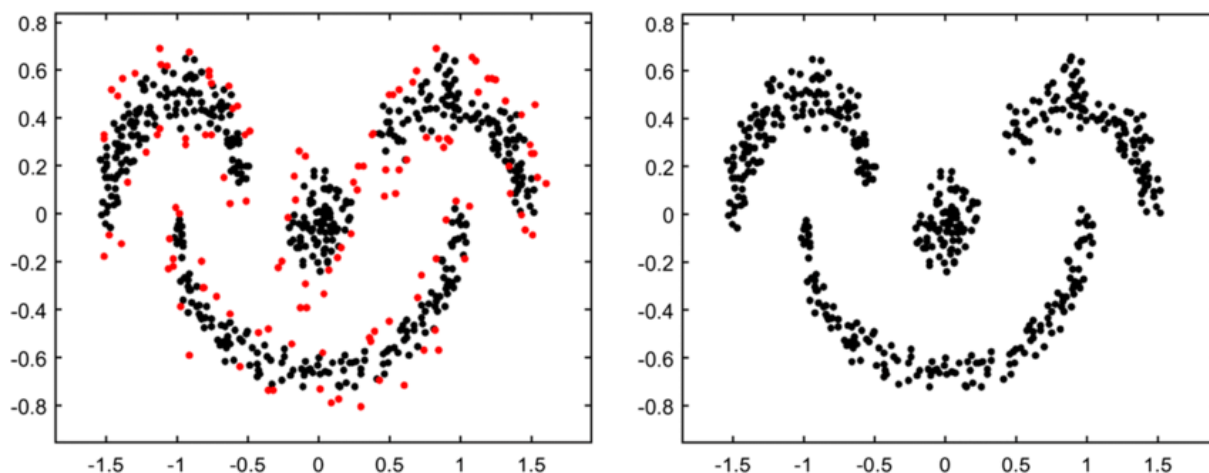
3.1.1. Statističko uklanjanje šuma (eng. *Statistical outlier removal*)

Statističko filtriranje tehnika je koja se koristi za uklanjanje šuma iz oblaka točaka, identificiranjem i uklanjanjem točaka koje značajno odstupaju od lokalne distribucije. Metoda je vrlo jednostavna te učinkovita u otklanjanju velikih šumova. Matematika koja stoji iza statističkog uklanjanja dosta je trivijalna.

Za svaku točku oblaka p_i te za zadani broj susjednih točaka k , moguće je odrediti srednju udaljenost d_i do susjednih točaka p_j . (1)

$$d_i = \frac{1}{k} \sum_{j=1}^k \|p_i - p_j\| \quad (1)$$

One točke koje imaju veću srednju udaljenost od definiranog praga se smatraju šumovima i odbacuju.



Slika 8. Statističko uklanjanje šuma [6]

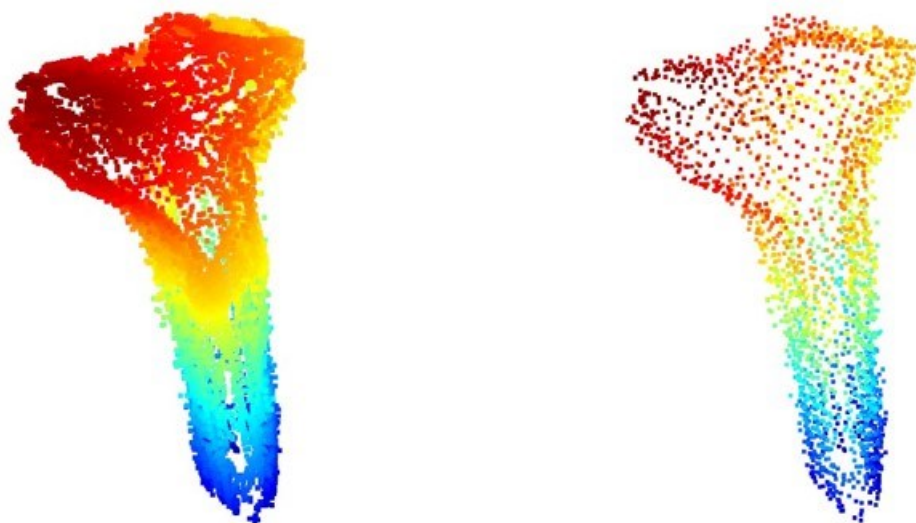
Međutim, prilikom upotrebe statističkog uklanjanja šuma treba biti vrlo oprezan. U nekim slučajevima, pogrešan odabir parametara broja susjednih točaka i praga rezultira u uklanjanju korisnih informacija, odnosno onih točaka koje nisu šum iz oblaka točaka.

3.1.2. *Voxel – grid filtriranje*

Voxel-grid filter jedna je od tehnika smanjivanja oblaka točaka (engl. downsampling), koja radi na principu stvaranja rešetke sitnih kockica podjednake dimenzije (voksela) u 3D prostoru. Točke se grupiraju u pojedine voksele na temelju svojih koordinata te ovisno o vrsti voxel-grid filtriranja zamjenjuju jednom reprezentativnom točkom.

Ovisno kojom se točkom zamjenjuje skup točaka unutar voksel, ova metoda u principu ima dva pristupa: približno (eng. approximate voxel-grid filtering) te filtriranje s obzirom na težište točaka (eng. centroid voxel-grid filtering). Kod približnog filtriranja sve točke unutar voksel se zamjenjuju točkom u centru samog voksel te je ova tehnika nešto manje precizna. Druga metoda računa težište svih točaka unutar voksel te ih zamjenjuje jednom točkom koja se nalazi u istom.

Voxel-grid filtriranje značajno smanjuje broj točaka u oblaku te tako ubrzava daljnje procese obrade oblaka točaka kao što je registracija, vizualizacija ili rekonstrukcija. S druge strane, ova metoda može djelovati negativno, smanjujući preciznost i detalje oblaka točaka, pa je veličinu voksel potrebno prilagoditi situaciji.



Slika 9. Voxel-grid filter

3.2. Kruta registracija oblaka točaka

2019. godine katedrala Notre-Dame u Parizu doživjela je razoran požar, koji je uništio drveni krov te većinu unutrašnjosti katedrale. Jedan od simbola Francuske trebao je obnovu, a inženjeri su bili suočeni s pitanjem kako rekonstruirati katedralu točno onakvom kakva je bila. Igrum slučaja, profesor umjetnosti Andrew Tallon 2010. godine odlučio je napraviti digitalnu kopiju katedrale uz pomoć 3D laserskog skenera. Tallon je pet dana proveo skenirajući spomenik izvana i iznutra te je pritom napravio preko 50 oblaka točaka iz desetak kutova. Prostornom registracijom te preklapanjem ovih oblaka točaka inženjeri su uspjeli dobiti potpunu 3D rekonstrukciju te uz pomoć nje napraviti temelje za početak obnove katedrale Notre - Dame. [7]

Registracija je proces određivanja prostorne transformacije između dva skupa točaka, omogućujući njihovo preklapanje unutar zajedničkog koordinatnog sustava. Ova fundamentalna tehnika postala je ključna u raznim domenama kao što su automatizacija, robotika, strojni vid, medicinska navigacija, arhitektura itd. Svaki proces registracije sastoji se od dva koraka; traženje korespondencija te izračunavanje matrice transformacije. S obzirom na prirodu registracije razlikujemo krutu (eng. *rigid registration*) i deformabilnu registraciju (eng. *deformable registration*).

Deformabilna registracija koristi se kod elastičnih objekata kada je uz rotaciju i translaciju oblak točaka potrebno skalirati te dodatno deformirati kako bi se postiglo preklapanje. Tipične primjene deformabilnih registracija mogu se naći u zdravstvu kada je potrebno registrirati model nekog raka tkiva, mozga itd.

Kruta registracija uključuje pronalaženje odgovarajuće translacije i rotacije između parova točaka koje međusobno ne mijenjaju udaljenost. Posljednjih nekoliko desetljeća opsežno je proučavana, a posebno je zanimljiva u medicinskom kontekstu, gdje predstavlja temelj računalno navođenih operacija.

Iako je broj algoritama za krutu registraciju svakim danom sve veći te je podjela jako puno, ovaj završni rad će se dotaknuti četiri najraširenija i najfundamentalnija algoritma za grubu registraciju:

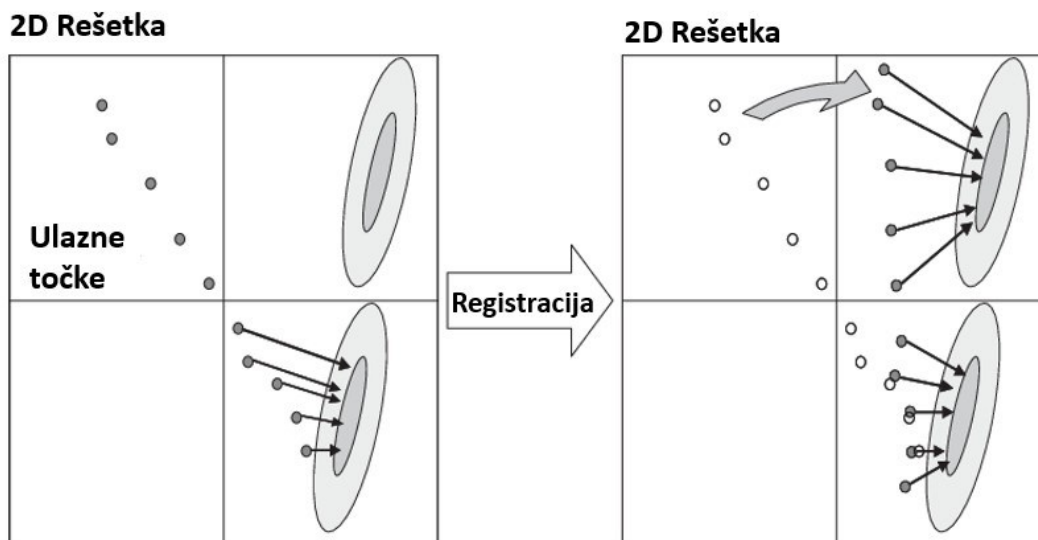
- NDT (eng. *Normal Distributions Transform*),
- SVD (eng. *Singular Value Decomposition*),
- PCA (eng. *Principal Component Analysis*),
- ICP (eng. *Iterative Closest Point*).

Odabir algoritma za registraciju ovisi o potrebnoj razini preciznosti, složenosti, veličini te početnom rasporedu oblaka točaka.

3.2.1. Normal distributions transform (NDT)

NDT (engl. Normal Distributions Transform) je široko korištena metoda za registraciju oblaka točaka, koju su prvi put svijetu predstavili 2003. godine profesori Peter Biber i Wolfgang Strasser sa Sveučilišta u Tübingenu. U početku namijenjen za 2D oblake točaka, algoritam je kasnije proširen na 3D te se danas koristi u robotici, autonomnim vozilima te računalnom vidu.

Za razliku od drugih metoda koje koriste ili sve točke ili značajnu količinu točaka, NDT koristi Gaussovu distribuciju točaka. Kao kod voxel-grid filtriranja, prostor se prvo segmentira na voksele. Za svaki voksel se izračunava normalna distribucija, odnosno težište točaka i kovarijanca, koje najbolje opisuje skup podataka unutar tog voksel. Nakon toga, algoritam uzima točke iz drugog skupa točaka te ih iterativno poravnava s ovim distribucijama u ćelijama prvog oblaka točaka, sve dok vjerojatnost svih točaka da pripadaju distribuciji unutar određenog voksel ne bude dovoljno velika.



Slika 10. Princip rada 2D NDT algoritma [8]

NDT je učinkovit, brz i robustan algoritam, koji se uglavnom koristi za iznimno velike oblake točaka, kao što su prostorne mape. Ima visoku otpornost na šumove, međutim potrebno je dosta dobro početno poravnanje oblaka točaka.

3.2.2. Singular value decomposition (SVD)

Singularna dekompozicija (engl. Singular Value Decomposition) vrlo je moćan alat linearne algebre koji se koristi u raznim granama znanosti, a između ostalog i registraciji oblaka točaka. SVD se koristi za manje skupova točaka te kada su poznate korespondencije između parova točaka. Točnije, za svaku točku x_i iz prvog skupa znamo koja je to točka y_i u drugom skupu. Računski postupak traženja matrice transformacije pomoću SVD metode izgleda ovako:

Neka su A i B skupovi točaka između kojih znamo korespondenciju točaka.

$$A = \{x_1, x_2, x_3, \dots, x_n\}$$

$$B = \{y_1, y_2, y_3, \dots, y_n\}$$

Tražimo parametre transformacije koji će minimizirati ovu funkciju, odnosno rotaciju R i translaciju t za koju će ukupna udaljenost svih točaka između dva skupa biti minimalna. (2)

$$\min \rightarrow d^2 = \frac{1}{n} \sum_{i=1}^n \|y_i - (Rx_i + t)\|^2 \quad (2)$$

Računaju se težišta oba oblaka, kako bi se uklonila translacija. (3)

$$A_{centroid} = \frac{1}{n} \sum_{i=1}^n x_i \quad B_{centroid} = \frac{1}{n} \sum_{i=1}^n y_i \quad (3)$$

Određuje se matrica kovarijance H koja predstavlja povezanost orijentacija između dva oblaka točaka. (4)

$$H = \sum_{i=1}^n (x_i - A_{centroid})(y_i - B_{centroid})^T \quad (4)$$

Na matrici kovarijance primjenjuje se singularna dekompozicija, gdje U i V predstavljaju ortogonalne matrice koje opisuju rotaciju, a Σ dijagonalna matrica koja sadrži informacije o skaliranju. (5)

$$H = U\Sigma V^T \quad (5)$$

Rotacijska matrica R tada se dobiva iz ortogonalnih matrica U i V . (6)

$$R = UV^T \quad (6)$$

Nakon dobivanja rotacijske matrice, translacija se izračunava koristeći izračunata težišta te matricu rotacije. (7)

$$t = A_{centroid} - R B_{centroid} \quad (7)$$

Iako je algoritam brz, učinkovit i pouzdan, ograničen je jer je potrebno poznavati odnose između parova točaka u pojedinom skupu, što u praksi i nije baš čest slučaj.

3.2.3. *Principal component analysis (PCA)*

Analiza glavnih komponenti (engl. Principal Component Analysis) je statistička metoda redukcije linearne dimenzionalnosti velikih skupova podataka. Nerijetko otkriva ovisnosti između varijabli, koje na prvi pogled nisu lako uočljive te se primjenjuje u raznim poljima poput populacijske genetike, strojnog učenja i meteorologije. Svoju primjenu nalazi i u registraciji oblaka točaka. Glavna ideja je da se preklope vektori najvećih smjerova rasprostiranja oblaka točaka. Matematika koja stoji iza samog algoritma opisana je u nastavku.

Neka su A i B oblaci točkaka koje analiziramo. Slično kao i kod SVD metode prvo računamo težište točkaka. (8)

$$A_{centroid} = \frac{1}{j} \sum_{i=1}^j x_i \quad B_{centroid} = \frac{1}{k} \sum_{i=1}^k y_i \quad (8)$$

Od svake točke u oblaku se oduzimaju izračunata težišta, u svrhu stavljanja u središte. (9)

$$x_{i,mean} = x_i - A_{centroid} \quad y_{i,mean} = y_i - B_{centroid} \quad (9)$$

Nakon toga se računaju matrice kovarijance za oba oblaka. (10)

$$H_A = \frac{1}{j} \sum_{i=1}^j x_{i,mean} x_{i,mean}^T \quad H_B = \frac{1}{k} \sum_{i=1}^k y_{i,mean} y_{i,mean}^T \quad (10)$$

Primjenom dekompozicije svojstvenih vrijednosti nad matricama kovarijance dobivaju se svojstveni vektori te svojstvene vrijednosti. Ovi svojstveni vektori predstavljaju glavne osi (ili smjerove najveće varijance), a svojstvene vrijednosti iznos varijance duž tih osi. (11)

$$H_A v_A = \lambda_A v_A \quad H_B v_B = \lambda_B v_B \quad (11)$$

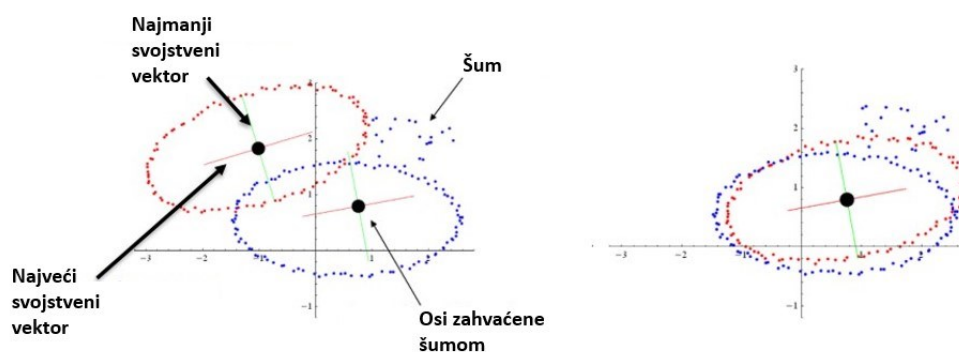
U jednadžbi (11) v_A i v_B predstavljaju svojstvene vektore, a λ_A i λ_B svojstvene vrijednosti oblaka točkaka A i B . Nakon što su izračunati svojstveni vektori potrebno je naći matricu rotacije koja preklapa glavne osi oblaka. (12)

$$R = v_A^T v_B \quad (12)$$

Translacija se dobije iz izračunatih težišta iz jednadžbe (8) te matrice rotacije:

$$t = A_{centroid} - R B_{centroid} \quad (13)$$

Pomoću PCA metode veliki skup točkaka reduciran je na samo dvije informacije: svojstvene vektore te iznose svojstvenih vrijednosti, što čini PCA algoritam moćnom tehnikom za predprocesuiranje podataka. Analiza glavnih komponenti se uglavnom koristi za grubu registraciju oblaka točkaka. Razlog tome je što je ova metoda izrazito osjetljiva na šumove te radi dobro samo ako je svaki oblak točkaka približno jednako i normalno raspoređen.



Slika 11. Utjecaj šuma na PCA registraciju [9]

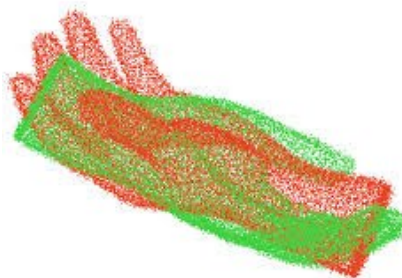
3.2.4. Iterative closest point (ICP)

Iterativna metoda najbližih točaka (engl. *Iterative closest point*) je algoritam koji se koristi za preciznu registraciju oblaka točaka iterativno minimizirajući razliku između parova točaka. Metodu su prvi predstavili Paul J. Besl i Neil D. McKay 1992. godine u svom radu „*A Method for Registration of 3-D Shapes*“.

Rad algoritma se može podijeliti na nekoliko koraka:

1. svaka točka izvornog oblaka točaka se povezuje s najbližom točkom referentnog oblaka (najbliža na primjer po principu euklidske udaljenosti),
2. nakon što se sve točke povežu određuje se rotacija i translacija koja minimizira metričku udaljenost između parova točaka iz prethodnog koraka (ovaj dio obično se vrši SVD metodom),
3. primjenjuje se transformacija,
4. iterativno se ponavljaju prethodni koraci dok ne dođe do konvergencije.[10]

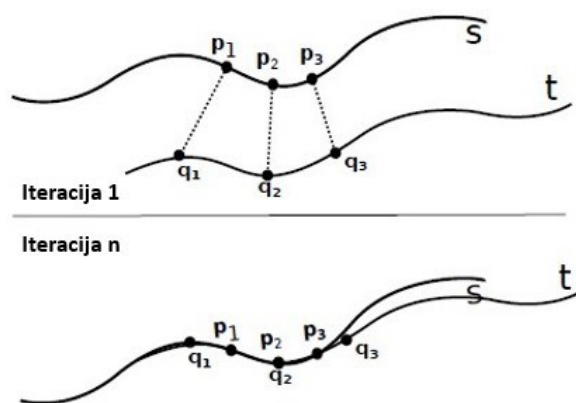
U praksi se obično koriste neki kriteriji konvergencije kod kojih algoritam staje nakon što su kriteriji ispunjeni. To može biti broj iteracija, gdje algoritam, nakon što dosegne maksimalan broj iteracija, a da prije toga u prethodnim iteracijama nije našao rješenje, staje. Nadalje, kriterij također može biti i zaostala greška (engl. *residual error*), odnosno udaljenost između odgovarajućih točaka. Ako je udaljenost parova točaka između dva oblaka ispod prethodno definiranog praga ili ako se ova razlika svakom novom iteracijom ne mijenja previše, algoritam zaustavlja s radom. Glavni nedostatak ICP algoritma je taj što zahtijeva dosta dobar početni raspored oblaka točaka. U protivnom, ako je raspored oblaka kompliciraniji (npr. jedan oblak točaka je zarotiran u odnosu na drugi za 180°) algoritam može zapeti u lokalnom minimumu, to jest može naći rješenje, iako je ono daleko od dobrog.



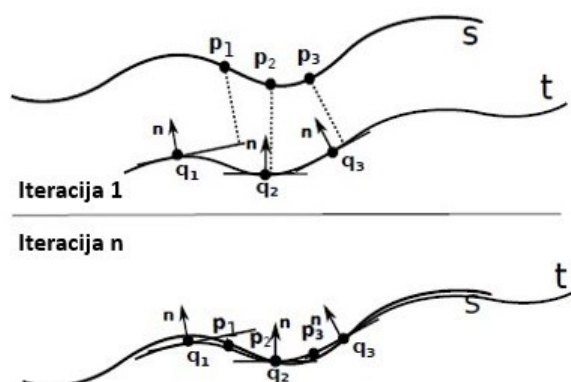
Slika 12. Problem lokalnog minimuma [11]

Danas je ICP varijanti jako puno te se one razlikuju s obzirom na način na koji konvergiraju ka rješenju. Tako, uz metodu povezivanja točke na točku (engl. *point-to-point ICP*) opisane u prvom koraku iteracije, razlikujemo još varijantu točka na ravninu (engl. *point-to-plane ICP*) te generalizirani ICP algoritam (engl. *generalized ICP*). Varijanta točke na ravninu umjesto minimiziranja udaljenosti od točke do točke minimizira udaljenost između točke u izvornom oblaku i tangentne ravnine najbliže točke u ciljnom oblaku, što dovodi do brže konvergencije. Generalizirani ICP algoritam koristi kombinaciju *point-to-point* i *point-to-plane* algoritama za pronalazak najoptimalnijeg rješenja.

S ciljem rješavanja problema lokalnog minimuma 2013. godine troje članova Instituta Inženjera Elektrotehnike i Elektronike (IEEE) objavilo je papir "*Go-ICP: Solving 3D Registration Efficiently and Globally Optimally*" u kojem opisuju novu varijantu ICP algoritma, koji će se kasnije ispostaviti kao jednom od najboljih verzija ovog tipa registracije. [11]



Slika 13. Point-to-point ICP [12]

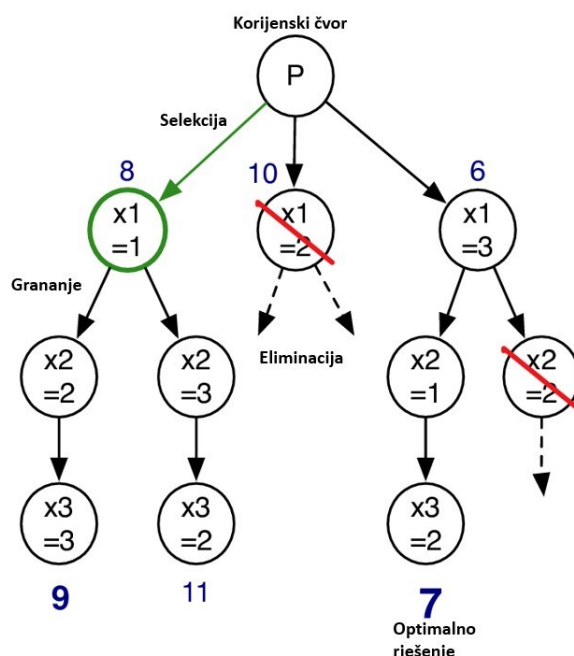


Slika 14. Point-to-plane ICP [12]

3.2.4.1. Globalno optimalni ICP (Go-ICP)

Globalno optimalni ICP algoritam (engl. *Globally optimal iterative closest point*) je integracija klasičnog ICP pristupa sa *branch-and-bound* metodom u svrhu smanjena mogućnosti konvergencije u lokalni minimum. Radi na principu ispitivanja te mijenjanja početnih orijentacija izvornog oblaka točaka kako bi se našlo globalno optimalno rješenje.

Branch-and-bound algoritam je tehnika koja se koristi u problemima kombinatorne optimizacije za sustavno traženje najboljeg rješenja. Djeluje tako da problem dijeli na manje podprobleme ili grane (eng. *branches*), a zatim eliminira određene grane na temelju granica optimalnog rješenja. Ovaj proces se nastavlja dok se ne pronađe najbolje rješenje ili dok se ne istraže sve grane. *Branch-and-bound* često je korištena metoda u informatici i umjetnoj inteligenciji.

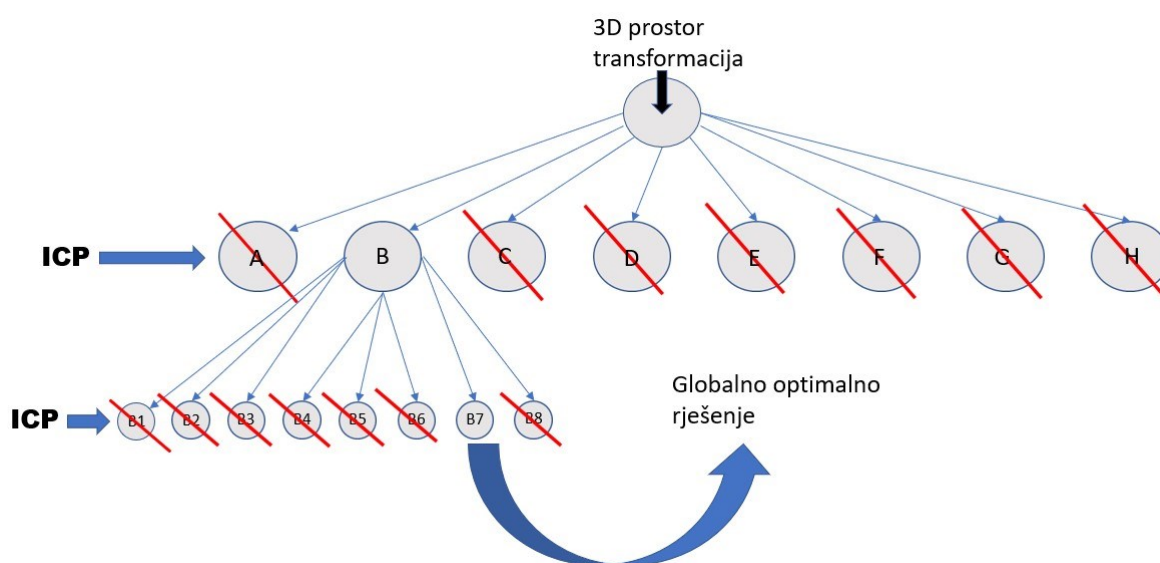


Slika 15. Ilustracija branch-and-bound algoritma [13]

U kontekstu *Go-ICP* registracije oblaka točaka *branch-and-bound* algoritam pretražuje prostor 3D transformacija (translacije i rotacije) kako bi našao najbolje rješenje. Prostor se rekurzivno dijeli na osam dijelova (grana) na principu oktostabla (eng. *octree*). Za svaku granu računa se donja granica pogreške registracije. Ako je donja granica za neku granu veća od trenutnog najboljeg rješenja, ta se grana odbacuje (obrezivanje). To pomaže u učinkovitom sužavanju regija prostora transformacije koje vjerojatno sadrže globalni minimum. Prilikom ovog procesa, za svaku kandidatsku transformaciju to jest za svaku granu, primjenjuje se ICP algoritam kako

bi se dodatno prilagodila transformacija i smanjila pogreška poravnanja. Ako se transformacija pokaže boljom od dosadašnjeg najboljeg rješenja, ona se uzima kao novo najbolje rješenje. Algoritam nastavlja pretragu i proces obrezivanja grana sve dok se sve obećavajuće regije prostora transformacije ne istraže ili obrežu te algoritam ostane s rješenjem koje konvergira u globalni minimum. [13]

Broj točaka u pojedinom oblaku te početni raspored uvelike utječu na vrijeme izvršavanja algoritma. Iako je dosta otporan na *outliere* (velike šumove), mali šumovi mogu dosta utjecati na sam proces dobivanja optimalnog rješenja, pa je potrebno koristiti neku od metoda filtriranja.



Slika 16. Pojednostavljeni prikaz Go-ICP algoritma

U sklopu ovog završnog rada za registraciju oblaka točaka bit će korišten Go-ICP algoritam u kombinaciji s PCA metodom za dobivanje matrice transformacije između prostora slike i objekta u stvarnom svijetu.

4. DIGITALIZACIJA POVRŠINE

Proces digitalizacije površine u ovom završnom radu označavat će postupak generiranja oblaka točaka površine objekta od interesa. Ideja je prikupljati točke prelaskom ručno vođenog alata s pasivnim markerima (eng. *stylus*) preko površine krutog objekta. Koordinate točaka će biti u odnosu na referentni statični koordinatni sustav, točnije kruti objekt s pasivnim reflektirajućim markerima. Prvi korak je bio riješiti matrice transformacija, kako bi se pronašle koordinate vrha *stylusa* u referentnom koordinatnom sustavu.



Slika 17. Ručno vođeni alat s pasivnim markerima - stylus

4.1. Rješavanje matrica homogenih transformacija

Matrice homogenih transformacija opisuju položaj i orijentaciju jednog koordinatnog sustava u odnosu na drugi. One su uvijek 4x4 dimenzija te se sastoje od matrice rotacije i vektora položaja. Njihov opći oblik izgleda ovako:

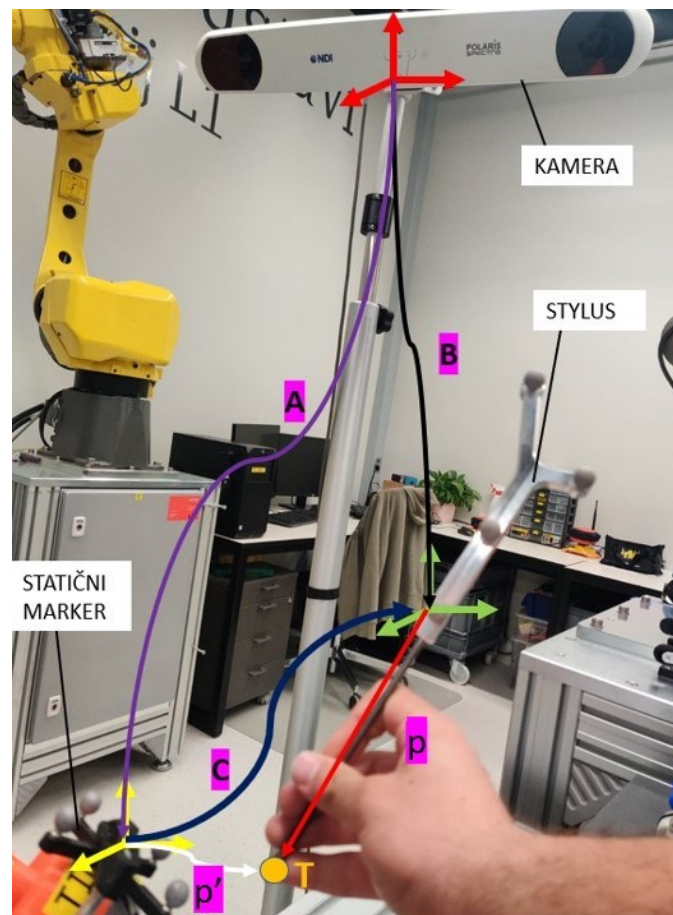
$${}^mT_n = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} i_x & j_x & k_x & p_x \\ i_y & j_y & k_y & p_y \\ i_z & j_z & k_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

gdje je R matrica rotacije, a p vektor pomaka koordinatnog sustava n u odnosu na koordinatni sustav m . U ovom slučaju poznate su matrice transformacija između koordinatnog sustava *stylusa* i kamere te između koordinatnog sustava statičnog referentnog sustava (pasivnih markera) i kamere. To su matrice transformacija koje šalje upravljački sustav kamere. U Pythonu postoji knjižnica *scikit-surgerynditracker* koja omogućuje komunikaciju s kamerom, u ovom slučaju NDI *Polaris Spectra* koja se koristila u ovom radu.

Prvo je potrebno definirati postavke NDI uređaja (tip kamere, portovi, ROM datoteke markera) te napraviti objekt u klasi *NDITracker*, nakon čega pozivom naredbe *start_tracking()* počinje praćenje kamere. Naredbom *get_frame()* se dobivljaju informacije koje kamera šalje računalu.

```
settings_polaris = {
    "tracker type": "polaris", "serial port": "COM3",
    "romfiles": ["C:/FEM0001.rom", "C:/8700340.rom", "C:/8700339.rom",
"C:/8700338.rom"]
}

tracker= NDITracker(settings_polaris)
tracker.start_tracking()
tracking=tracker.get_frame()
```



Slika 18. Shema matrica transformacija

Dakle, prema slici (18) jedino što je poznato su matrice transformacija A i B , a potrebno je naći vektor p' , to jest koordinate točke T (vrha igle *stylusa*) u koordinatnom sustavu statičnog markera. Treba naglasiti da su koordinatne osi sustava na slici (18) samo shematski prikazane te u stvarnosti nisu tako orijentirane.

Za početak jednostavnom matematikom pronalazimo matricu transformacija koordinatnog sustava pomičnog *stylusa* u referentnom koordinatnom sustavu statičnog markera. Neka je A matrica transformacija između statičnog markera i kamere, a B matrica transformacija između *stylusa* i kamere. Tada vrijedi :

$$AC = B$$

$$C = A^{-1}B$$

Odnosno, množenjem inverzne matrice transformacije A te matrice B dobije se matrica transformacije između *stylusa* i statičnog markera. Nadalje vrijedi:

$$p' = C \times p$$

gdje je p' traženi vektor, odnosno koordinate točke T u koordinatnom sustavu statičnog markera, a p predstavlja vektor između koordinatnog sustava *stylusa* te vrha igle *stylusa*, odnosno točke T . Vektor p , odnosno pomak vrha *stylusa* u odnosu na ishodište *stylusa*, nam nije poznat te ga je potrebno naći vrlo važnom tehnikom pivot kalibracije (engl. *pivot calibration*).

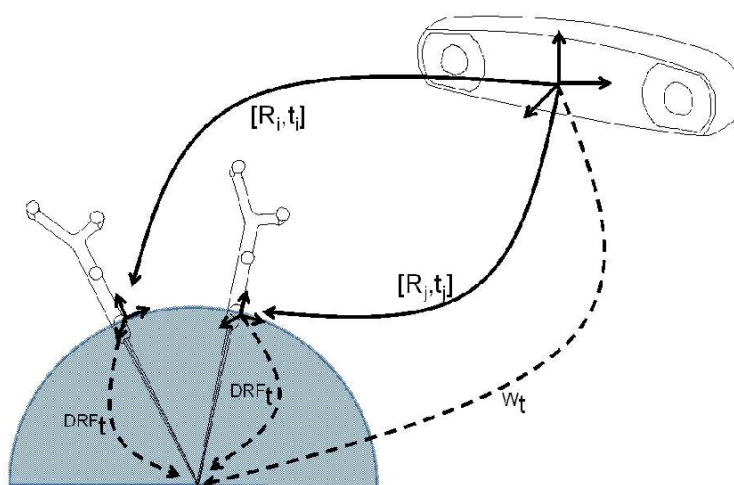
4.2. Pivot kalibracija

Pivot kalibracija ili kalibracija vrha alata je proces određivanja transformacije između koordinatnog sustava alata i vrha alata. Kalibracija alata igra ključnu ulogu u preciznosti i pouzdanosti računalno navođenih operacija. Najčešći način na koji se ovo radi je zakretanje praćenog alata oko fiksne točke u prostoru. Prilikom procesa se spremaju matrice transformacije između koordinatnog sustava kamere i koordinatnog sustava alata te se dalje obrađuju i koriste za izračun vrha alata. Prema profesoru Zivu Yanivu s Istraživačkog instituta u Izraelu postoji nekoliko načina na koje je moguće izračunati poziciju vrha alata:

- *sphere fitting* (SF),
- *algebraic one step* (AOS),
- *algebraic two step*. (ATS).

Sve ove metode rješavaju isti problem, ali uzimaju matrice transformacije drugačijim redoslijedom. U svom istraživanju Yaniv koristi NDI *Polaris Vicra* stereovizijsku kameru te ispituje svaku od ovih metoda procjenjujući vrijednost rasipanja pivot kalibracije. Yaniv zaključuje kako su algebarske formulacije (AOS i ATS) preciznije od *sphere fitting* metode sa srednjom vrijednošću greške od 0,25 mm naspram SF metode koja ostvaruje 0,35 mm. [14]

Nadalje, uočeno je da lokacija unutar mjernog volumena uređaja za praćenje ima mjerljiv učinak na preciznost i točnost rezultata te da za optimalne rezultate kalibraciju treba izvesti u blizini središta mjernog volumena. Treba još dodati kako ove metode ostvaruju približno iste rezultate i kod postojanja manjih šumova.



Slika 19. Pivot kalibracija [15]

U ovom završnom radu pivot kalibracija će se provesti pomoću IGT modula unutar 3D Slicera, softverskog paketa otvorenog koda za analizu slika i znanstvenu vizualizaciju o kojem će biti riječi kasnije.

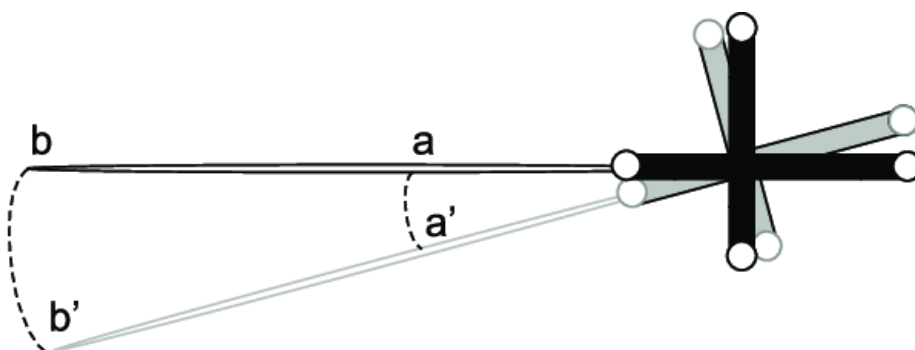
4.2.1. Rasipanje prilikom pivot kalibracije

U svrhu utvrđivanja koliko se vrijednosti točke u vrhu *stylusa* rasipaju, napravljeno je deset pivot kalibracija od kojih je svaka provedena u periodu od 20 sekundi. U 3D Sliceru potrebno je prvo instalirati ekstenziju IGT (engl. *Image-guided therapy*), alat napravljen upravo za računalno navođene operacije. IGT ima funkciju *Pivot Calibration*, koja nakon kalibracije izbacuje srednju kvadratnu pogrešku (engl. *root mean square error*) te matricu transformacije s položajem vrha. Rezultati deset ispitivanja su u tablici koja slijedi:

Tablica 2. Rasipanje prilikom pivot kalibracije

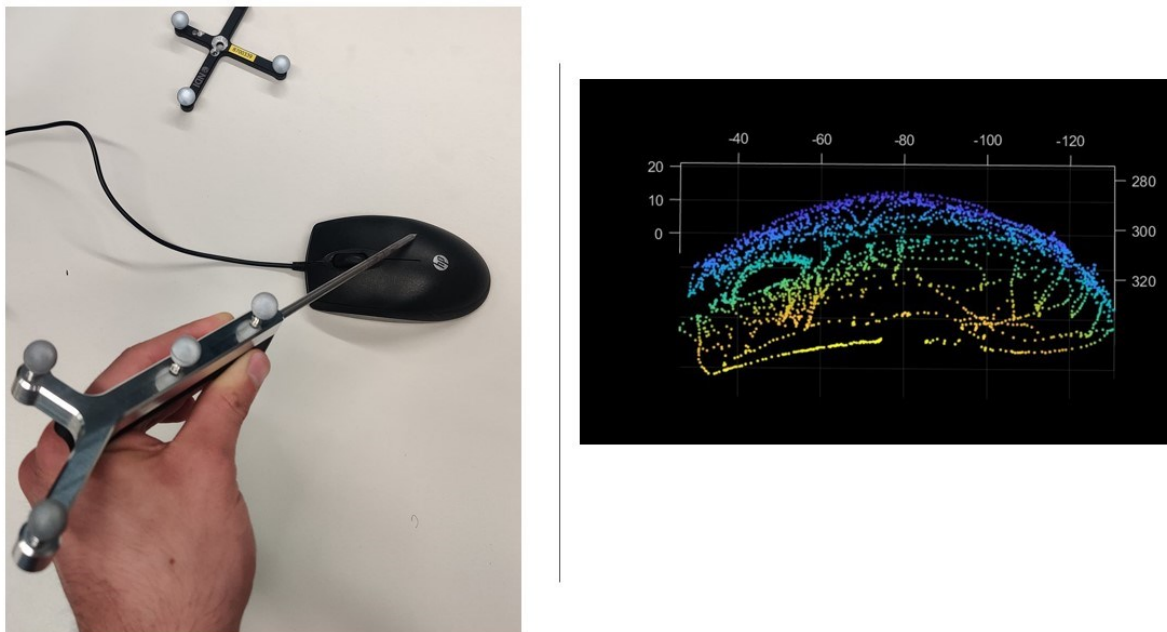
Broj ispitivanja	1	2	3	4	5	6	7	8	9	10
RMS greška (mm)	0,39	0,32	0,56	0,44	0,51	0,52	0,48	0,54	0,41	0,45

Izračunavanjem srednje vrijednosti pogreške dolazimo do vrijednosti od 0,46 mm. Treba uzeti u obzir da na kvalitetu pivot kalibracije uvelike utječu šumovi, duljina igle na alatu i vanjski reflektirajući predmeti u okolini. Što je duljina igle veća, to je i greška veća. Ovo se naziva *lever-arm* efekt. To je propagacija pogreške u mjerenju rotacije markera prema vrhu alata, a posljedično uzrokuje rasipanje vrijednosti vrha. [15]

Slika 20. *Lever-arm* efekt [15]

4.3. Stvaranje oblaka točaka

Nakon što je riješena matematika te je obavljena kalibracija vrha alata, može se početi sa digitalizacijom površine. Digitalizacija površine obavlja se prelaženjem *stylusa* preko površine objekta laganim i preciznim pokretima. Potrebno je osigurati jasnu liniju vidljivosti između alata, statičnog markera i kamere za vrijeme procesa. Vrijeme trajanja ovog procesa može ovisiti o veličini i složenosti površine, ali u prosjeku za oblak točaka koji sadrži oko 10 000 točaka potrebno je približno 6 minuta. Treba uzeti u obzir da će se prilikom procesa, zbog zakrivljenosti objekta koji se digitalizira, *stylus* nakrivljivati što će često dovesti do toga da kamera ne može prepoznati reflektirajuće markere.



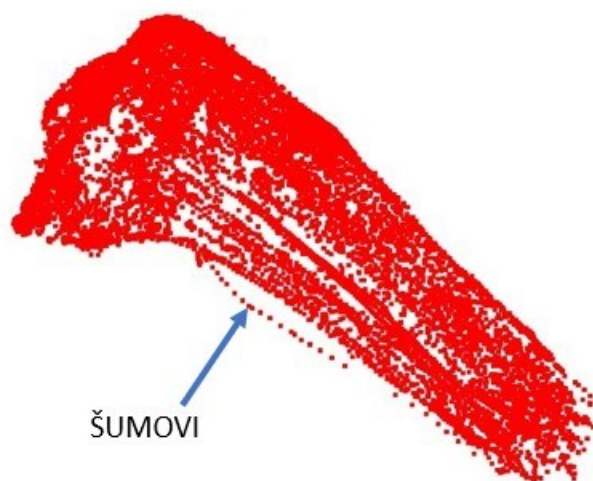
Slika 21. Digitalizacija površine miša te prikaz oblaka u Matlabu

Kako je alat ručno vođen normalno je za očekivati da će generirani oblak točaka sadržavati dosta šumova, odnosno točaka koje su se slučajno zabilježile „bježanjem“ ili drhtanjem ruke. Iz tog razloga će se za uklanjanje ovih šumova koristiti statističko uklanjanje šumova opisano u poglavlju 3.1.1. U Pythonu koristimo Open3D, biblioteku namijenjenu za obradu, vizualizaciju i rad s 3D podacima. Pozivom funkcije `remove_statistical_outlier()` izvršava se filtriranje, a ulazni parametri funkcije su `nb_neighbors` i `std_ratio` koji se prilagođavaju ovisno o tome želi li se provesti agresivnije ili manje agresivno filtriranje. Niža vrijednost `std_ratio` te veći `nb_neighbors` značit će agresivniji filter i obrnuto.

Svi ovi zabilježeni oblaci točaka spremat će se u obliku `.xyz` datoteke, gdje se koordinate točaka spremaju u redu jedna iza druge.



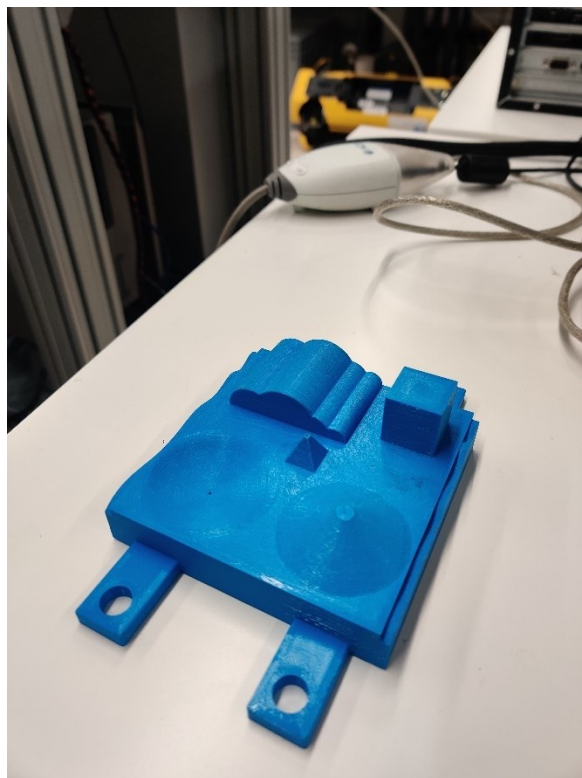
Slika 22. 3D printana kost tibia



Slika 23. Šumovi u oblaku točaka

5. TESTIRANJE ALGORITAMA ZA REGISTRACIJU

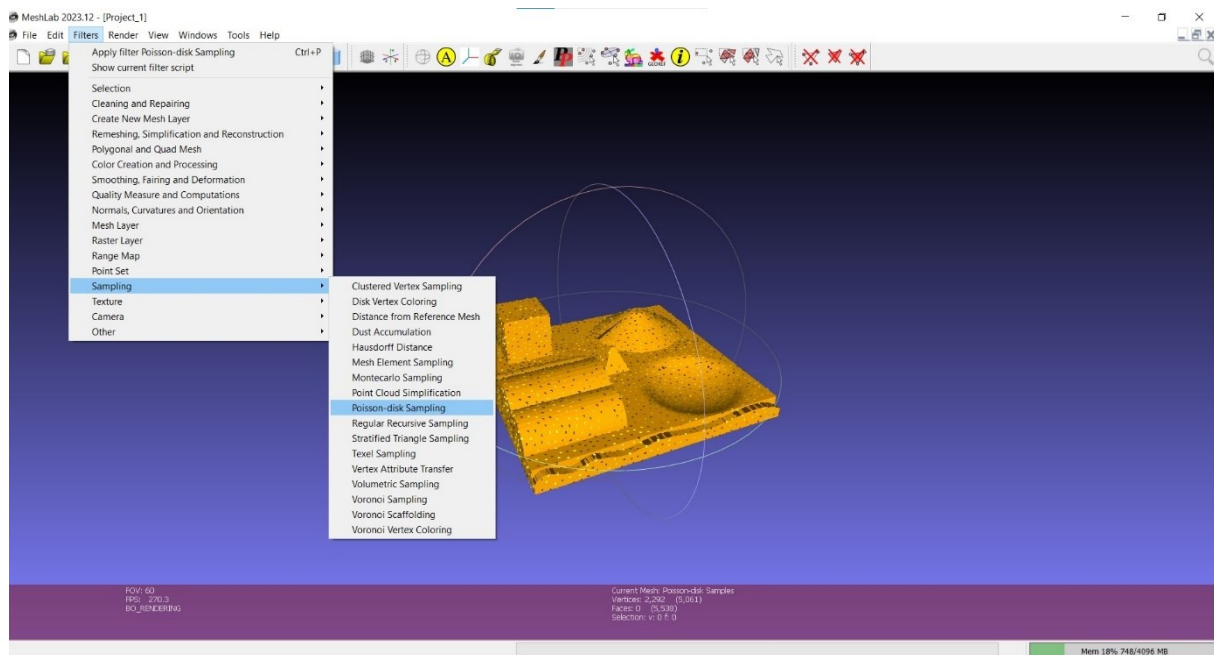
U svrhu početnih testiranja, pločica s nepravilnom površinom dimenzija 10x10 centimetara najprije je dizajnirana u SolidWorks-u, a zatim izrađena pomoću 3D printera. Za početak je izvršena digitalizacija, to jest oblak točaka površine u radnom prostoru, nakon čega je trebalo dobiti oblak točaka iz CAD modela pločice.



Slika 24. 3D printana testna pločica

5.1. Dobivanje oblaka točaka iz CAD modela

Za dobivanje oblaka točaka iz CAD modela koristit će se MeshLab, softverski alat za obradu, analizu, filtriranje i vizualizaciju 3D modela i oblaka točaka. U MeshLab-u se prvo otvori STL datoteka površine, nakon čega se primjenjuje jedan od filtera koji stvara mrežu točaka preko površine 3D objekta. U ovom slučaju koristit će se *poisson-disk sampling* za generiranje oblaka. Ovaj rad se neće baviti načinom na koji navedena metoda radi, ali se više o njoj može pročitati u djelu „*Efficient and flexible sampling with blue noise properties of triangular meshes*“, autora Massimiliano Corsinia, Paolo Cignonia i Roberto Scopigno [16].



Slika 25. Dobivanje oblaka točkica iz CAD modela pomoću MeshLab-a

5.2. PCA za grubu registraciju

Za pronalaženje početne grube matrice transformacije između zabilježenog oblaka i oblaka točkica CAD modela koristila se PCA metoda opisana u poglavlju 3.2.3. iz *sklearn.decomposition* knjižnice za Python. Koriste se sljedeće naredbe u Pythonu:

```
# Primjena PCA na prvi ciljani oblak točkica
pca1 = PCA(n_components=3)
pca1.fit(points1)

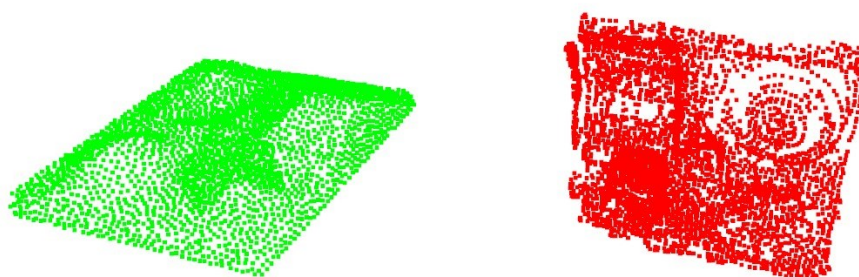
# Primjena PCA na drugi izvorni oblak točkica
pca2 = PCA(n_components=3)
pca2.fit(points2)

# Izračunavanje matrice transformacije (rotacije i translacije)
rotation = pca1.components_.T @ pca2.components_
translation = pca1.mean_ - rotation @ pca2.mean_
```

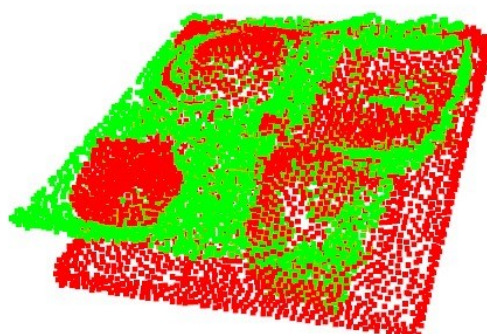
Već nakon prvih testiranja uvidjeli su se glavni nedostaci ove metode. Naime, kako je cilj izračunati svojstvene vrijednosti i vektore te nakon toga poklopiti osi najvećih smjerova rasipanja točkica, ova metoda neće dobro raditi kod objekata koji su višestruko osnosimetrični (iako u ovom slučaju testna površina nije osnosimetrična, jako je blizu tome). Razlog tomu je što se točke u približno istoj mjeri rasipaju u više različitih smjerova te je više mogućih kombinacija i orijentacija glavnih osi (svojstvenih vektora). Isto tako, područja različitih

gustoća u oblaku također će utjecati na kvalitetu PCA metode. S obzirom da je digitalizacija površine provedena ručno te je *stylusom* preko objekta negdje prelašeno brže, a negdje sporije, zabilježeni oblak će imati gušća i rjeđa područja točaka. Jedan od načina kako eliminirati problem gustoće točaka jest koristeći *voxel-grid* filtriranje opisano u točki 3.1.2.

Na slici (25) prikazan je početni raspored oblaka točaka, gdje crveni oblak predstavlja digitalizirani oblak točaka, a zeleni oblak točaka CAD modela.



Slika 26. Početni raspored oblaka točaka



Slika 27. Loše preklapanje PCA metodom

5.3. Go- ICP za preciznu registraciju

Ideja je, nakon početne grube registracije PCA metodom, koristiti ICP algoritam za preciznu registraciju. Koristimo Python knjižnicu *scikit-surgerysurfacematch* koja sadrži implementirane ICP i Go-ICP algoritme. Kako PCA metoda može dati vrlo pogrešnu početnu grubu registraciju, a koja je onda nepovoljna za klasični ICP pristup zbog problema lokalnog minimuma već objašnjenog u poglavlju 3.2.4, koristit će se globalno optimalno rješenje ICP algoritma, koje ne zahtijeva previše dobar početni raspored oblaka točaka. Iz *scikit-surgerysurfacematch* koristi se *goicp_registration* modul te se uzima klasa *Rigidregistration*.

Unutar *RigidRegistration* klase poziva se metoda *register*, koja ima nekoliko ulaznih parametara. Među njima su *rotation_limits*, koje treba promijeniti ako su orijentacije oblaka točaka u početku dosta različite i *trans_limits*, koje treba promijeniti ako su početne udaljenosti između oblaka veće.

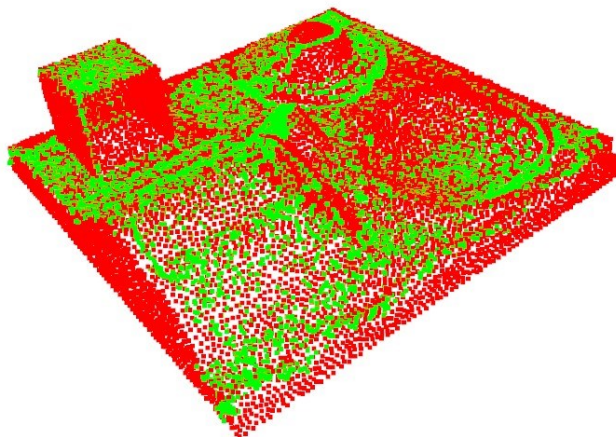
```
# Inicijalizacija registracijskog algoritma
registration = RigidRegistration()
# Izvođenje registracije
residual, transformation_matrix = registration.register(moving_cloud,
fixed_cloud)
```

Kao izlaz, algoritam daje ukupnu srednju kvadratnu grešku između točaka na početku, zatim prilikom traženja optimalnog rješenja u svakoj iteracije te krajnju pogrešku s matricom transformacije. Kako bi se ispitao algoritam registrirano je deset oblaka točaka testne pločice. Rezultati su prikazani u tablici u nastavku.

Tablica 3. Rezultati nakon Go-ICP algoritma

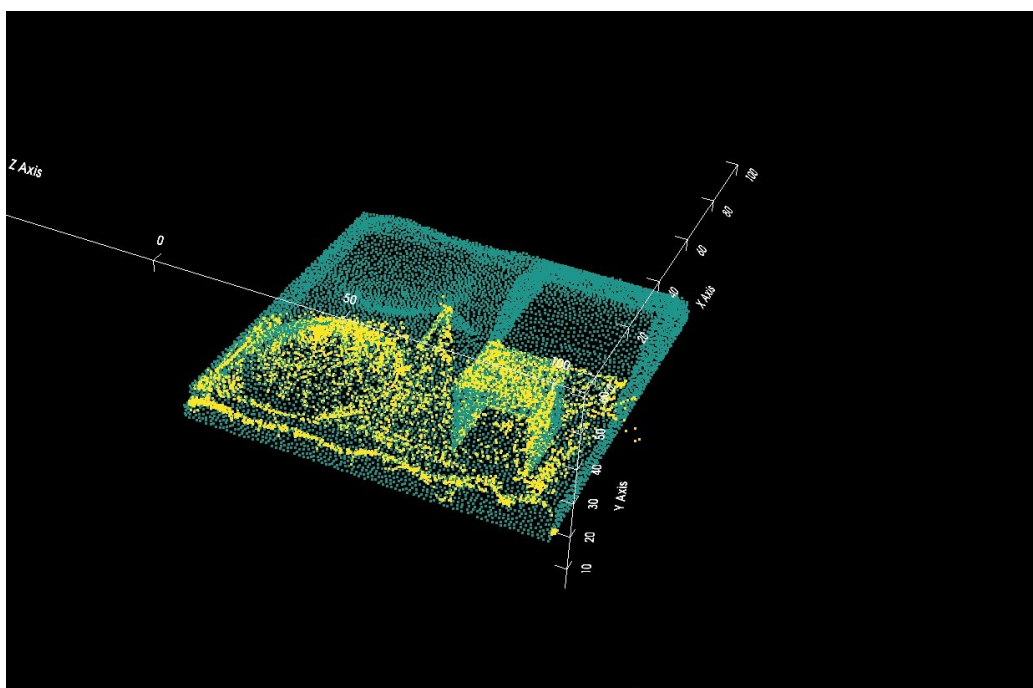
Broj točka	2738*	2448*	3041	4576	4921*	5967	8600	9094	10042	11549
RMS greška (mm)	0,33	0,44	0,35	0,19	0,27	0,18	0,23	0,27	0,19	0,20

Izračunavanjem srednje pogreške dobije se vrijednost od 0,26 milimetara. Kod oblaka koji imaju zvjezdicu (*) u tablici upotrijebljeno je *voxel-grid* filtriranje ili statističko uklanjanje šumova, jer u protivnom algoritam nije davao dobre rezultate.



Slika 28. Preklapanje nakon Go-ICP algoritma

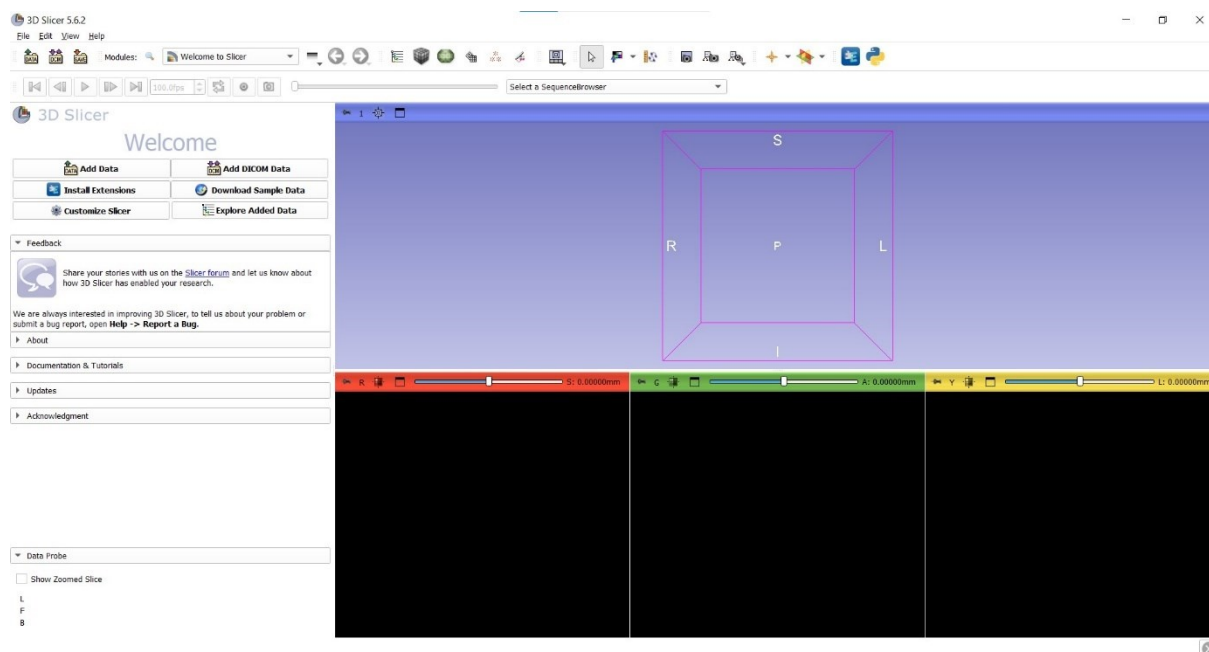
Pitanje koje se nameće jest ima li potrebe digitalizirati cijelu površinu objekta kako bi algoritam funkcionirao. U svrhu odgovaranja na ovo pitanje napravljen je oblak od oko 5000 točaka samo polovice testne površine. Konačni rezultat nije puno drugačiji. Algoritam vrlo brzo uspijeva uspješno naći optimalno rješenje uz srednju kvadratnu pogrešku od 0,25 mm, što dokazuje da nije potrebno napraviti digitalnu kopiju cijele površine te se bitno ušteduje vrijeme i skraćuje proces digitalizacije.



Slika 29. Rad algoritma pri polovičnoj digitalizaciji objekta

6. 3D SLICER

3D Slicer je besplatni softver otvorenog koda za vizualizaciju i obradu medicinskih slika. Prvenstveno je razvijen u istraživačke svrhe, ali se danas koristi u različitim područjima, kao što je medicinska dijagnostika, računalno navođene operacije, medicinska robotika, biomehanika itd. Najvažnija karakteristika ovog softvera jest proširivost; sadrži veliki broj ekstenzija koje korisnici mogu instalirati. Jedna od tih ekstenzija omogućuje povezivanje NDI stereovizijskih kamera za praćenje s 3D Slicer-om te će se iz tog razloga u ovom završnom radu koristiti upravo ovaj softver.



Slika 30. Korisničko sučelje 3D Slicer-a

6.1. PLUS

PLUS je softverski alat otvorenog koda za prikupljanje i obradu podataka i kalibraciju alata u računalno navođenim procesima. Iako je izvorno razvijen za obradu slika ultrazvuka, odakle je i dobio ime (engl. *Public software Library for UltraSound imaging research*), kroz godine je proširen i na druge tehnologije. [17]

U sklopu završnog rada koristit će se kao poveznica između NDI kamere i računala, točnije koristit će se za slanje podataka s kamere u 3D Slicer.

Za komunikaciju 3D Slicera s PLUS serverom, ali i za daljnji rad i obradu s podacima s kamere potrebno je instalirati nekoliko ekstenzija:

- *SlicerOpenIGTLink*,
- *SlicerIGT*,

- *SlicerIGSIO*,
- *SlicerJupyter*.



Slika 31. Potrebne ekstenzije unutar 3D slicer-a [18]

Kako bi PLUS mogao komunicirati s vanjskim hardverom, u ovom slučaju NDI kamerom, potrebno je napraviti PLUS konfiguracijsku datoteku. PLUS konfiguracijske datoteke su u XML formatu te ih je vrlo jednostavno uređivati običnim uređivačem teksta. U konfiguracijskoj datoteci potrebno je navesti kameru koja se koristi te listu praćenih objekata s odgovarajućim ROM datotekama:

```
<DeviceSet
  Name="PlusServer: NDI Polaris tracker with passive markers"
  Description="Broadcasting tool tracking data through OpenIGTLink
For NDI Polaris passive marker starting kit: Reference (8700339), Stylus
(8700340)" />
  <Device
    Id="TrackerDevice"
    Type="PolarisTracker"
    ToolReferenceFrame="Tracker" >
    <DataSources>
      <DataSource Type="Tool" Id="Reference"
RomFile="NdiToolDefinitions/8700339.rom"/>
      <DataSource Type="Tool" Id="Stylus"
RomFile="NdiToolDefinitions/8700340.rom" />
    </DataSources>
    <OutputChannels>
      <OutputChannel Id="TrackerStream" >
        <DataSource Id="Stylus"/>
        <DataSource Id="Reference"/>
      </OutputChannel>
    </OutputChannels>
  </Device>
```

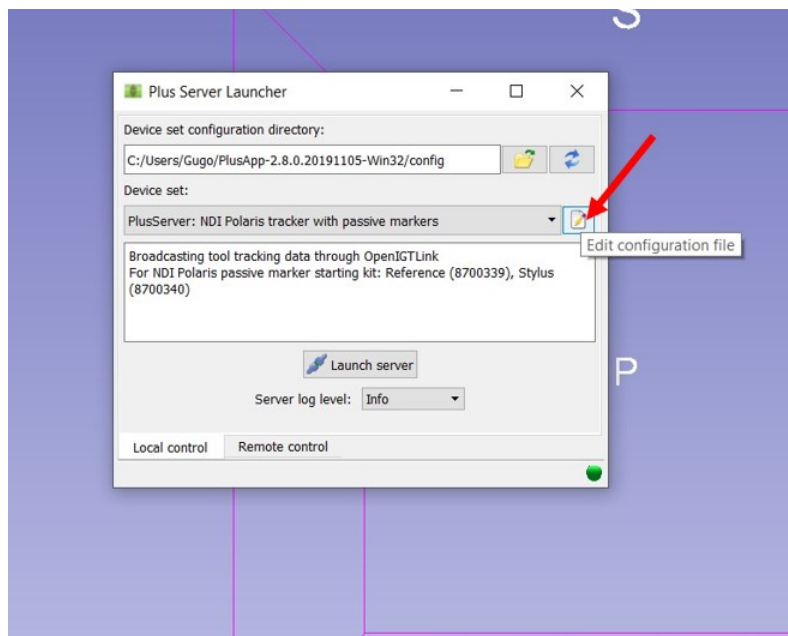
Također je potrebno definirati informacije koje će se slati 3D Slicer-u; matrice transformacija između prethodno definiranih krutih tijela s markerima:

```

<DefaultClientInfo>
  <MessageTypes>
    <Message Type="TRANSFORM" />
  </MessageTypes>
  <TransformNames>
    <Transform Name="StylusTipToStylus" />
    <Transform Name="StylusToReference" />
    <Transform Name="StylusTipToReference" />
  </TransformNames>

```

Na internetu se uglavnom mogu naći već gotove konfiguracijske datoteke svih dostupnih uređaja za medicinsko praćenje i navigaciju.

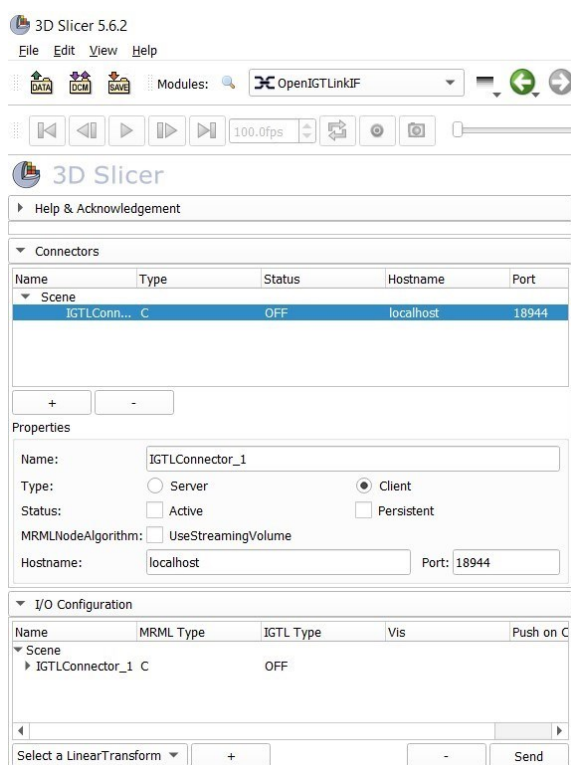


Slika 32. PLUS server launcher

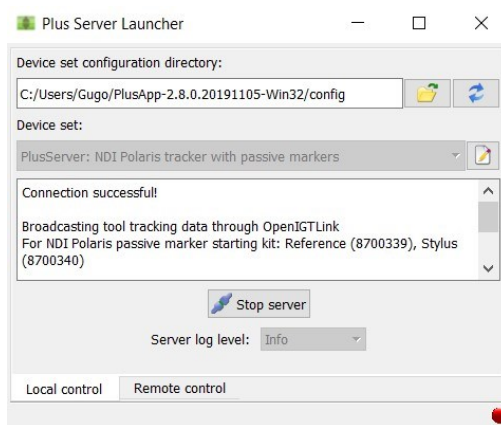
Nakon podešavanja PLUS konfiguracijske datoteke potrebno je stisnuti *Launch server* te otvoriti 3D Slicer i slijediti nekoliko jednostavnih koraka:

1. odabrati *OpenIGTLinkIF* modul,
2. dodati novi konektor pritiskom gumba +
3. stisnuti pod status *Active*.

Indikator statusa bi se trebao prebaciti na ON što znači da se Slicer uspješno povezo s PLUS serverom. Ono što bi se trebalo vidjeti pod proširenjem *Scene* konfiguracije su aktivne matrice transformacije koje su definirane da se žele dobivati od kamere u XML datoteci.



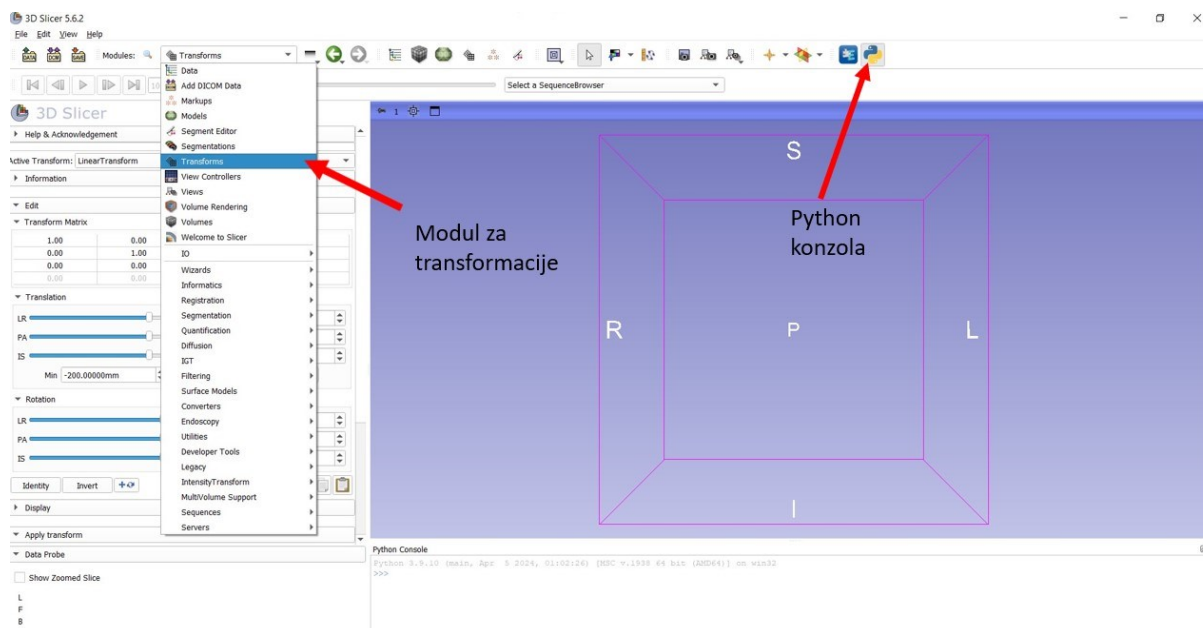
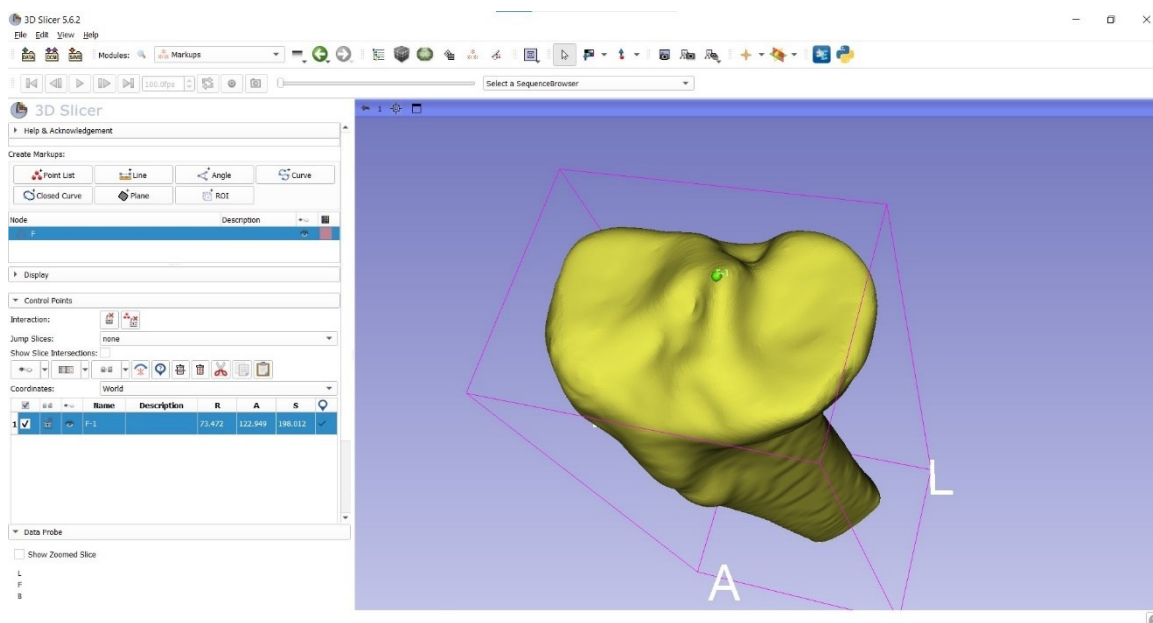
Slika 33. Izgled sučelja IGT modula



Slika 34. PLUS server nakon uspješnog povezivanja s kamerom

6.2. Moduli Transforms i Markups

Od ostalih modula u 3D Sliceru koristit će se modul *Transforms* te modul *Markups*. *Transforms* služi za poprilično jednostavno unošenje matrica transformacija. On će biti korišten za definiranje matrica transformacija potrebnih za dobre orijentacije modela *stylusa* i kosti u prostoru. Za unošenje matrica transformacija dobivenih algoritmima za registraciju koristit će se Python konzola koju 3D Slicer ima ugrađenu. U sklopu modula *Markups* moguće je vrlo precizno postavljati točke u prostoru te mjeriti udaljenosti.

Slika 35. *Transforms* modul te *Python* konzolaSlika 36. *Markups* modul

6.2.1. Orijentiranje modela *stylusa* i kosti tibije

Kako bi se osiguralo da se *stylus* u 3D Slicer-u translacija i rotacija točno kao i u stvarnosti potrebno je pomoću modula *Transforms* primijeniti matrice transformacije nad modelom. To se čini pomoću opcije *Transform hierarchy* tako da se povuče pokazivačem miša model ili

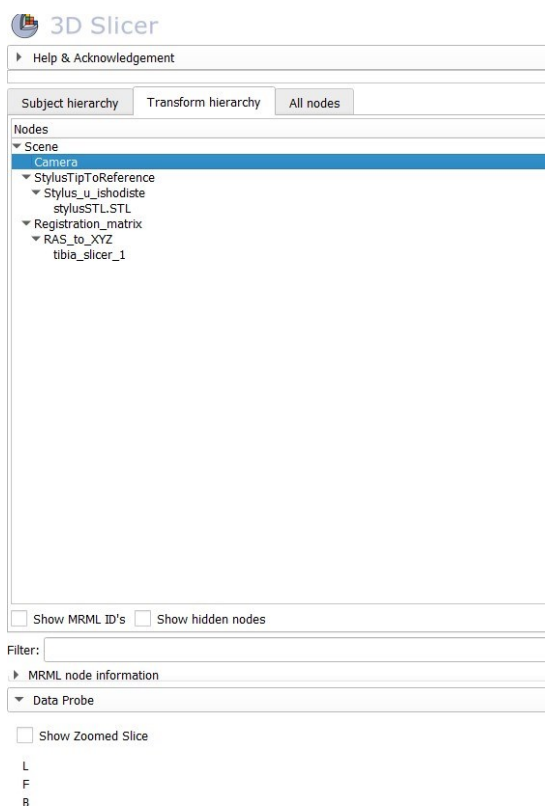
transformacija te se spusti na nadređenu transformaciju. Prvo se određuje pomak vrha igle *stylusa* u odnosu na ishodište RAS koordinatnog sustava definiranog od strane Slicer-a. To je moguće precizno učiniti pomoću modula *Markups* te funkcije *Point list*. Druga opcija je da se koristi već gotovi model igle iz IGT ekstenzije te modula *CreateModels* u kojem je moguće definirati duljinu i radijus igle. U tom slučaju 3D Slicer automatski postavlja vrh igle u ishodište, a pravilne orijentacije automatski primjenjuje nakon pivot kalibracije. Poslije pravilnog transliranja u ishodište i rotiranja modela potrebno je primijeniti *StylusTipToReference* transformaciju.

Što se tiče modela kosti, potrebno je prvo kost prebaciti u kartezijev desnokretni koordinatni sustav s obzirom na to da je 3D- Slicer u RAS koordinatnom sustavu. To činimo pomoću matrice:

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

nakon čega se primjenjuje matrica transformacija dobivena algoritmima za registraciju.

U konačnici, hijerarhija transformacija bi trebala izgledati kao na sljedećoj slici:



Slika 37. Hijerarhija transformacija

7. VALIDACIJA I PROCJENA GREŠKE

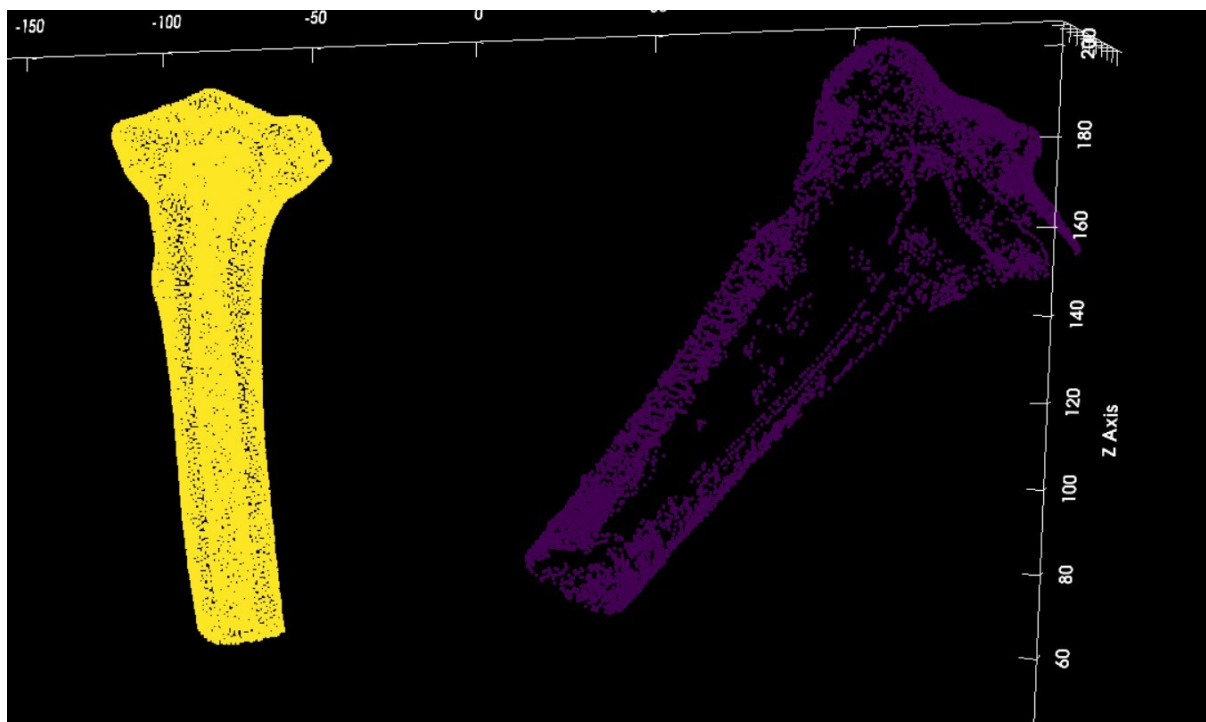
Na temelju postupaka opisanih u prethodnim poglavljima proveden je postupak digitalizacije površine 3D printane kosti tibije. Na kost je pričvršćen statični marker koji predstavlja referentni koordinatni sustav.



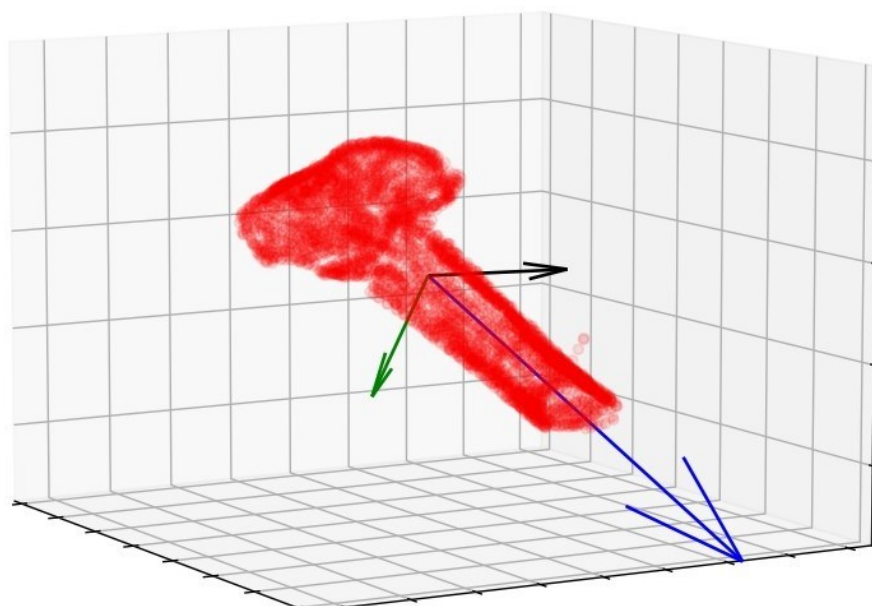
Slika 38. 3D printana kost tibije sa statičnim markerom

Napravljeno je pet oblaka točaka različitih veličina od kojih najmanji ima približno 5000 točaka, a najveći oko 20.000 točaka. Kod oblaka je prvo primijenjeno statističko uklanjanje šumova nakon čega se koristio *voxel-grid* filter za relativno jednoliku raspodjelu točaka u prostoru. Rezultati su bili iznimno zanimljivi. Naime, prilikom upotrebe samo Go-ICP algoritma dobri rezultati postignuti su u četiri slučaja, dok je u jednom, Go-ICP algoritam uspio zapeti u lokalnom minimumu. Međutim, korištenjem PCA metode za grubu registraciju u kombinaciji s Go-ICP algoritmom dobili su se dobri rezultati u svih pet slučajeva.

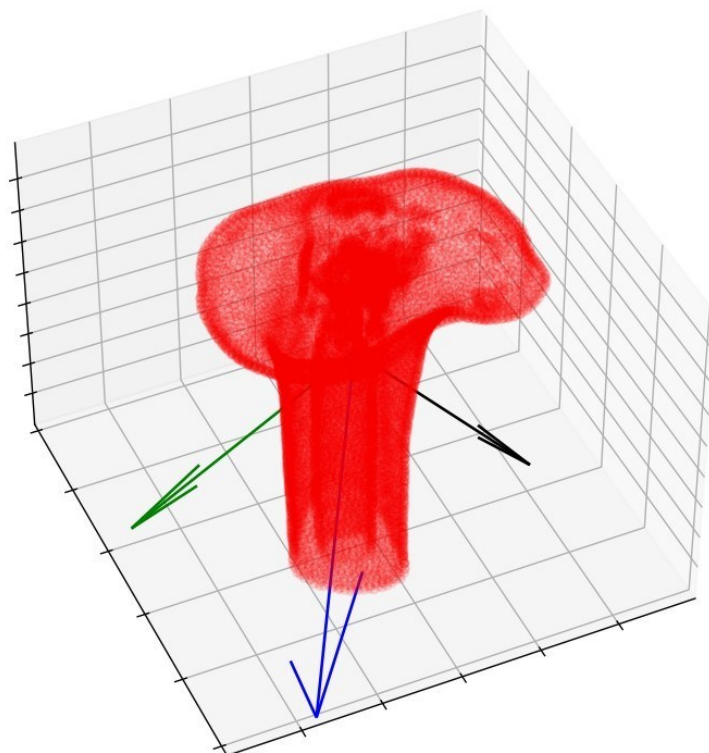
Na slici (40) prikazan je početni raspored oblaka točaka, gdje je ljubičastom bojom prikazan oblak točaka dobiven digitalizacijom površine koji sadrži oko 5000 točaka, a žutom bojom oblak točaka CAD modela dobiven u MeshLab-u. Za vizualizaciju svojstvenih vektora u pythonu je korištena *matplotlib* knjižnica točnije modul *pyplot*.



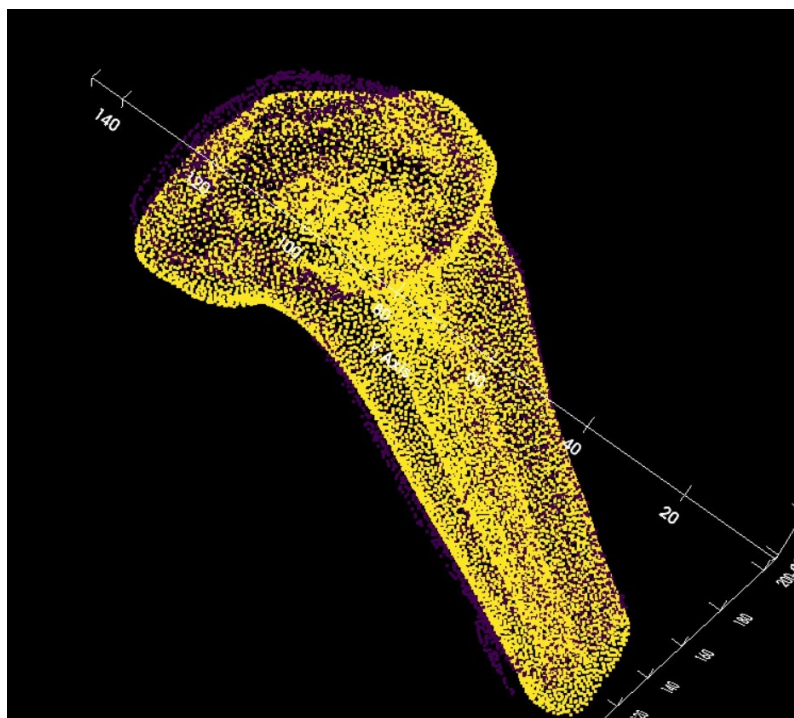
Slika 39. Početni raspored oblaka točaka kosti tibije



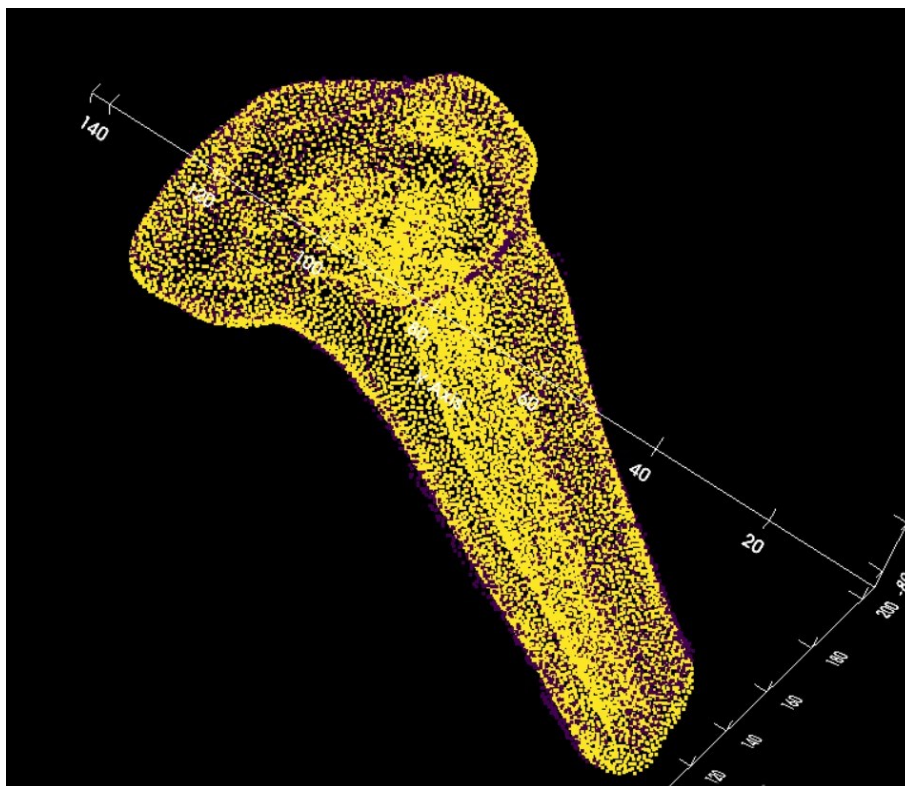
Slika 40. Svojstveni vektori digitaliziranog oblaka točaka (plavom - svojstveni vektor najvećeg smjera rasipanja, zelenom - vektor drugog najvećeg smjera rasipanja, crnom - vektor najmanjeg smjera rasipanja)



Slika 41. Svojstveni vektori oblaka točaka CAD modela (plavom - svojstveni vektor najvećeg smjera rasipanja, zelenom - vektor drugog najvećeg smjera rasipanja, crnom - vektor najmanjeg smjera rasipanja)



Slika 42. Oblaci točaka nakon PCA registracije



Slika 43. Oblaci točkaka nakon Go-ICP registracije

U svih pet slučajeva Go- ICP algoritam je generirao RMS grešku manju od 0,25 mm sa vremenom izvršavanja algoritma od prosječno 4,5 sekundi, dok je najmanja greška iznosila 0,096 mm i to kod oblaka sa najvećim brojem točkaka, što sugerira da bi veličina oblaka točkaka mogla biti jedan od faktora preciznosti samog procesa.

```
Error*: 0.0963387, LB: 0, epsilon: 1
CLEARING THE GOICP SYSTEM ::
Transformation matrix:
[[ 9.79934579e-01  1.49454017e-01 -1.31879902e-01  4.71696403e+00]
 [-1.50811444e-01  9.88562540e-01 -3.10346721e-04 -3.51944254e+00]
 [ 1.30324964e-01  2.01930308e-02  9.91265699e-01  7.55857706e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

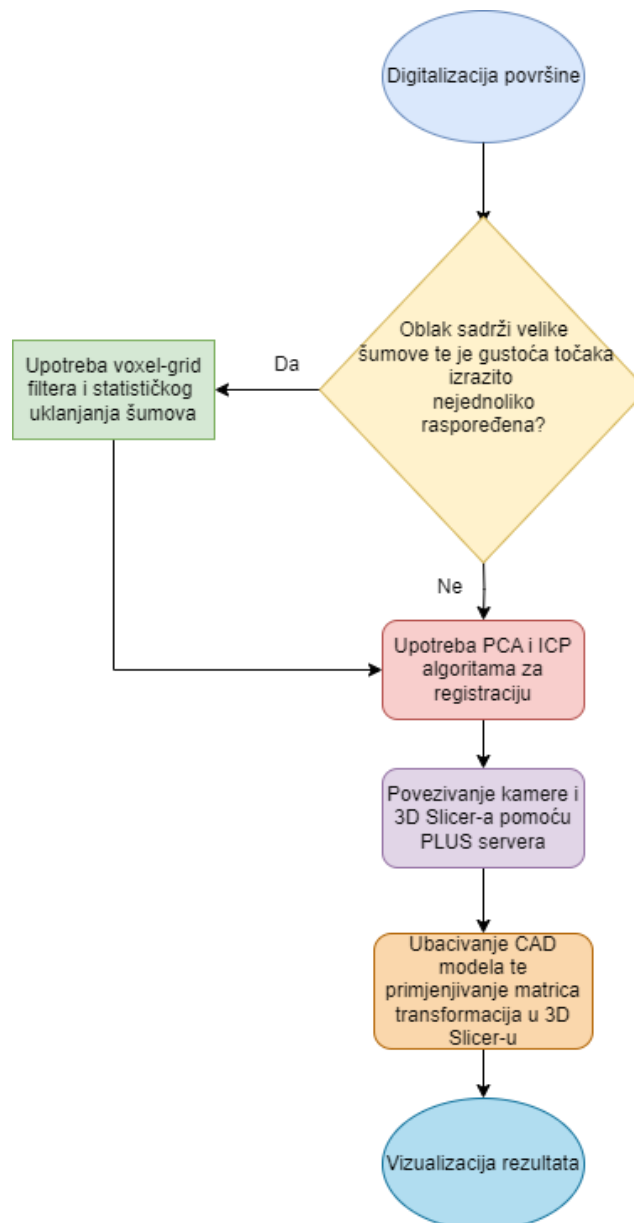
Slika 44. Rezultati Go-ICP algoritma kod oblaka s najvećim brojem točkaka

Ukupna matrica transformacije dobivena registracijskim algoritmima tada se računa jednostavnim matričnim množenjem:

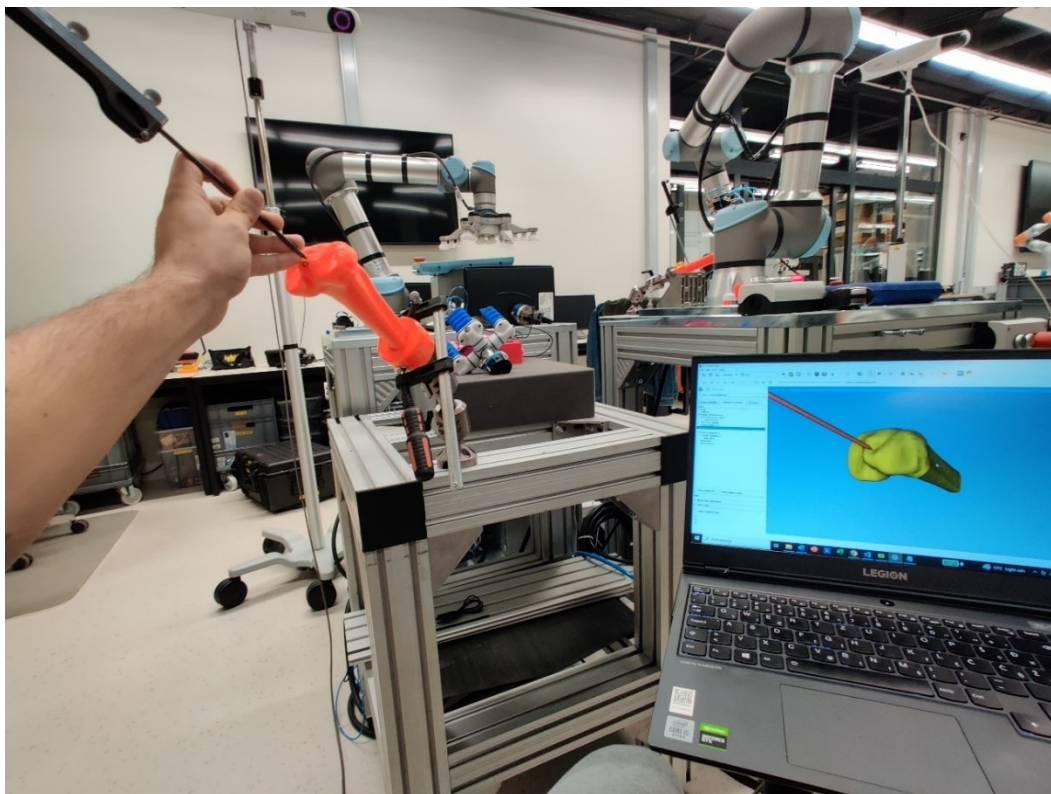
$$T_{registration} = T_{Go-ICP} T_{PCA}$$

Ubacivanjem ukupne matrice transformacija u 3D Slicer zajedno sa prethodno definiranim dobije se konačni produkt CAD modela kosti u prostoru slike.

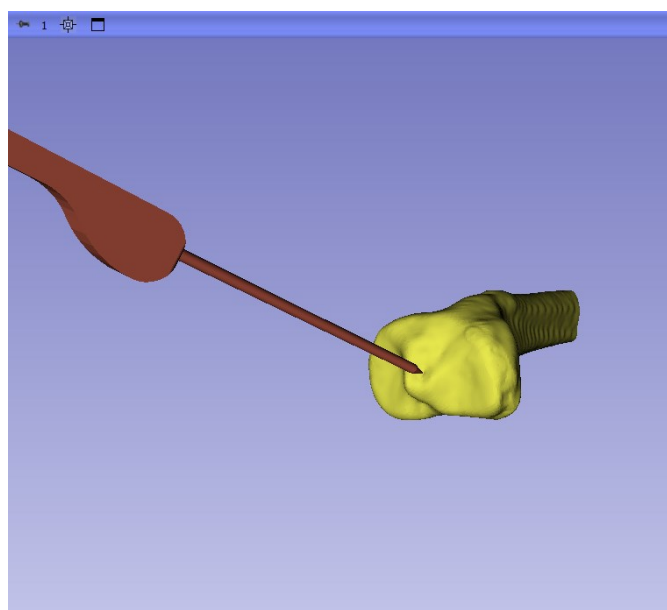
Sveukupan postupak lokalizacije i registracije te daljni koraci dani su pomoću sljedećeg dijagrama:



Slika 45. Dijagram toka

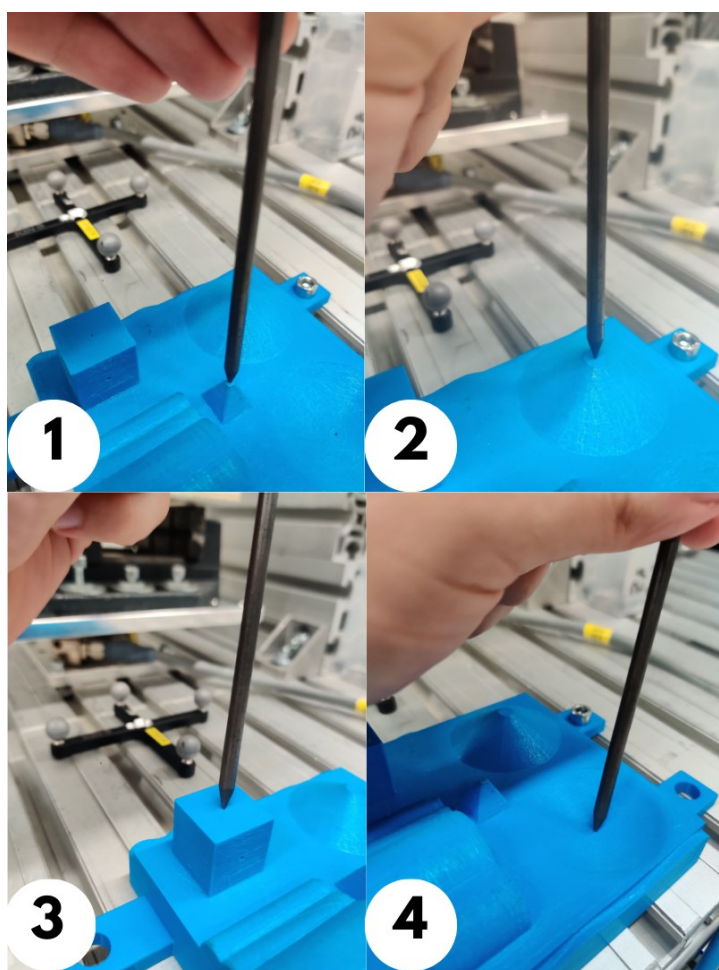


Slika 46. Konačan rezultat

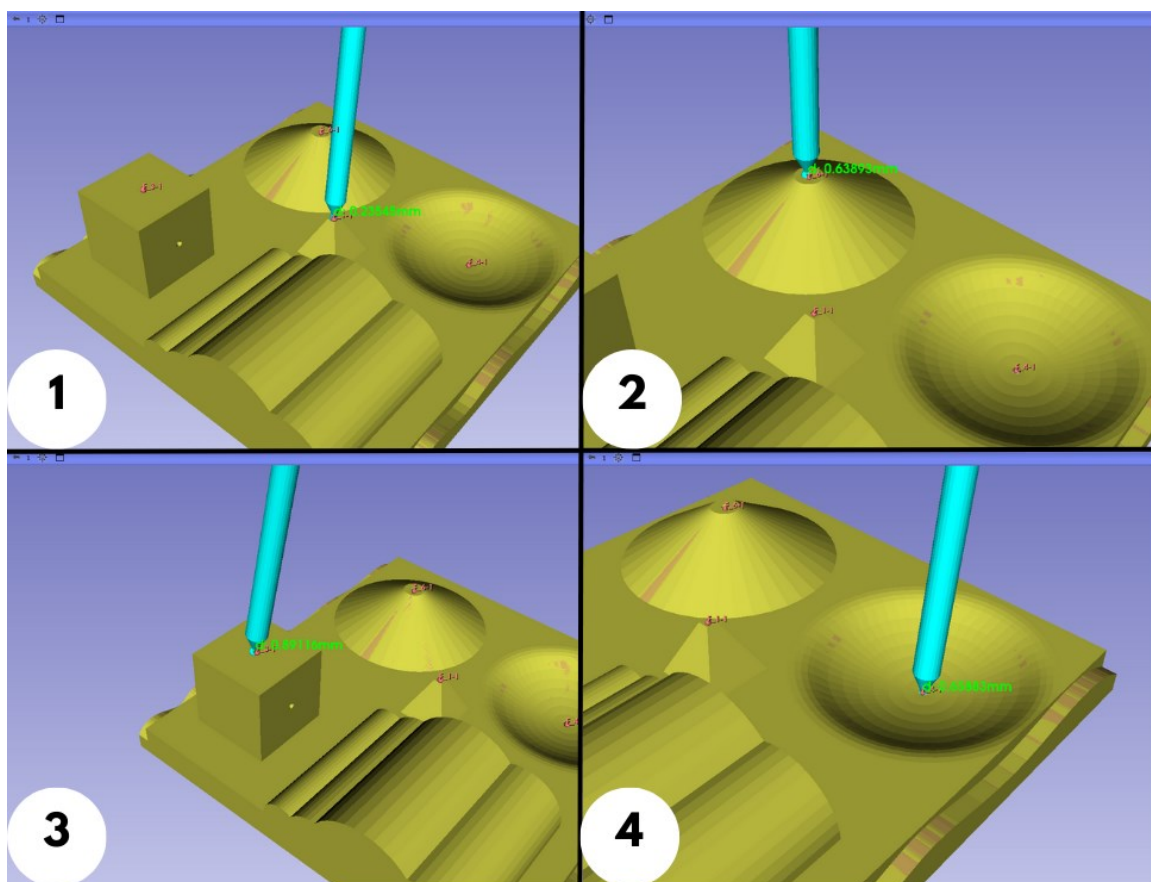


Slika 47. 3D pogled u Slicer-u

Kako bi se ispitala točnost cijelog postupka korištena je već spomenuta testna pločica na čijoj su površini raspoređeni konični utori promjera 1 mm. Ispitivanje se provelo uz pomoć *BreachWarning* ekstenzije unutar Slicer-a koja omogućuje praćenje udaljenosti između definirane točke u prostoru te vrha stylusa u stvarnom vremenu. Najprije je registriran oblak točaka te je nađena matrica transformacije. Nakon namještanja radnog okruženja unutar Slicer-a, definirale su se četiri točke na površini modela, raspoređene u koničnim utorima. Stylus je doveden u sve četiri točke i položen uspravno uz minimalne nagibe te je promatrana vrijednost udaljenosti između definirane točke te vrha stylusa.



Slika 48. Dovođenje stylusa u definirane točke



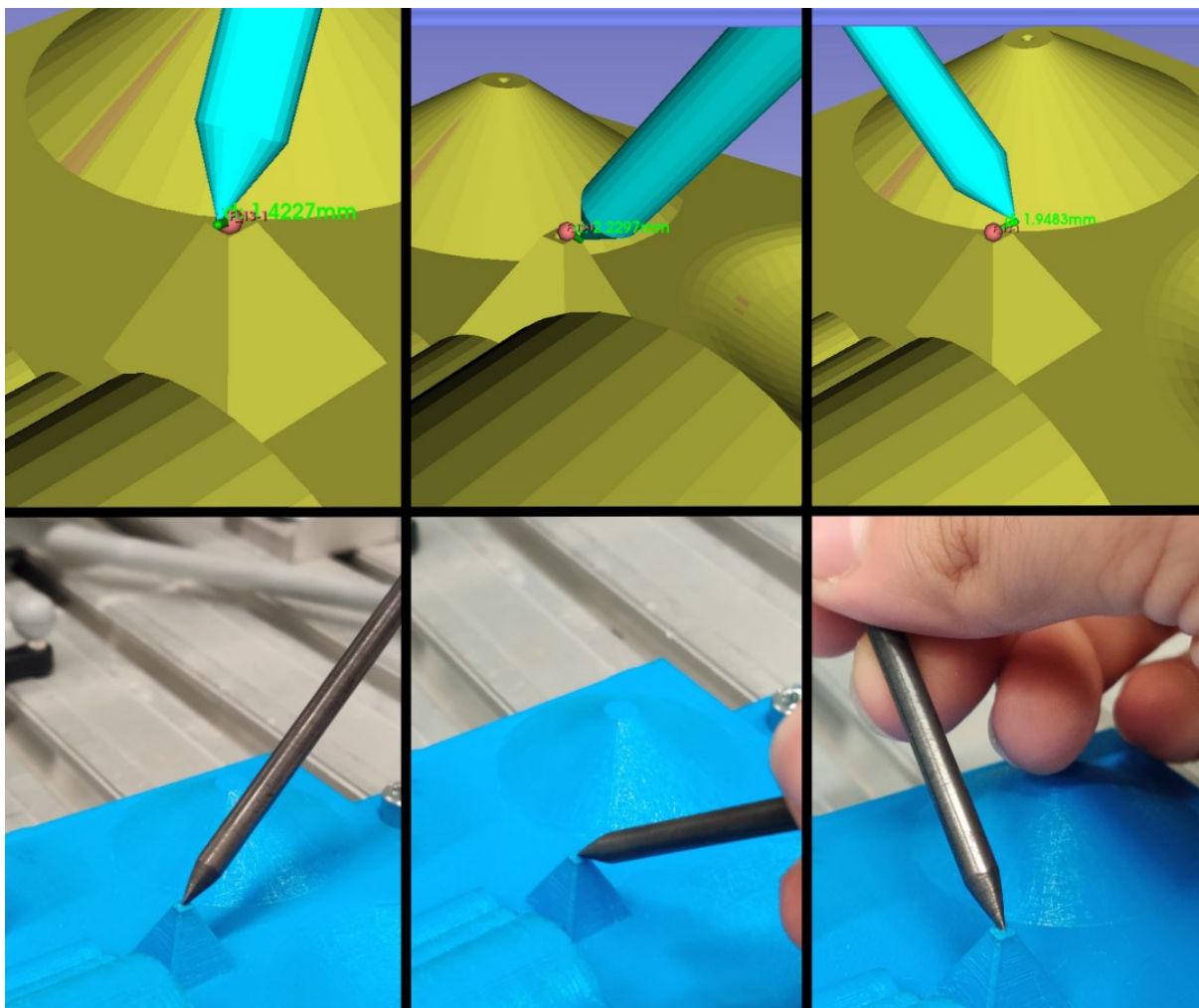
Slika 49. Prikaz mjerenja u 3D Slicer-u

Ispitivanje je provedeno na pet registriranih oblaka točaka, a dobiveni rezultati su u tablici u nastavku.

Tablica 4. Rezultati mjerenja (mm)

	Mjerenje 1	Mjerenje 2	Mjerenje 3	Mjerenje 4	Mjerenje 5
Točka 1	0,24	1,11	0,90	0,75	0,82
Točka 2	0,66	0,29	0,71	0,54	0,78
Točka 3	0,64	0,71	0,85	1,18	1,29
Točka 4	0,90	0,65	0,58	1,01	0,48
Srednja greška	0,61	0,69	0,76	1,02	0,84

Izračunavanjem ukupne prosječne greške dobije se vrijednost od 0,784 milimetara. Treba opet naglasiti da su ovo statička mjerenja kod uspravno položenog markera. Kod većih nagiba stylusa situacija se znatno mijenja te dolazi do povećanja greške. Ovo povećanje pogreške doseže vrijednosti i do dva milimetra.



Slika 50. Greška kod nagiba stylusa

Najveća odstupanja vrha od definiranih točaka primijećena su kod velikih zakreta stylusa oko aksijalne osi. Naime, greška tada izrazito raste (na vrijednosti preko 3 mm) iako je provedena pivot kalibracija alata nekoliko puta. Razlog ovolikoj grešci može ležati u činjenici da je duljina alata izrazito velika pa dolazi do pogrešnih kutnih očitavanja optičke kamere opisane u točki 4.2.1. Daljnjim istraživanjima bi se mogao naći pravi razlog velike mjere rasipanja vrha stylusa prilikom vrtnje oko vlastite osi.

8. ZAKLJUČAK

Cilj ovog završnog rada bio je upoznati se s optičkim sustavima za praćenje te algoritmima i tehnologijama za obradu i preklapanje oblaka točaka, ali i sa samim principom funkcioniranja računalno navođenih operacija.

Digitalizacijom površine, ručno navođenim alatom, kreiran je oblak točaka kosti tibije te je algoritmima za krutu registraciju uspješno pronađena matrica transformacije, prvo PCA metodom u svrhu poboljšanja početnih orijentacija oblaka te izbjegavanja problema lokalnog minimuma, a onda Go-ICP algoritmom za precizno preklapanje točaka. Ponovljivost algoritma pokazana je na desetak oblaka točaka. Nadalje, ostvarena je komunikacija između NDI *Polaris Spectra* stereovizijske kamere s 3D Slicer-om putem PLUS servera. Validacija je napravljena računanjem greške između točke u prostoru slike i vrha stylusa uz pomoć 3D Slicer ekstenzije te je utvrđena greška od 0,784 milimetra, a pri velikim rotacijama i nagnjanjem alata greška raste i do nekoliko milimetara.

Zaključno, ovaj završni rad uspješno je demonstrirao implementaciju algoritama za preklapanje oblaka točaka u procesu registracije i lokalizacije. S druge strane ukazao je na izazove u preciznosti praćenja optičkih kamera te kalibracije alata. Rezultati dokazuju funkcionalnost te obećavajuću budućnost primjene tehnologije oblaka točaka u računalno navođenim operacijama, ali istovremeno ističu potrebu za daljnjim istraživanjem i optimizacijom kako bi se minimizirale greške u složenijim operativnim uvjetima.

LITERATURA

- [1] <https://mphy0026.readthedocs.io/en/latest/tracking/optical.html> Pristup: 13.9.2024.
- [2] <https://www.ndigital.com/optical-navigation-technology/navigation-markers/>, Pristup: 11.9.2024.
- [3] <https://www.semanticscholar.org/paper/Comparing-Measured-and-Theoretical-Target-Error-of-Banivaheb/7d110dba05626253b6a3dc3b7249e6c001ddbce0>, Pristup: 16.9.2024.
- [4] <https://compumedicsneuroscan.com/product/ndi-spectra/> Pristup: 17.9.2024.
- [5] <https://www.geospatialworld.net/> Pristup: 17.9.2024.
- [6] <https://www.open3d.org/docs/release/tutorial/geometry/> Pristup: 12.9.2024.
- [7] <https://www.vassar.edu/news/leaving-a-trace> Pristup: 11.9.2024.
- [8] <https://www.semanticscholar.org/paper/NDT-scan-matching-method-for-high-resolution-grid-Takubo-Kaminade/324fa0f0ffaa8ba865dee9d13a859de5eea13ce1>, Pristup: 14.9.2024.
- [9] Šuligoj F.: Spatial patient registration in medical robotics, predavanja Medicinska robotika, Fakultet strojarstva i brodogradnje u Zagrebu, 2024.
- [10] Besl PJ, McKay ND, „A Method for Registration of 3-D Shapes“, 1992.
- [11] Yang J., Hongdong L., Jia J. : „Go-ICP: Solving 3D Registration Efficiently and Globally Optimally, 2013.
- [12] Bellekens B., Spruyt V., Berkvens B. i Weyn M.: „A Survey of Rigid 3D Pointcloud Registration Algorithms“, stranice 9-13, 2014.
- [13] <https://www.geeksforgeeks.org/branch-and-bound-algorithm/> Pristup: 14.9.2024.
- [14] Yaniv Z.: „Which pivot calibration?“, 2017.
- [15] Guofang X., Bonmati E., Thompson S.: “Electromagnetic tracking in image-guided laparoscopic surgery: Comparison with optical tracking and feasibility study of a combined laparoscope and laparoscopic ultrasound system”, 2018.
- [16] Cignoni P., Corsini M., Scopigno R.: „Efficient and Flexible Sampling with Blue Noise Properties of Triangular Meshes“, 2012.
- [17] <https://plustoolkit.github.io/features> Pristup: 17.9.2024.
- [18] <https://www.slicerigt.org/wp/user-tutorials/> Pristup: 17.9.2024.

PRILOZI

Github link: https://github.com/Gugo1507/Markerbased_registration_and_localization