

# Aktivna rekonstrukcija 3D modela objekta primjenom robotske ruke

---

Krivošija, Tarik

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:152813>

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

Download date / Datum preuzimanja: **2025-03-15**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 89

**AKTIVNA REKONSTRUKCIJA 3D MODELA OBJEKTA  
PRIMJENOM ROBOTSKE RUKE**

Tarik Krivošija

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 89

**AKTIVNA REKONSTRUKCIJA 3D MODELA OBJEKTA  
PRIMJENOM ROBOTSKE RUKE**

Tarik Krivošija

Zagreb, lipanj 2024.

## DIPLOMSKI ZADATAK br. 89

Pristupnik: **Tarik Krivošija (0036554557)**  
Studij: Informacijska i komunikacijska tehnologija  
Profil: Automatika i robotika  
Mentorica: izv. prof. dr. sc. Tamara Petrović

Zadatak: **Aktivna rekonstrukcija 3D modela objekta primjenom robotske ruke**

### Opis zadatka:

Radom na diplomskom zadatku potrebno je implementirati strategiju upravljanja robotskom rukom radi autonomne rekonstrukcije 3D modela objekta koji se nalazi u okolini robotske ruke. Strategija je algoritam koji odlučuje koji je najbolji idući pogled robotske ruke i upravlja pokretima robotske ruke kako bi se istražili svi dijelovi objekta te stvorio što detaljniji 3D model, uz izbjegavanje kolizija. Koristeći gotov model Kinova robotske ruke izgraditi prikladan simulacijski svijet za autonomnu rekonstrukciju objekta koji treba testirati na nekoliko različitih objekata. S pomoću OctoMap biblioteke vizualizirati rekonstruirani model. Metodu je potrebno provjeriti u simulacijskom okruženju te usporediti s drugim pristupima u literaturi.

Rok za predaju rada: 28. lipnja 2024.

## Sadržaj

1. Uvod .....	1
2. Korištene tehnologije.....	3
2.1. Kinova Gen 3 robotska ruka .....	3
2.2. Kortex API.....	4
2.3. Octomap biblioteka.....	4
3. Volumetrijska informacija .....	5
3.1. Volumetrijska informacija .....	5
3.2. Nesigurnost mape .....	5
3.3. Volumetrijska informacija temeljena na okluziji .....	6
3.4. Volumetrijska informacija temeljena na nepoznatim voksela .....	6
3.5. Volumetrijska informacija temeljena na pozadinskim voksela.....	7
3.6. Volumetrijska informacija temeljena na entropiji pozadinskih vokseli .....	7
3.7. Volumetrijska informacija temeljena na broju bliskih vokseli .....	7
3.8. Grafički prikaz različitih volumetrijskih informacija .....	8
4. Metoda za rekonstrukciju 3D objekta temeljena na volumetrijskoj informaciji .....	10
4.1. Ažuriranje 3D mape na temelju mjerenja LiDaR-a.....	11
4.2. Odabir idućeg pogleda na temelju procjene volumetrijske informacije .....	12
4.3. Promjena položaja robota .....	15
5. Simulacija .....	16
5.1. Cijev.....	16
5.2. Igračka .....	23
5.3. Čajnik.....	27
6. Zaključak .....	31
Literatura .....	32
Sažetak.....	33
Summary.....	34



# 1. Uvod

3D rekonstrukcija objekta važan je korak u zadacima robusne percepcije i manipulacije. Da bi rekonstruirao objekt, robot mora pozicionirati senzor na različite točke gledišta kako bi dobio sliku cijelog objekta. Detaljno promatranje objekta traje dugo, pa je ključno za svaki naredni korak odabrati točku gledišta koja će pružiti najviše informacija u svrhu učinkovitog obavljanja zadatka [1]. Točka gledišta koja će pružiti najviše informacija u idućem koraku se naziva „sljedeći najbolji pogled” (engl. Next-Best-View, NBV), a robot ju mora sam birati na osnovu informacija koje ima u tom trenutku.

U literaturi je istraženo nekoliko metoda za rekonstrukciju 3D modela objekta na temelju NBV-a. Ove metode se mogu kategorizirati u dva glavna pristupa: metode temeljene na modelu objekta (engl. model-based) te metode koje nisu temeljene na modelu (engl. non-model-based). Metode temeljene na modelu objekta polaze od pretpostavke da je barem približan model scene poznat unaprijed, dok druge metode rade s minimalnim pretpostavkama o strukturi objekta, što omogućava prilagodljivost u različitim scenarijima, ali zahtjeva da se potrebne informacije procjenjuju tijekom rada sustava na osnovu prikupljenih podataka.

Jedan od izazova u 3D rekonstrukciji je upravljanje velikim količinama podataka dobivenih sa senzora. Algoritmi za kompresiju podataka i optimizaciju mapiranja igraju ključnu ulogu u učinkovitosti procesa rekonstrukcije. Na primjer, algoritmi temeljeni na Octree [2] strukturama omogućuju učinkovito skladištenje i pristup velikim količinama trodimenzionalnih podataka.

Ovaj rad posebno razmatra problem 3D rekonstrukcije objekta ili scene koja je a priori nepoznata, ali koja je prostorno omeđena. Metoda koja je implementirana opisana je u [1]. Koristi se Kinova Gen 3 robotska ruka koja na mjestu alata ima postavljen LiDaR senzor. Radi se o senzoru dometa 1 metar, kuta detekcije od jednog radijana i rezolucije 0.01 metar. Za upravljanje rukom koristi se `ros_kortex`, službeni ROS paket za interakciju s Kortexom i njegovim srodnim proizvodima [3]. Za rekonstrukciju objekta koristi se Octomap, biblioteka koja omogućava 3D mapiranje mreže zauzetosti, implementirajući strukture podataka i algoritme mapiranja u C++ [4].

U idućem poglavlju dan je opis svih korištenih hardverskih i softverskih rješenja, te su navedene i osnove specifikacije. U trećem poglavlju su predstavljeni algoritmi na osnovu kojih će robot obavljati zadatke. Za svaki algoritam prikazano kako se računa nagrada na osnovu koje će robot znati svoj idući pokret. Četvrto poglavlje donosi strukturu i opis rada cijelog sustava. Detaljno se opisuje svaki dio sustava uključujući strukturu robotske ruke, strukturu razmijenjenih poruka, programsku i matematičku podlogu. U petom prikazani su rezultati simulacije za različite objekte, te je provedena usporedba rada algoritma za različite ulazne parametre. Na kraju rada prikazan je zaključak.



## 2. Korištene tehnologije

U ovom poglavlju opisana je robotska ruka Kinova Gen3 koja se koristi za 3D rekonstrukciju, kao i programski paketi nužni za implementaciju algoritma za rekonstrukciju.

### 2.1. Kinova Gen 3 robotska ruka

Kinova Gen 3 robotska ruka, poznata i kao Kinova Gen3 Ultra Lightweight Robot [5], predstavlja napredni model robotske ruke razvijen od strane kanadske kompanije Kinova Robotics. Ova ruka dio je serije robota dizajniranih za različite primjene, uključujući istraživanje, medicinske aplikacije, industrijsku automatizaciju i asistivne tehnologije. Ruka ima 7 stupnjeva slobode, domet joj je oko 90 cm kada je potpuno rastegnuta, a maksimalna brzina joj je 50cm/s što su, za potrebe primjene iz ovog rada, zadovoljavajuće specifikacije.



Slika 1. Kinova gen 3 [5]

Gen3 se lako integrira s drugim tehnologijama pomoću Kinova Kortex™ API softvera. Moguće je učitati programski kod na robotsku ruku pomoću sučelja Matlab Simulink, implementirati mobilnu manipulaciju pomoću ROS upravljačkih programa ili razviti potpuno prilagođenu aplikaciju koristeći Kortex API dostupan u Python i C++. [5]

## 2.2. Kortex API

ROS\_Kortex je softverski paket razvijen za integraciju i upravljanje Kinova robotskih ruku, unutar operativnog sustava za robote (engl. Robot Operating System, ROS). ROS\_Kortex omogućava korisnicima da lako povežu Kinova ruke s ROS-om, što olakšava razvoj aplikacija, simulacija i upravljačkih algoritama za različite primjene. Paket pruža API koji omogućava kontrolu svih aspekata Kinova robotske ruke, uključujući kinematiku, dinamiku, senzore i aktuatore.

Ovaj paket sadrži URDF (engl. Unified Robot Description Format) i konfiguracijske datoteke za Kortex-kompatibilne robote. Sadrži i automatski generirane MoveIt! datoteke za korištenje Kinova Gen3 i Gen3 lite ruku sa MoveIt! framework-om za planiranje pokreta. Paket također podržava simulacije u Gazebo simulatoru i ima kreirane primjere napisane u c++ ili Pythonu. [3]

## 2.3. Octomap biblioteka

OctoMap [4] je biblioteka za modeliranje okruženja u 3D prostoru koja koristi podatke prikupljene iz senzora kao što su LIDAR i RGB-D kamere. Biblioteka je dizajnirana da bude efikasna, fleksibilna i jednostavna za upotrebu u aplikacijama koje zahtjevaju prostorno mapiranje, navigaciju i percepciju u stvarnom vremenu. Neke od odlika su:

- Karta može modelirati proizvoljna okruženja bez prethodnih pretpostavki o njima. Nepoznata područja okoline su implicitno kodirana u karti. Dok je razlika između slobodnog i zauzetog prostora ključna za sigurnu navigaciju robota, informacije o nepoznatim područjima važne su, npr. za autonomno istraživanje okoliša.
- U svakom trenutku moguće je dodati nove informacije ili očitavanja senzora.
- Opseg karte ne mora biti poznat unaprijed. Umjesto toga, karta se po potrebi dinamički proširuje.

### 3. Volumetrijska informacija

U ovom poglavlju opisat će se teorijska osnova algoritma za 3D rekonstrukciju objekta koji je implementiran u ovom radu, a koji se temelji na članku [1]. Algoritam u svakom koraku odabire idući najbolji pogled za robotsku ruku. Taj pogled određuje se na temelju procjene volumetrijske informacije (engl. volumetric information, VI) tj. količine informacija sadržanih u vokselima karte. U nastavku su prikazani različiti načini za računanje volumetrijske informacije.

#### 3.1. Volumetrijska informacija

Volumetrijska informacija (VI) definira se kao količina informacija koja se očekuje od pomaka robotske ruke sa senzorom u voksel za koji se VI računa. Sljedeći najbolji pogled (NBV) je prikaz koji maksimizira ovu metriku, uzimajući u obzir sve procijenjene troškove. U literaturi se volumetrijska informacija nekad naziva i informacijska dobit (engl. information gain).

Za svaki pogled unutar skupa mogućih položaja senzora  $V$ , 3D točke sa senzora udaljenosti su projicirane u kartu. Projekcija se provodi kroz generiranje zraka za svaki pogled (engl. Ray Casting). Zraka završava kada dođe na površinu ili kada dosegne granicu karte. Predviđeni VI za pogled  $v \in V$ , označen kao  $G_v$ , je kumulativna volumetrijska informacija koja se dobiva za taj pogled iz svih generiranih zraka  $r$  koje izlaze iz  $v$ :

$$G_v = \sum_{R_v} \sum X_i, \forall r \in R_v, \forall x \in X_i \quad (1)$$

gdje je  $R_v$  skup svih zraka iz svakog pogleda, a  $X_i$  skup svih vokseli koje zrake dosegnu. Formulacija VI u jednadžbi (1) i skup pogleda koji se ocjenjuje definiraju ponašanje sustava.

#### 3.2. Nesigurnost mape

Nesigurnost volumetrijske karte može se definirati korištenjem entropije vokseli:

$$H(x) = -P_o(x) \ln P_o(x) - \overline{P_o}(x) \ln \overline{P_o}(x) \quad (2)$$

gdje  $P_o(x)$  označava vjerojatnost okupiranosti vokseli  $x$  dok je  $\overline{P_o}(x)$   $P_o$ , tj.  $P_o = 1 - \overline{P_o}$ . Voksel koji nije istražen ima popunjenost ( $P_o = 0.5$ ), a s time i najveću nesigurnost ( $P_o(x) = 0.5$  i  $H(x) = 1$ ). Ako se uzme u obzir da pogled koji promatra područje karte na kojem je visoka nesigurnost vjerojatno daje više informacija, entropija se može koristiti

kao metrika za maksimiziranje ukupnog iznosa novih informacija prikupljenih u svakoj iteraciji rekonstrukcije. Stoga se definira VI na temelju entropije:

$$I_e(x) = H(x) \quad (3)$$

### 3.3. Volumetrijska informacija temeljena na okluziji

Volumetrijska informacija temeljena na okluziji (engl. Occlusion Aware VI) je formulacija koja računa očekivanu nagradu entropije svakog vokselu i vjerojatnosti vidljivosti. Ova definicija uzima u obzir činjenicu da će se vjerojatnost uočavanja vokselu smanjivati kako se prijeđe više nepoznatih vokselu, što prirodno rješava problem okluzije. Vjerojatnost vidljivosti  $P_v$  vokselu  $x_n$  dana je sa:

$$P_v(x_n) = \prod_{i=1}^{n-1} P_o(x_i) \quad (4)$$

gdje su  $x_i$ ,  $i = 1, \dots, n - 1$  svi vokseli prijedeni zrakom prije nego što dosegne voksel  $x_n$ . Koristeći jednadžbu (4) može se definirati:

$$I_v(x) = P_v(x)H(x) \quad (5)$$

Iz ove formule se može zaključiti da će robot favorizirati one voksele za koje se utvrdi da je velika vjerojatnost da će biti vidljivi. Također, kao što je rečeno, najveću entropiju imaju oni vokseli s najvećom nesigurnošću, odnosno oni koji nisu istraženi pa je suma ove dvije funkcije nagrada koja se može očekivati koristeći ovu metodu.

### 3.4. Volumetrijska informacija temeljena na nepoznatim vokselima

Volumetrijska informacija temeljena na nepoznatim vokselima (engl. Unobserved Voxel VI) ograničava skup vokselu koji doprinose novim informacijama na voksele koji još nisu promatrani. Ova formulacija daje prednost pogledima koji otkrivaju nepoznate dijelove objekta, čime se povećava pokrivenost površine objekta. Vrlo je slična prethodnoj uz razliku što je sada nagrada:

$$I_k(x) = I_u(x)I_v(x) \quad (6)$$

gdje je:

$$I_u(x) = 1, x \text{ nije istražen}$$

$I_u(x) = 0$ ,  $x$  je istraženo

### 3.5. Volumetrijska informacija temeljena na pozadinskim vokselima

Volumetrijska informacija temeljena na pozadinskim vokselima (engl. Rear Side Voxel VI) je metoda koja broji voksele za koje se očekuje da će biti vidljivi sa stražnje strane okupiranih vokseli. Najjednostavniji prikaz za ovo je indikatorska funkcija, koja određuje je li voksel dio područja interesa, u binarnom obliku:

$$I_b(x) = \{1, x \in S_0; 0, x \notin S_0\} \quad (7)$$

gdje je  $S_0$  skup pozadinskih neopaženih vokseli, takvih da se sljedeći voksel na njihovoj zruci procjenjuje da će biti zauzet. Otkrivanje svakog pozadinskog vokseli uvećava nagradu za 1.

### 3.6. Volumetrijska informacija temeljena na entropiji pozadinskih vokseli

Volumetrijska informacija temeljena na entropiji pozadinskih vokseli (engl. Rear Side Entropy VI) slična je prethodnoj metodi uz razliku što se, kao kod volumetrijske informacije temeljene na okluziji, nagrada uvećava za umnožak entropije i vjerojatnosti vidljivosti pozadinskog vokseli. Iz jednadžbe (5) dobiva se:

$$U(x) = I_b(x)I_v(x) \quad (8)$$

### 3.7. Volumetrijska informacija temeljena na broju bliskih vokseli

Problem sa prethodne dvije formulacije je što se pretpostavlja da svi vokseli iza promatrane površine imaju istu težinu. Za kontinuirane objekte, za voksele koji su bliže veća je vjerojatnost da će biti zauzeti nego za oni koje su dalje. Prema tome, volumetrijska informacija temeljena na broju bliskih vokseli (engl. Proximity Count VI) određuje se na sljedeći način. Iz prethodnog položaja robota, određuju se svi pozadinski vokseli do neke udaljenosti. Za svaki voksel koji se prijeđe iza okupiranog vokseli, nagrada će se postupno smanjivati i to po pravilu  $\frac{1}{1+x}$  gdje je  $x$  udaljenost od okupiranog vokseli.

### 3.8. Grafički prikaz različitih volumetrijskih informacija

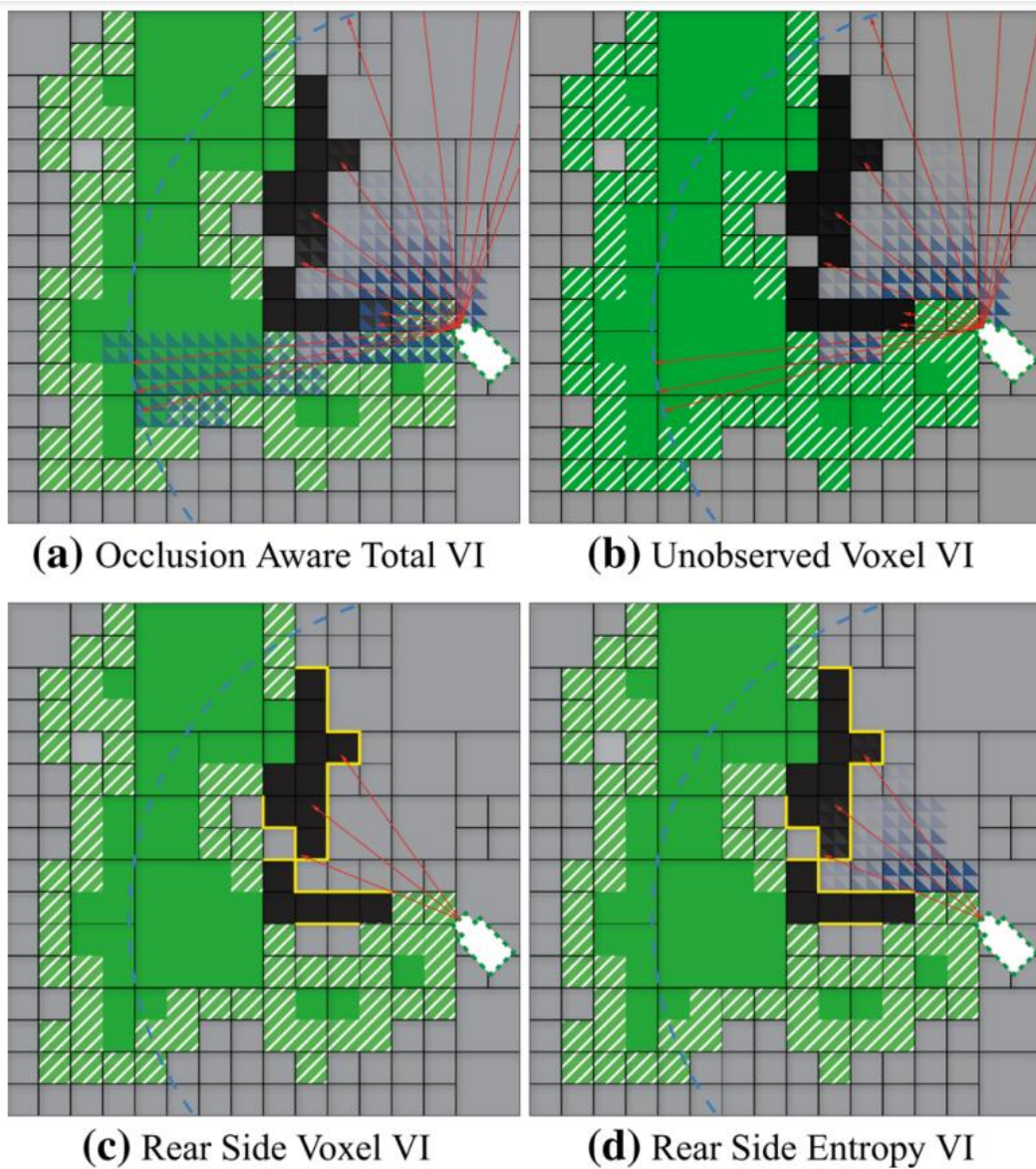
Na slici 2 prikazana je 2D mapa koja objašnjava spomenute načine računanja volumetrijske informacije. Mapa se sastoji od idućih elemenata:

- Crna polja – zauzeti vokseli
- Siva polja - neistraženi vokseli
- Zelena polja – slobodni vokseli
- Zelena prugasta polja - granični vokseli
- Žute linije – nepoznate strane objekta
- Generirane zrake – crvene linije
- Isprekidane plave linije – maksimalni doomet zraka
- Polja sa plavim trokutima – polja koja doprinose nagradi

Jedina razlika između implementiranog algoritma u ovom radu i prikazanog na slici 2 je ta što je mapa binarna, a na slici je probabilistička. To znači da u ovome radu ne postoje zelena prugasta polja, odnosno za svaki otkriveni voksel se zna sa sigurnošću je li zauzet ili nije.

Na slici a) se vidi princip volumetrijske informacije temeljene na okluziji. Kao što je već rečeno, vokseli koji nisu istraženi imaju veću entropiju, a oni koji se nalaze bliže LiDaR-u imaju veću vjerojatnost vidljivosti pa time daju veću nagradu. Budući da se koristi binarna mapa, slika a) i slika b) bi trebale biti iste jer i a) i b) donose nagradu samo ako se zraka uperi ka vokselu koji nije istražen.

Slike c) i d) donose nagradu ako se zrake upere u prethodno definirane nepoznate pozadinske voksele. Volumetrijska informacija temeljena na broju bliskih vokselu radi na sličnom principu, ali ga je teško jasno vizualizirati zbog težina koje ovise o udaljenosti.



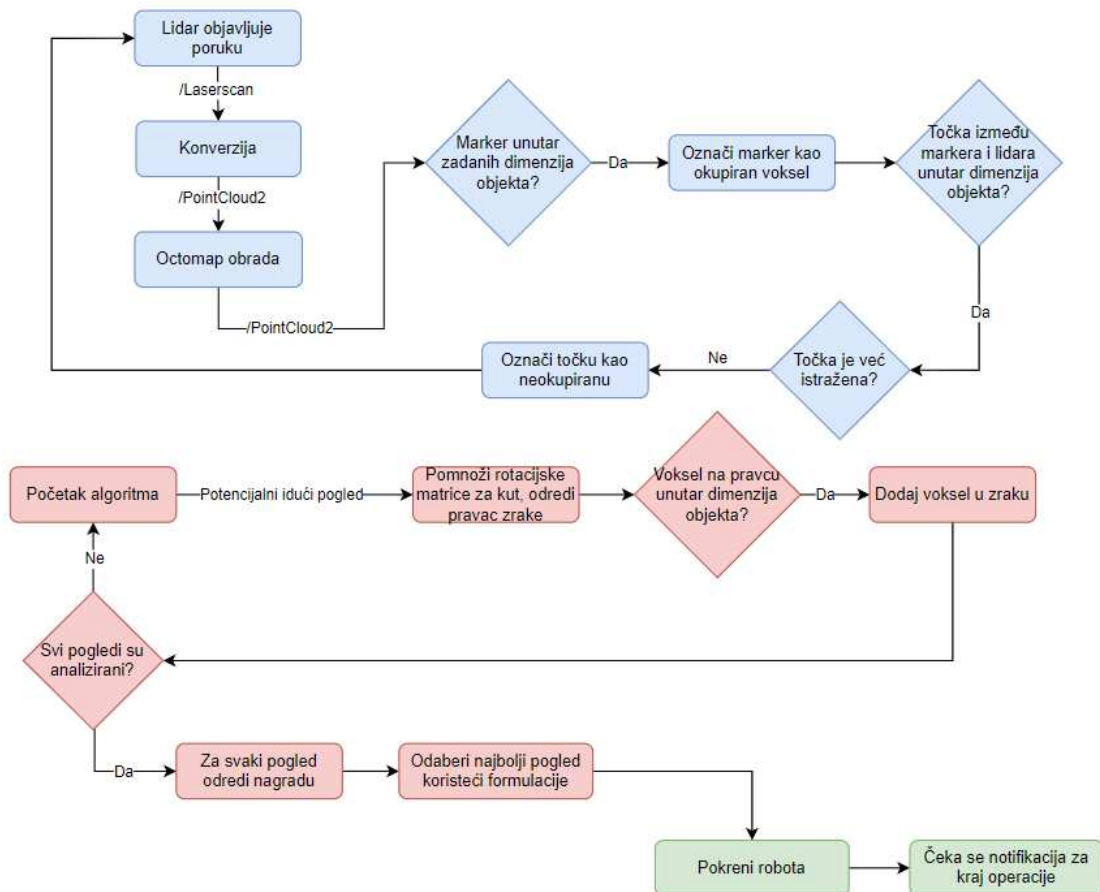
Slika 2. Grafički prikaz načina računanja volumetrijske informacije [1]; a) Volumetrijska informacija temeljena na okluziji; b) Volumetrijska informacija temeljena na nepoznatim vokselim; c) Volumetrijska informacija temeljena na pozadinskim vokselim; d) Volumetrijska informacija temeljena na entropiji pozadinskih voksela

## 4. Metoda za rekonstrukciju 3D objekta temeljena na volumetrijskoj informaciji

Dijagram metode prikazan je na Slici 3.

Metoda se sastoji od tri funkcionalne cjeline:

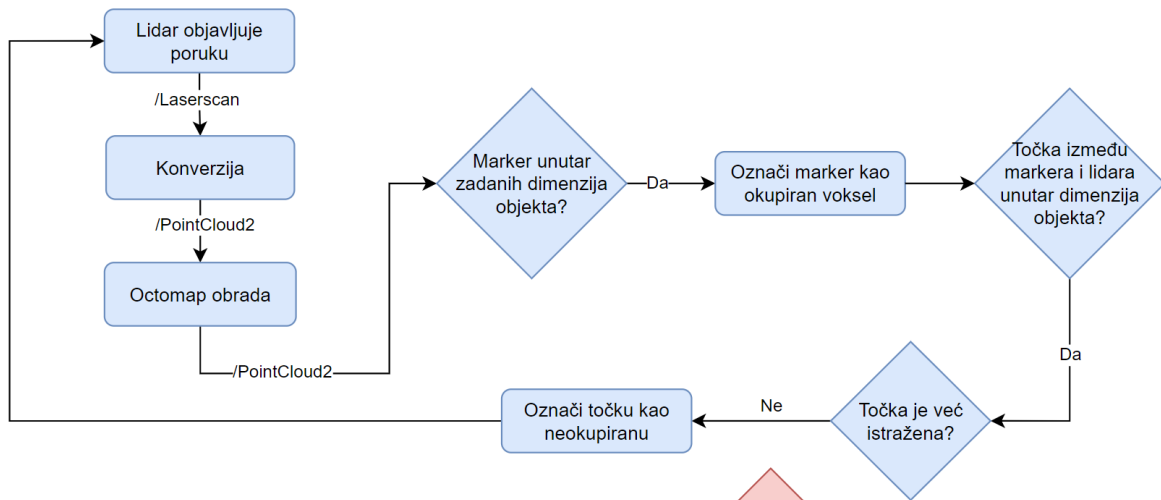
- 1) Ažuriranje 3D mape na temelju mjerenja LiDaR-a (označeno plavom bojom na Slici 3.)
- 2) Odabir idućeg pogleda na temelju procjene volumetrijske informacije (označeno crvenom bojom na Slici 3.)
- 3) Promjena položaja robota u odabrani pogled (označeno zelenom bojom na Slici 3.)



Slika 3. Dijagram sustava



## 4.1. Ažuriranje 3D mape na temelju mjerenja LiDaR-a



Slika 4. Dijagram procesa ažuriranja 3D mape

LiDaR je postavljen na mjesto šake, odnosno na *end\_effector\_link*. LiDaR je konfiguriran tako da mjeri sve točke na udaljenosti od 0.05m do 1.00m i u rasponu od -0.5rad do 0.5rad. Ukupno 100 zraka izlazi iz LiDaR-a pa je rezolucija 0.01. ROS tema na koju će LiDaR objavljivati poruke je */scan*. Tip ovih poruka je *sensor\_msgs/Laserscan* dok je po dokumentaciji *Octomap* biblioteke potreban tip ulaznih poruka *sensor\_msgs/PointCloud2*.

Konverzija se vrši jednom linijom koda:

```
pc2_msg = lp.projectLaser(msg)
```

gdje je *pc2\_msg* poruka tipa *PointCloud2*, *lp* je objekt klase *LaserProjection*, a *msg* je poruka tipa *LaserScan*. *Octomap* server je pretplaćen na temu */cloud\_in* pa će tako biti nazvana tema na koju se objavljuje *pc2\_msg*. Stavljajući u funkciju *Octomap* server, koristeći teme koje server objavljuje moguće je ažurirati 3D mapu.

Većina parametara je već konfigurirana u drugim datotekama *Octomap* biblioteke. U pokretačkoj datoteci je dodano da je rezolucija zbog konzistentnosti s LiDaR-om jednaka 0.01. Pored toga ubačen je filter zemlje što omogućava razdvajanje objekta od podloge, odnosno sve ispod 0.04m po z-osi neće biti označeno kao objekt. Za frame je stavljen world jer nema promjene lokacije cijelog robota, ne radi se o mobilnom robotu.

Nakon pretplate na temu koju objavljuje *Octomap* server */occupied\_cells\_vis\_array*, ažurira se 3D mapa u callback funkciji. 3D mapa je trodimenzionalni niz koji se ručno popunjava na osnovu podataka koje daje *Octomap* server. 3D mapa je osnova za određivanje idućeg najboljeg pogleda i sve će ovisiti od toga koliko je ona ispravna. Postavlja se pitanje zašto

ovisiti o ispravnosti 3D mape kada se mogu direktno provjeravati točke koje vraća *Octomap* server. Razloga je više kao na primjer smanjivanje računске složenosti. Operacija provjere da li neki vokal ima vrijednost 0 ili 1 ima mnogo manju složenost nego provjera da li se vokal nalazi među zauzetim vokselima u povratnoj funkciji. Drugi bitan razlog je nemogućnost razlikovanja neokupiranih vokselâ od neistraženih vokselâ. *Octomap* server vraća samo okupirane voksele, a onda bi nastao problem kod generiranja zraka za idući pogled jer ne postoji nigdje informacija je li vokal neistražen ili neokupiran.

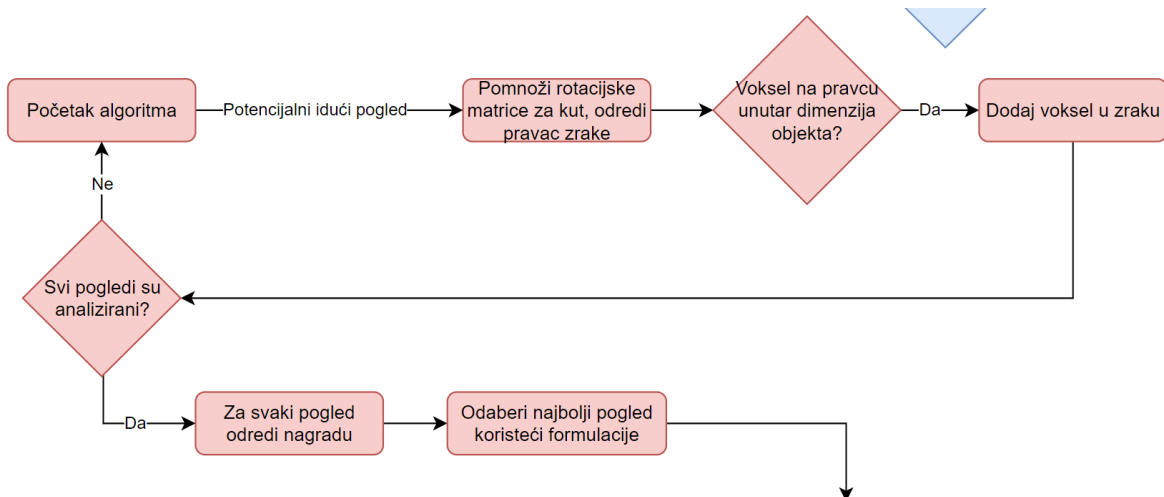
Kao što je u uvodu rečeno, model ne mora biti poznat, ali moraju biti približne dimenzije kako bi znali kojih dimenzija će biti 3D mapa. Početna vrijednost svih vokselâ je 0.5 što znači da su neistraženi.

Na početku callback funkcije, pomoću teme */tf* dobiva se položaj LiDaR-a. Nakon toga prolazi se kroz sve točke u nizu točaka kojeg vrati *Octomap*. Traži se marker koji predstavlja kraj zrake.

Zbog toga što postoji 100 zraka emitiranih iz LiDaR-a postoji i potencijalnih 100 markera. Iz svakog ovog markera je potrebno izvući (x,y,z) točke gdje zraka završava, odnosno gdje nailazi na objekat. Dalje pomoću formule se ta točka konvertuje u index koji se može zapisati u 3D mapu. Ono što je bitno je i označiti voksele između LiDaR-a i pronađenog okupiranog vokselâ kao neokupirane. Ukoliko vokseli nisu prije istraženi treba im dati vrijednost 0 odnosno označiti ih kao neokupirane a pronađeni vokal označiti kao okupiran (vrijednost 1). Tokom cijelog procesa se vodi računa da su vokseli unutar dimenzija objekta.

## **4.2. Odabir idućeg pogleda na temelju procjene volumetrijske informacije**

Da bi se odredila nagrada, odnosno VI, za svaki pogled, generira se stotinu zraka za svaki pogled koji još nije ostvaren. Zraka se generira u 3D mapi. Dijagram je prikazan na slici 5.



Slika 5. Dijagram procesa odabira idućeg najboljeg pogleda

U funkciju za generiranje zraka se prosljeđuje pozicija i orijentacija koju bi LiDaR trebao imati u tom pogledu ukoliko on bude izabran. U nastavku se generiraju matrice rotacije te se one množe.

Rotacijska matrica opisuje transformaciju vektora iz originalnog koordinatnog sistema u rotirani za neku poznatu rotaciju oko x, y ili z osi. Rotacijske matrice oko x, y i z osi su prikazane u nastavku:

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix} R_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} R_z(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

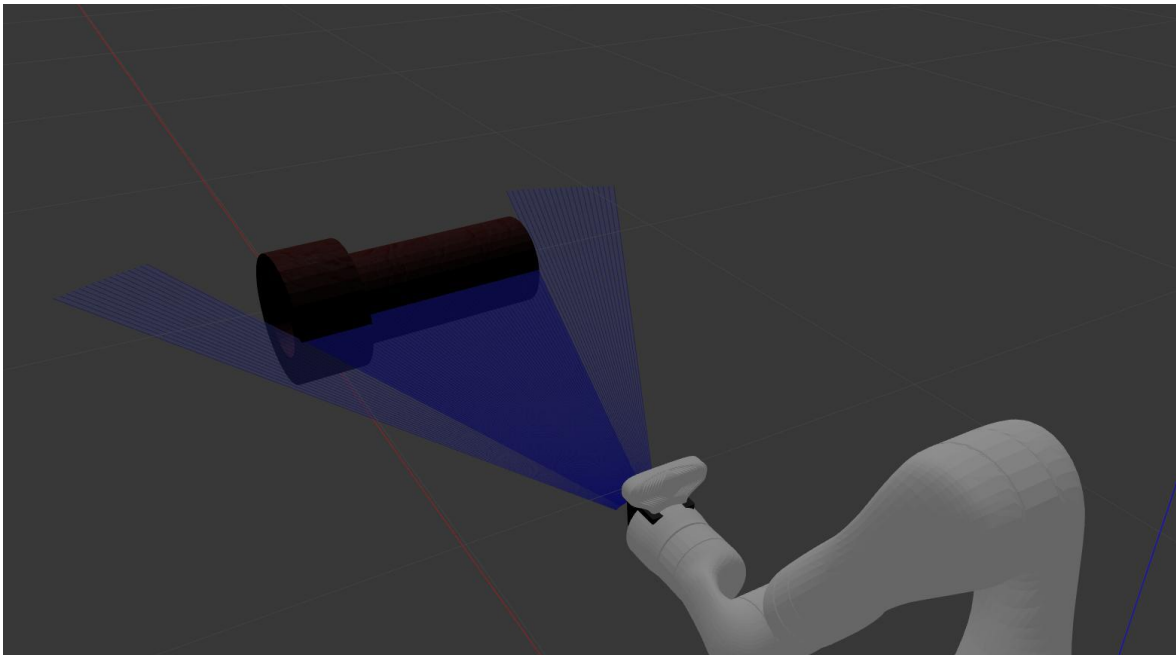
Slika 6. Matrice rotacije oko x, y i z osi

Za svaki pogled se ove matrice generiraju 100 puta jer iz LiDaR-a izlazi 100 zraka. U konfiguraciji LiDaR-a je određeno da su zrake razdvojene horizontalno. Zbog toga što ih ukupno ima 100, a rezolucija je 0.01, svaki put na kut poniranja treba dodati varijabilni kut koji se nalazi u rasponu od  $-0.5\text{rad}$  do  $0.5\text{rad}$ . Ukoliko se  $\alpha$  uzme kao kut valjanja (engl. Roll),  $\beta$  kao kut poniranja (engl. Pitch) i  $\gamma$  kao kut skretanja (engl. Yaw), onda se dobije 3-2-1 rotacijska matrica prikazana pomoću Eulerovih kuteva:

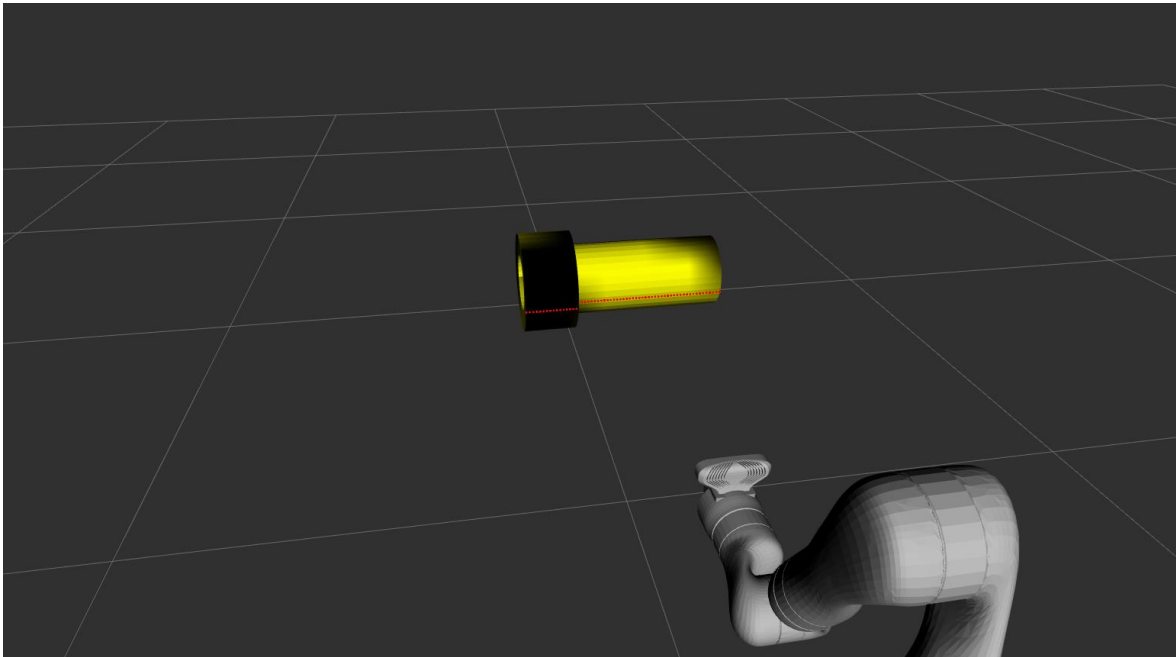
$$\begin{aligned}
R = R_z(\gamma) R_y(\beta) R_x(\alpha) &= \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \\
&= \begin{bmatrix} \cos \beta \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma \\ \cos \beta \sin \gamma & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma \\ -\sin \beta & \sin \alpha \cos \beta & \cos \alpha \cos \beta \end{bmatrix}
\end{aligned}$$

Slika 7. 3-2-1 matrica prikazana preko Eulerovih kuteva

Treći stupac dobivene matrice je vektor pravca zrake. Sada svaki voksel koji se nalazi na pravcu zrake od LiDaR-a pa sve do granica 3D mape ili okupiranog vokselu, treba ubaciti u niz *ray*. Ako se dođe do kraja zrake, kut se povećava za 0.01 i ponavlja se proces za iduću zraku. Na slici 8 je prikazan raspored ovih zraka na primjeru cijevi, a na slici 9 su prikazane točke dodira zraka i predmeta.



Slika 8. Raspored generiranih zraka



Slika 9. Točke dodira

Robot bira svoj idući položaj i orijentaciju među 40 ponuđenih pogleda.

Prvo se uzmu potencijalni sljedeći položaji, tj. one u kojima ruka još uvijek nije bila. Nakon toga se generira 100 zraka za svaki od potencijalnih položaja. Sada u ovisnosti koja formulacija je odabrana, računa se nagrada za svih 100 zraka svakog pogleda. Zbroj svih 100 nagrada je konačna nagrada za taj pogled. Na kraju se bira pogled za najvećom nagradom.

### 4.3. Promjena položaja robota

U ovaj korak je bitno proslijediti pogled iz prethodnog koraka, odnosno poziciju i orijentaciju iduće najbolje točke. U varijablu tipa *ExecuteActionRequest* se proslijedi iduća najbolja točka, robot kreće mijenjati položaj, a kada dođe na određite dobiva se notifikacija o tome i prelazi se na planiranje idućeg najboljeg pogleda ukoliko nije već izvršen traženi broj iteracija.

## 5. Simulacija

Pored spomenutih načina računanja volumetrijske funkcije iz drugog poglavlja, radi usporedbe će biti implementirana jednostavna metoda 3D rekonstrukcije koja predmet skenira sa 4 strane, a za svaku stranu će biti odabrana 2 pogleda. Prvi pogled će biti usmjeren ka najsjevernijoj točki na toj strani, a drugi ka najjužnijoj. Pored toga bit će prikazano i koliki postotak skeniranosti vidljive površine objekta će imati rekonstrukcija koja idući pogled bira slučajnim odabirom. Ovo se radi da bi se došlo do zaključka isplati li se koristiti rekonstrukciju na temelju volumetrijske informacije. Ukoliko postotak skeniranosti bude isti ili sličan onda se ne isplati jer će zasigurno bez nje robotu trebati manje vremena za skeniranje, odnosno smanjit će se računaska složenost programa.

U simulaciji će se koristiti 3 predmeta različitih složenosti. Da bi se znao odrediti postotak skenirane površine predmeta od ukupne površine koristi se Python biblioteka *trimesh* [6]. Ova biblioteka sadrži funkcije koje čitaju datoteke sa opisom objekta i vraćaju vidljivu površinu upisanog modela u kvadratnim metrima.

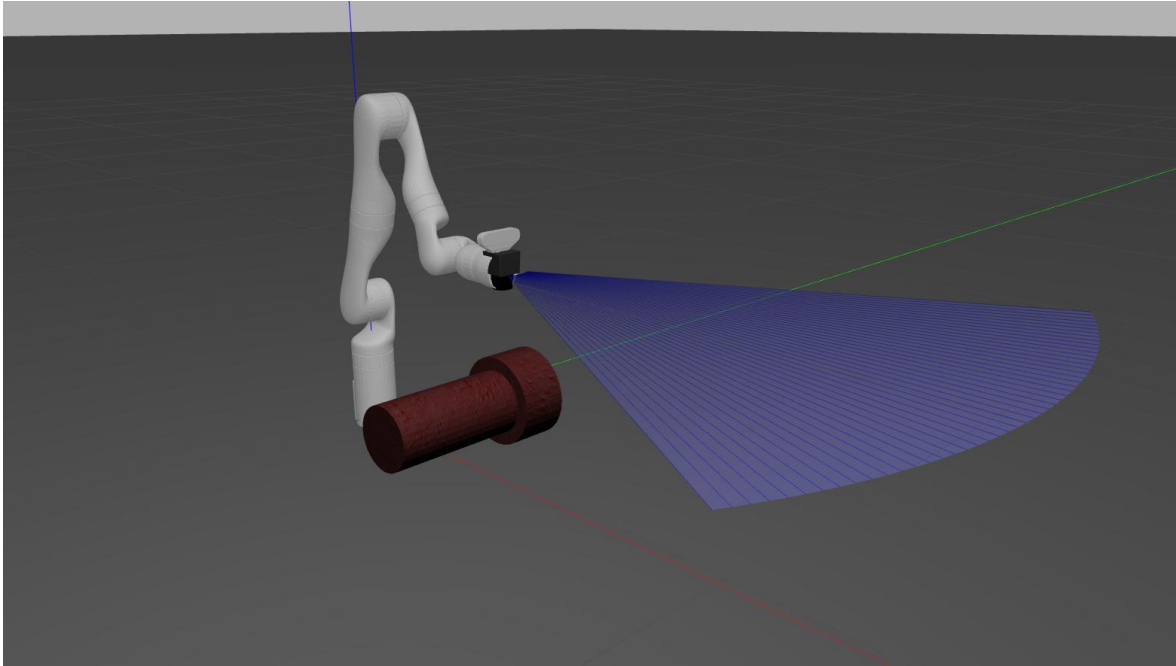
Za pokretanje simulacije potrebno je pokrenuti:

- `roslaunch kortex_gazebo spawn_kortex_robot.launch` – Pokreće Gazebo i Rviz, inicijalizira robota i omogućava njegovo kretanje inicijalizacijom MoveIt plugina.
- `Python3 laserscan_to_pointcloud.py` – Pokreće konverziju *LaserScan* poruka u *PointCloud2*
- `roslaunch octomap_server octomap_mapping.launch` – Pokreće *Octomap* server
- `roslaunch kortex_examples cartesian_poses_with_notifications_python.launch` – Pokreće aktivnu 3D rekonstrukciju.

U Rviz-u je potrebno frame staviti na world, te uključiti *RobotModel*, *MarkerArray* i *PointCloud2* opcije. Za prikaz mape koju daje *Octomap*, *MarkerArray* je potrebno pretplatiti na temu `/occupied_cells_vis_array`. Za prikaz interne 3D mape, *PointCloud2* je potrebno pretplatiti na temu `/voxel_grid_pointcloud`.

### 5.1. Cijev

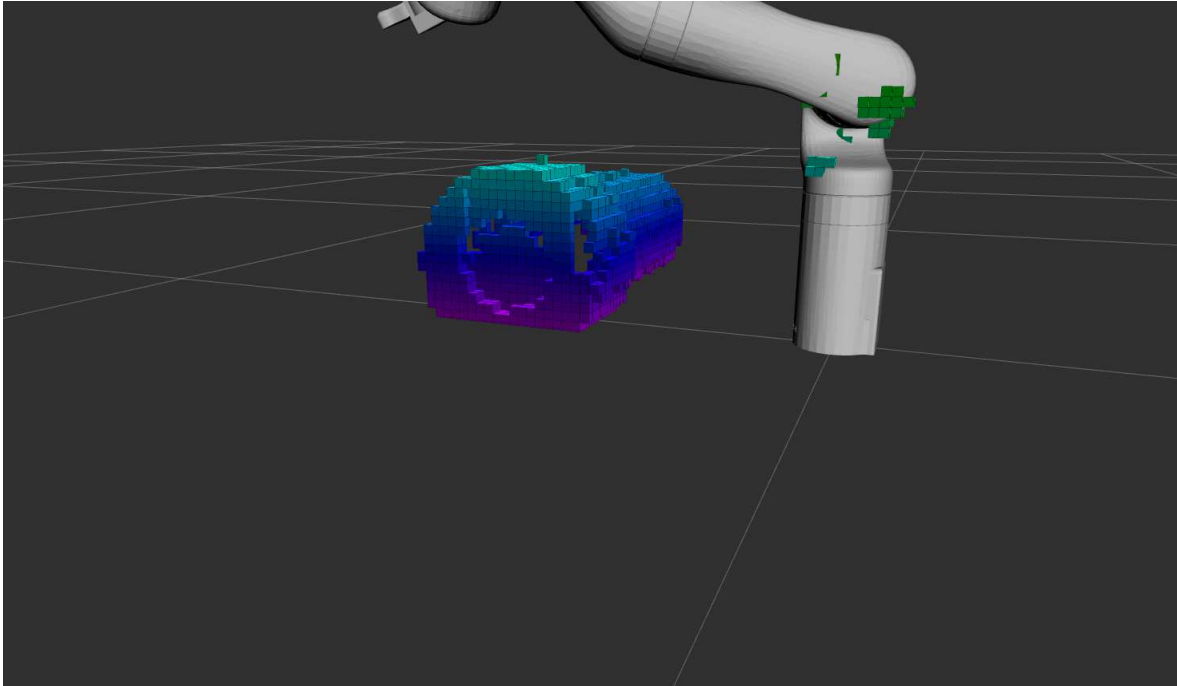
Prvi najjednostavniji predmet, u simulaciji, prikazan je na Slici 10 [7].



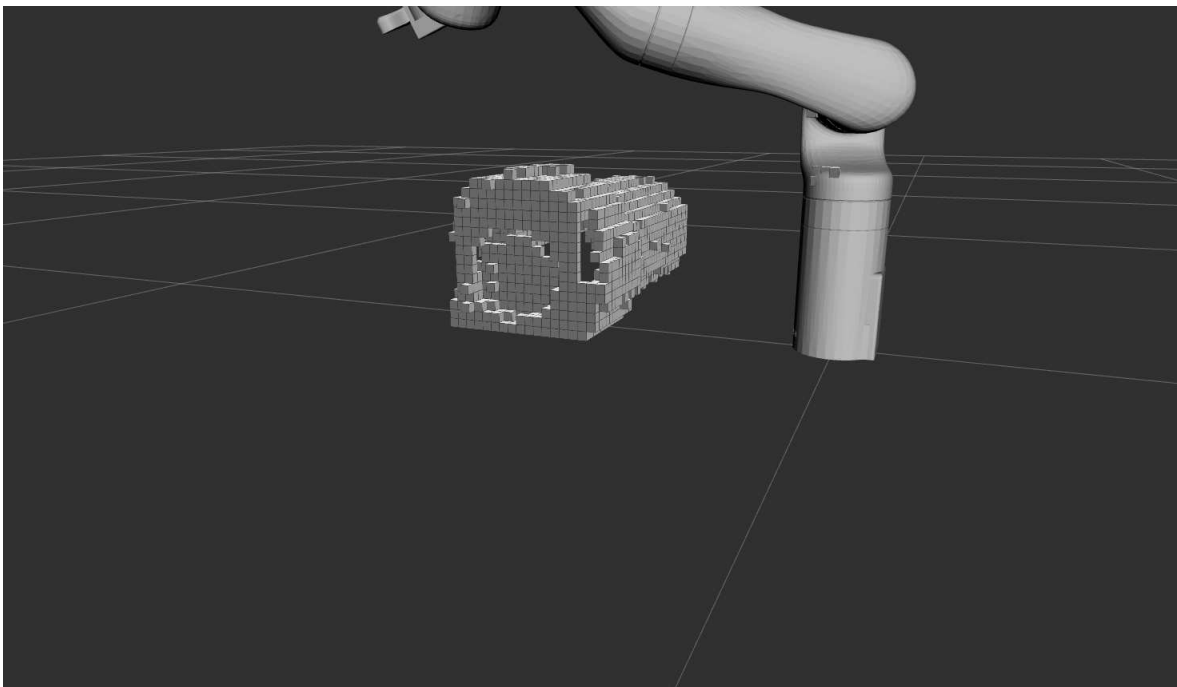
Slika 10. Početak simulacije u Gazebu

Model će se ograničiti s 0.5m po širini i dužini i s 0.3m po visini. Vidljiva površina predmeta je 0.28 kvadratnih metara. Prosječno vrijeme skeniranja ovog predmeta koristeći VI temeljenu na okluziji je 4-5min.

Nakon što je robot bio u 10 različitih položaja, 85% površine predmeta je skenirano. Slika 11 predstavlja ono što je zapravo skenirano i što Octomap vraća da je dio predmeta na osnovu čega se računa skenirana površina, dok će se na slici 12 prikazati ono što je zapisano u internu 3D mapu i na osnovu čega se upravlja robotom. Sličnost ove dvije slike govori da dio za ažuriranje informacija sustava radi dobar posao.



Slika 11. Model cijevi, VI temeljena na okluziji, Octomap rezultat



Slika 12. Model Cijevi, VI temeljena na okluziji, interna 3D mapa

U nastavku neće biti prikazivane slike za svaki od načina izračuna VI već će samo biti prikazana tablica s rezultatima. Rezultati za model cijevi su dani u sljedećoj tablici:



VI / Iteracija	2	4	6	8	10
VI temeljena na okluziji	32.7%	32.7%	51%	75.5%	85.8%
VI temeljena na nepoznatim vokselima	34,3%	38.8%	61.9%	84.7%	87.4%
VI temeljena na pozadinskim vokselima	32.8%	36%	42.9%	58.5%	63.3%
VI temeljena na entropiji pozadinskih vokseli	32.5%	36%	42.5%	59.6%	71.6%
VI temeljena na broju bliskih vokseli	32.2%	40.9%	56.4%	61.7%	66.4%
Nasumični odabir pogleda	27.4%	29.7%	55.5%	55.9%	60.9%
Metoda 4 strane	12.3%	32.4%	36.5%	43.8%	46.5%

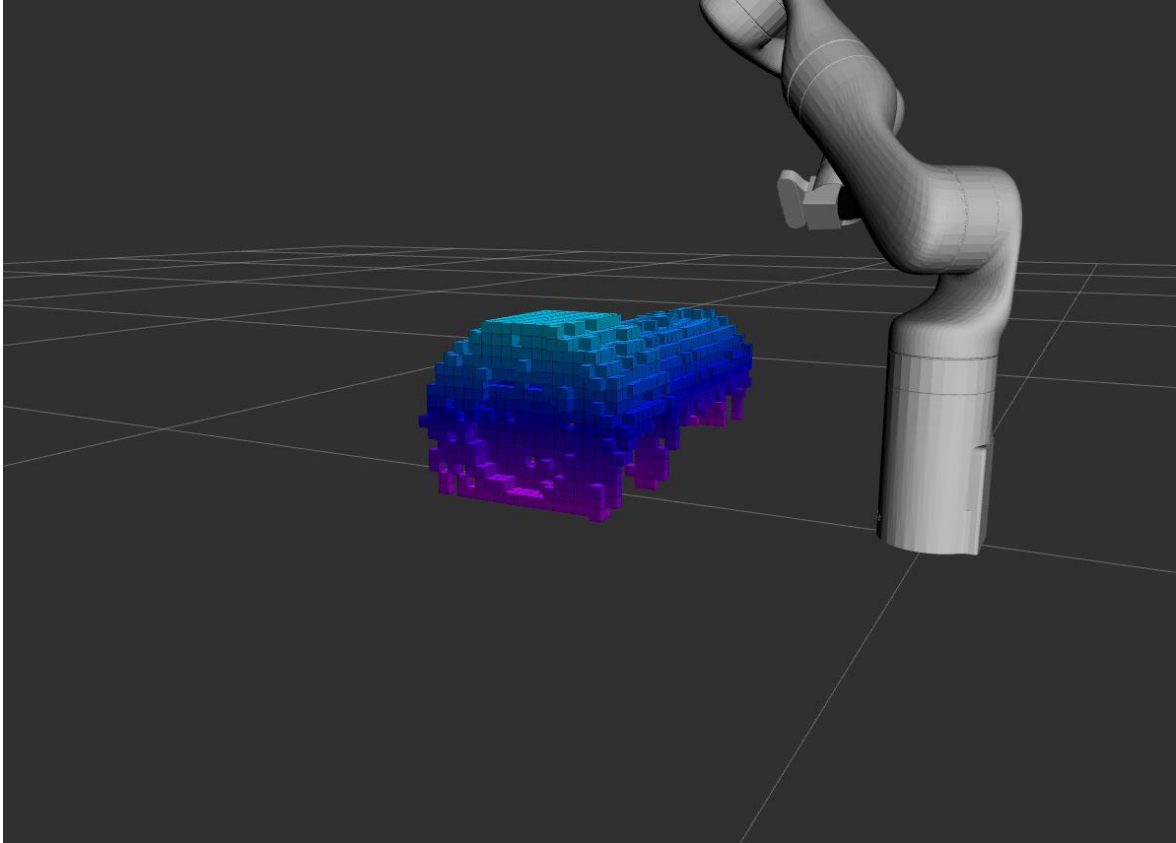
Tablica 1. Rezultati za predmet cijev

Ova mjerenja kao i sva mjerenja u nastavku su za svaku VI ponovljena 3 puta sa korištenim istim početnim uvjetima (ako nije drugačije naglašeno). Generalno se dobije dobra pokrivenost površine objekta s obzirom na to da otprilike 10% površine nije vidljivo jer se predmet nalazi na zemlji. Pored toga, uključen je filter zemlje pa sve ispod 0.04 ne može biti okupiran voksel.

Pokrivenost površine predmeta varira 10% - 15%. U tabeli 1 su prikazane prosječne vrijednosti. Bez obzira na varijaciju, rezultati dobiveni primjenom prve dvije metode izračuna VI su bolji od ostalih. Eksperiment je ponavljan više puta i te metode su uvijek imale bolje rezultate. Pokrivenost sa VI temeljenoj na pozadinskim vokselima, VI temeljenoj na entropiji pozadinskih vokseli, VI temeljenoj na broju bliskih vokseli i metodi nasumičnog odabira je uvijek bila između 60% i 70%, dok je sa VI temeljenoj na okluziji i VI temeljenoj na nepoznatim vokselima između 70% i 85%. Ovdje treba napomenuti da su se pozadinskim vokselima u metodama VI temeljenoj na pozadinskim vokselima, VI temeljenoj na entropiji pozadinskih vokseli, VI temeljenoj na broju bliskih vokseli smatrali samo pozadinski vokseli iz posljednjeg najboljeg pogleda.

Kada bi se uzimali pozadinski vokseli iz svih prethodnih najboljih pogleda, za očekivati je da bi rezultati bili i bolji. Primjena VI temeljena na pozadinskim vokselima sa svim prethodnim pozadinskim vokselima testirana je u jednoj simulaciji, a za izvršavanje jedne iteracije je bilo potrebno 20min.

Nasumična metoda ne daje tako loše rezultate u usporedbi s metodama koje se oslanjaju na pozadinske voksele. Ono što je najzanimljivije je da se metoda koja skenira 4 strane bez ikakvog algoritma pokazala kao najgora gledajući tabelu, međutim na slici 13 je prikazan rekonstruiran model pomoću nje.



Slika 13. Model rekonstruiran metodom 4 strane

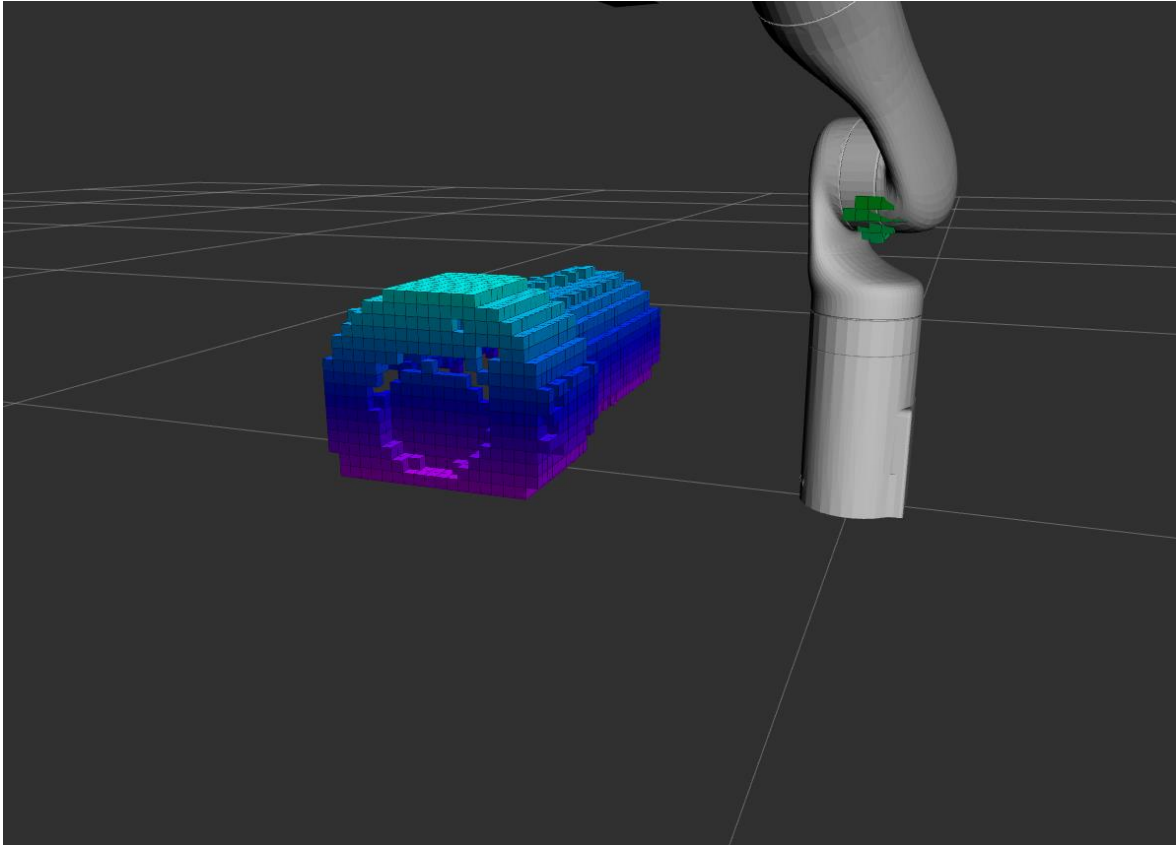
Gledajući sliku može se zaključiti da je model rekonstruiran skoro kao model sa slike 11. Razlog za ovako lošu pokrivenosti nije skroz jasan, ali gledajući oba modela sa svih strana utvrđeno je da je model skeniran metodom temeljenoj na okluziji popunjeniji s vana, a pogotovo unutra. Objekt je sasvim dobro rekonstruiran metodom 4 strane, međutim nedostaje gustoće vokselu vjerovatno zbog toga što se ova metoda dosta brže izvrši nego ostale, LiDaR puno brže pređe sve voksele pa neke od njih *Octomap* ne označi kao zauzete. Sada će se promijeniti početni uvjeti tako da robot iz početnog položaja neće imati nikakve informacije za razliku od prethode analize gdje je već u početnom položaju imao oko 25% skenirane površine predmeta. Rezultati su prikazani u tabeli 2:

VI / Iteracija	2	4	6	8	10
----------------	---	---	---	---	----

VI temeljena na okluziji	28.5%	47.5%	48.5%	66.7%	78.2%
VI temeljena na nepoznatim vokselima	17.1%	39.4%	45.6%	63.8%	63.8%
VI temeljena na pozadinskim vokselima	16.5%	33.5%	33.6%	39.5%	55%
VI temeljena na entropiji pozadinskih vokseli	16.8%	32.6%	36.7%	40.8%	54.2%
VI temeljena na broju bliskih vokseli	17.3%	40%	57.6%	63.3%	68.9%

Tabela 2. Rezultati za promijenjene početne uvjete

Očekivano dobiju se nešto lošiji rezultati, tako na slici 14 može se vidjeti rekonstruiran model koristeći VI temeljenu na okluziji. Vidi se da je rekonstrukcija opet kvalitetno obavljena, osim što se sa stražnje strane pojavila rupa koja nije vidljiva na slici pa je zato smanjena pokrivenost u ovom slučaju na 78%. Kod svih ostalih formulacija je riječ o sličnoj stvari pa je svakoj smanjen učinak za cca 10% osim kod VI temeljene na broju bliskih vokseli koja je nešto bolja za ove početne uvjete. Zbog činjenice da rezultati variraju 10-15% ne mora značiti nužno da je ovo smanjenje uspješnosti uzrokovano promjenom početnih uvjeta.



Slika 14. Rekonstruiran model sa promijenjenim početnim uvjetima

U tabeli 3 je prikazano trajanje izvršavanja algoritma za svaku metodu izračuna volumetrijske informacije. Mjerenja su ponavljana 2 puta, a vršena u jednakim uvjetima jedno za drugim.

VI	Trajanje rekonstrukcije
VI temeljena na okluziji	4-5min
VI temeljena na nepoznatim vokalima	4-5min
VI temeljena na pozadinskim vokalima	5-7min
VI temeljena na entropiji pozadinskih vokseli	5-7min
VI temeljena na broju bliskih vokseli	4-5min
Nasumično	3-4min

Metoda 4 strane	<1min
-----------------	-------

Tabela 3. Trajanje skeniranja objekta za svaku formulaciju

## 5.2. Igračka

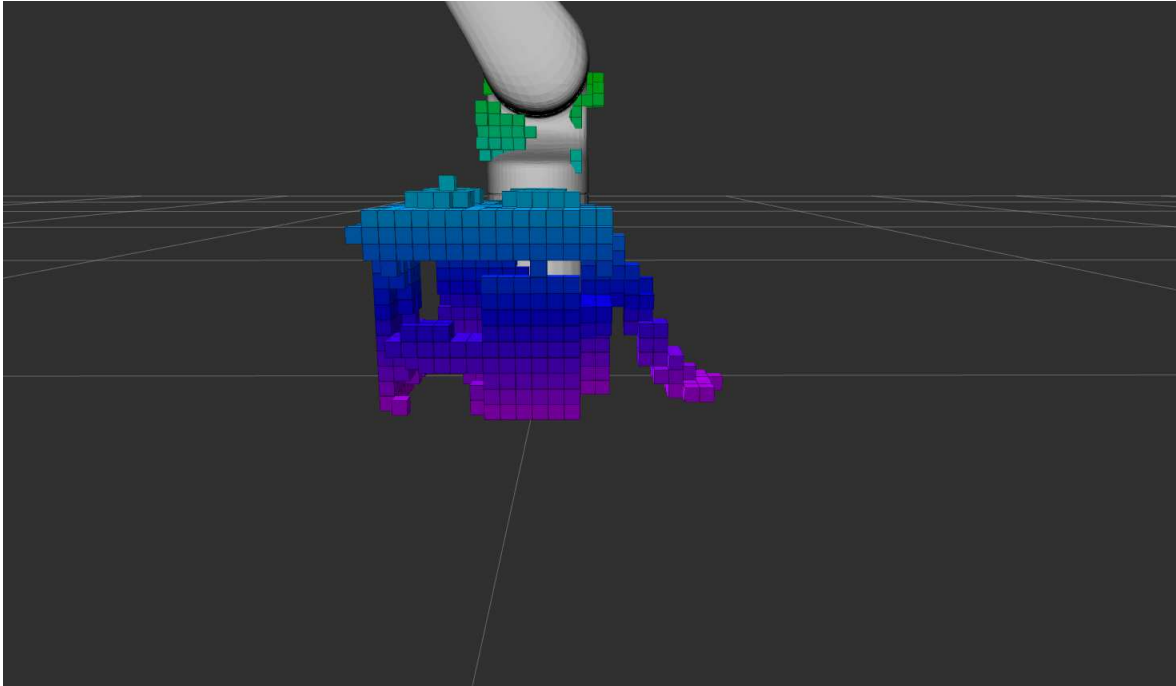
Sada će biti rekonstruiran mnogo složeniji predmet od prethodnog:



Slika 15. Igračka za djecu [8]

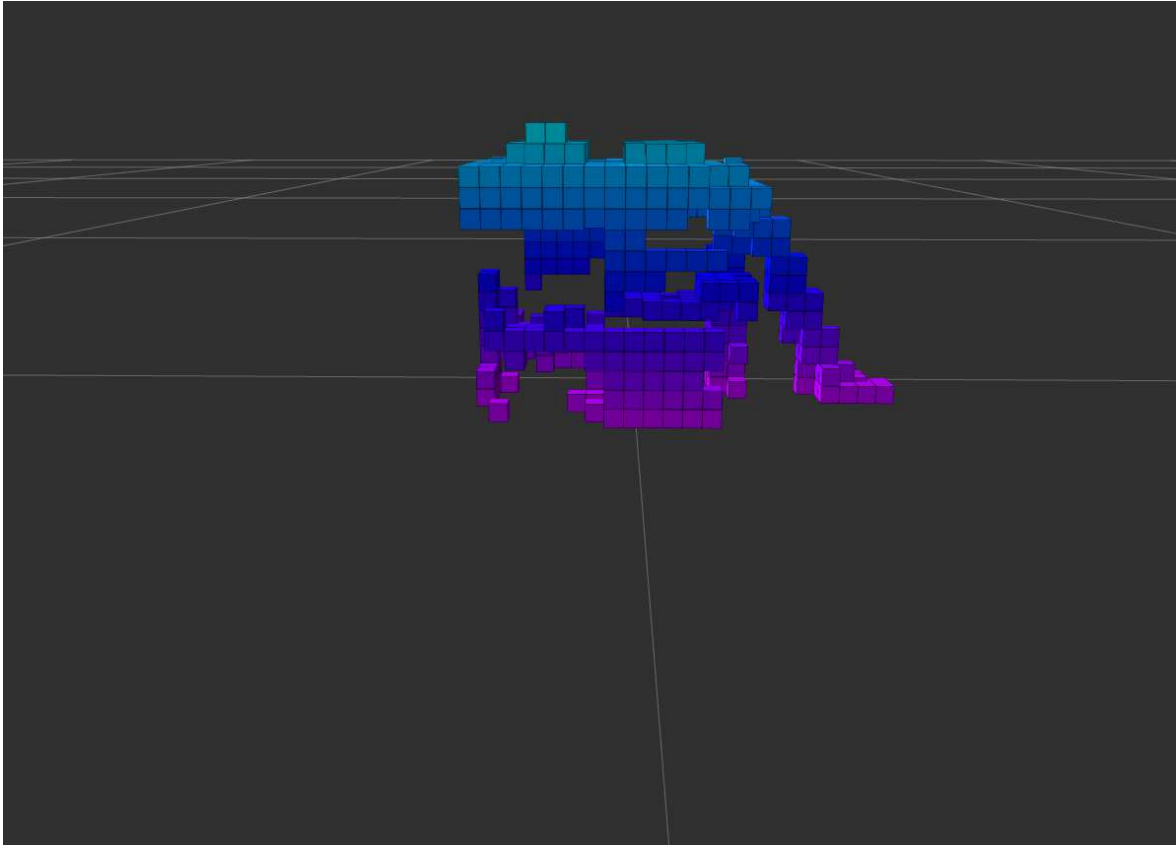
Radi se o igrački za djecu. Odmah je na prvu vidljivo koliko je predmet složen zbog svog unutaršnjeg prostora. Predmet je zapreminom dosta manji od cijevi u prvom primjeru, ali je njegova vidljiva površina zbog šupljina gotovo ista i iznosi 0.22 kvadratnih metara.

Na slici 16. je prikazan rezultat rekonstrukcije za VI temeljenu na okluziji:



Slika 16. Rekonstrukcija složenog predmeta koristeći VI temeljenu na okluziji

Postotak skeniranosti vidljive površine iznosi 50% što je lošije nego kod prvog predmeta. Ovo je donekle bilo i očekivano jer se radi o predmetu koji pored vanjskog dijela ima i složenu unutrašnjost pa je moguće da se određeni zidovi skeniraju prije nego unutrašnjost i na taj način se ona nikad i ne skenira. Također ovaj predmet ima manju vidljivu površinu i drugačijeg je oblika, odnosno ne sužava se u donjem dijelu, pa ako je kod prvog predmeta 10% otpadalo na voksele koji ne mogu biti skenirani (ground filter i podloga), onda je to kod ovog predmeta negdje oko 30%. Gledajući sliku 16 može se zaključiti da je konstrukcija predmeta dobro skenirana, ali da nedostaju detalji. Također na slici 17 će biti prikazan rezultat za VI temeljenu na pozadinskim vokselima gdje je postotak skeniranosti samo 37%.



Slika 17. Rekonstrukcija složenog predmeta koristeći VI temeljenu na pozadinskim vokselima

U tabeli 4. se mogu vidjeti rezultati rekonstrukcije za ovaj predmet:

VI / iteracija	2	4	6	8	10
VI temeljena na okluziji	22.3%	44.7%	44.7%	47.5%	50%
VI temeljena na nepoznatim vokselima	26%	35.5%	44.7%	47.8%	49.8%
VI temeljena na pozadinskim vokselima	21.3%	26.2%	29.6%	34%	37.2%
VI temeljena na entropiji pozadinskih vokseli	21.3%	21.9%	29.3%	30%	30%
VI temeljena na broju bliskih vokseli	21.5%	37.5%	40%	40.5%	45.5%
Nasumično	25.3%	38.8%	43.5%	48.7%	49.5%
Metoda 4 strane	10.8%	20.5%	27.7%	27.7%	30.3%

Tabela 4. Rezultati za model igračke

Za manji, a složeniji predmet se isplati koristiti nasumičnu metodu. Naravno, točke treba pažljivo birati, odnosno treba ponuditi robotu samo one poglede od kojih će imati koristi. Za ostale metode to nije neophodno jer sam robot koristeći određenu metodu zaključi da ne treba odabrati pogled koji primjera radi gleda u nebo.

VI koje koriste pozadinske voksele se nisu dobro pokazale na ovom predmetu. Dodavanje novih pogleda niti povećanje broja iteracija nije doprinijelo poboljšanju pokrivenosti, budući da su svi novi pogledi s volumetrijskom vrijednošću nula. Što se tiče rješenja, najbolje bi bilo promijeniti trenutne ponuđene poglede ili početni položaj robota. Promatrajući izvršavanje metoda koje se temelje na pozadinskim vokselima da se primijetiti da robot veći dio vremena provede skenirajući predmet odozgo. To radi jer nije nikad otišao na stranu predmeta gdje bi mu pozadinski vokseli bili opet na strani, samo suprotnoj, ovako je na neki način zaglavljn.

Metoda 4 strane daje loš rezultat, ali ukoliko je potreban predmet skeniran sa 4 strane sa vrlo malo detalja onda se može iskoristiti ova metoda jer se dosta brže izvršava.

U tabeli 5 su prikazana vremena izvršavanja algoritama za ovaj predmet. Vidi se da nema velike razlike u odnosu na prvi predmet osim kod VI temeljene na pozadinskim vokselima i VI temeljene na entropiji pozadinskih vokseli. Naime ove dvije formulacije za ovaj slučaj ne rade ispravno, često se dešavalo da nagrada za svaki pogled u nekoj iteraciji bude 0, što znači da nema nikakvog računanja i prolaženja kroz niz tako da su se ove dvije formulacije i brže izvršavale.

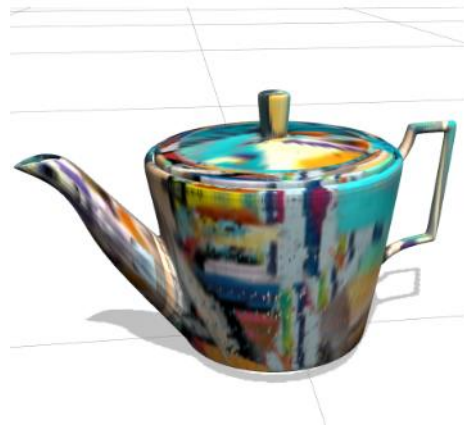
VI	Trajanje rekonstrukcije
VI temeljena na okluziji	4-5min
VI temeljena na nepoznatim vokselima	4-5min
VI temeljena na pozadinskim vokselima	4-5min
VI temeljena na entropiji pozadinskih vokseli	2min
VI temeljena na broju bliskih vokseli	4-5min
Nasumično	3-4min
Metoda 4 strane	<1min

Tabela 5. Trajanje skeniranja složenijeg objekta



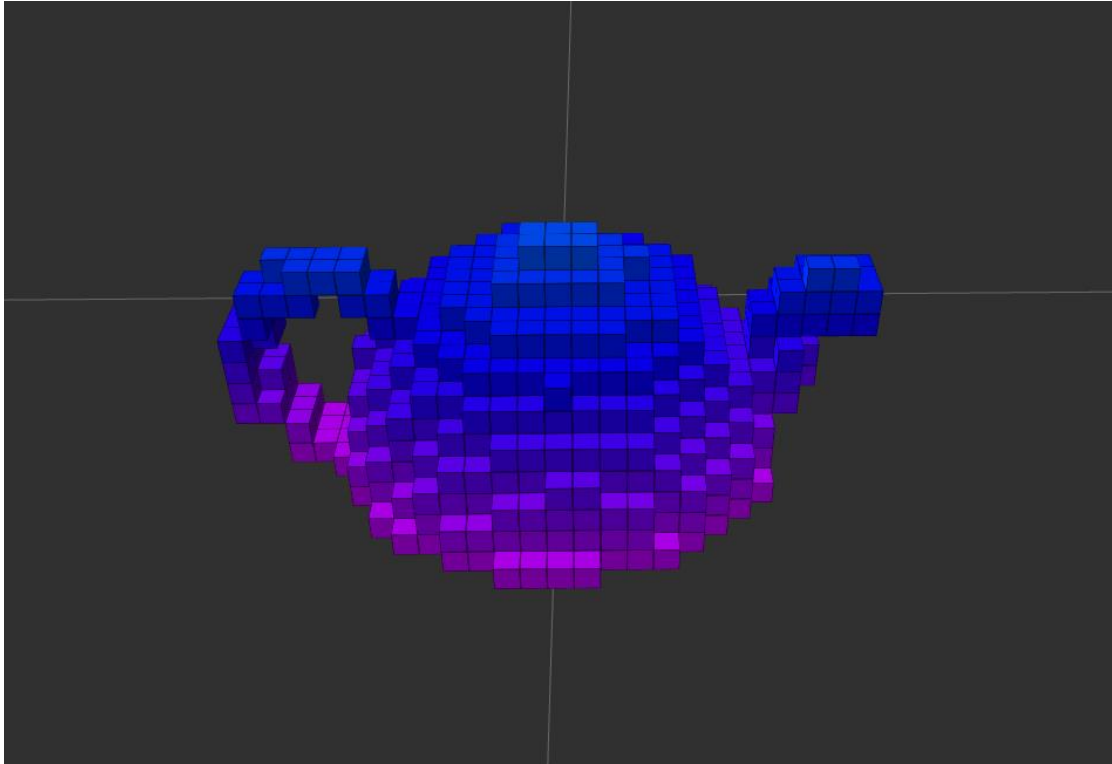
### 5.3. Čajnik

Nakon analiziranja vrlo jednostavnog i vrlo složenog predmeta došlo je vrijeme za jednu drugačiju vrstu testa. Ovaj put će biti rekonstruiran model čajnika, isti onaj koji je skeniran u radu [10]. Model koji se rekonstruira u ovom radu je nešto drugačiji od modela iz tog rada, ali ono što je bitno da se u oba slučaja radi o čajniku sličnih dimenzija i složenosti. Ovo se radi kako bi se mogla uporediti kvaliteta rekonstrukcije između ovih radova. Model iz ovog rada je prikazan na slici 18 s lijeve strane [9], a model iz navedenog rada s desne. [10]

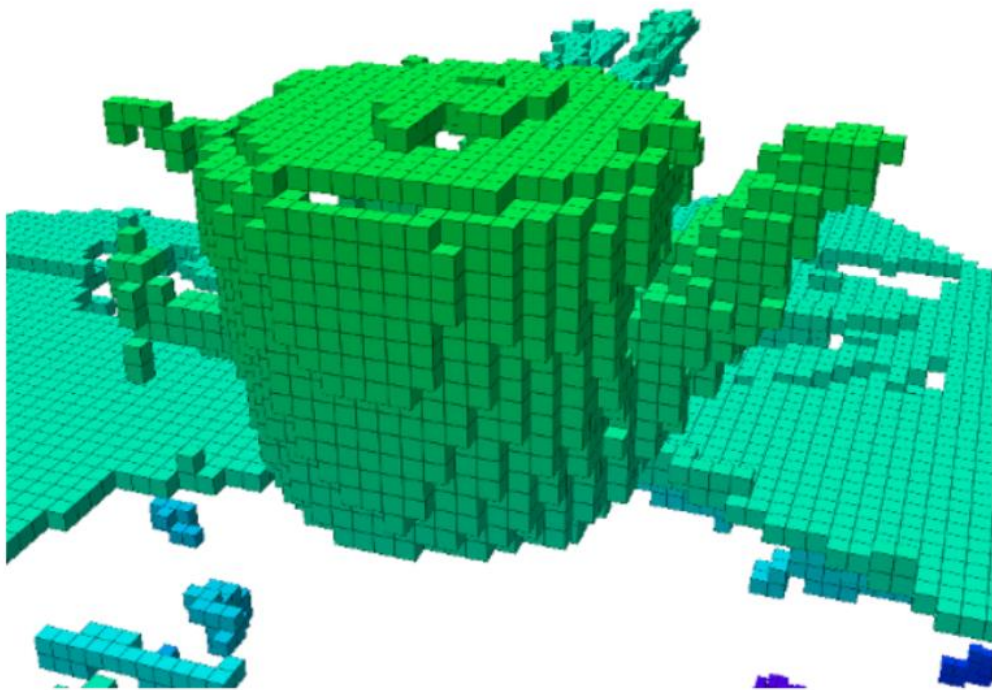


Slika 18. Dvije varijante modela čajnika

Na slici 19 je rekonstruiran model iz ovog rada metodom temeljenoj na okluziji, a na slici 20 je prikazana rekonstrukcija iz rada [10].

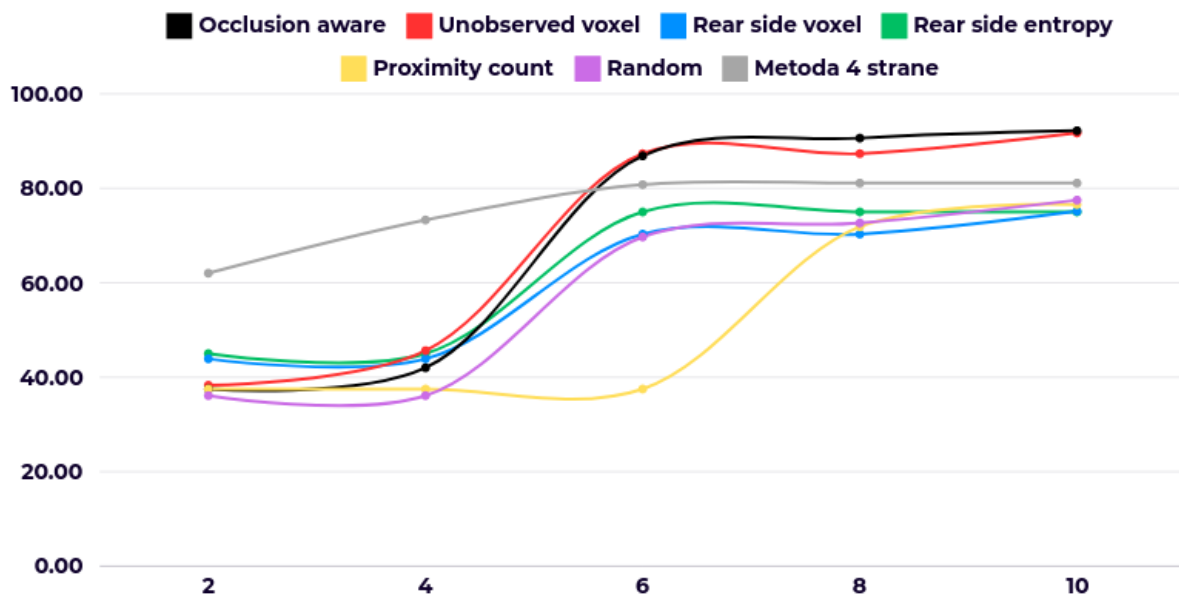


Slika 19. Rekonstruiran model čajnika metodom temeljenoj na okluziji



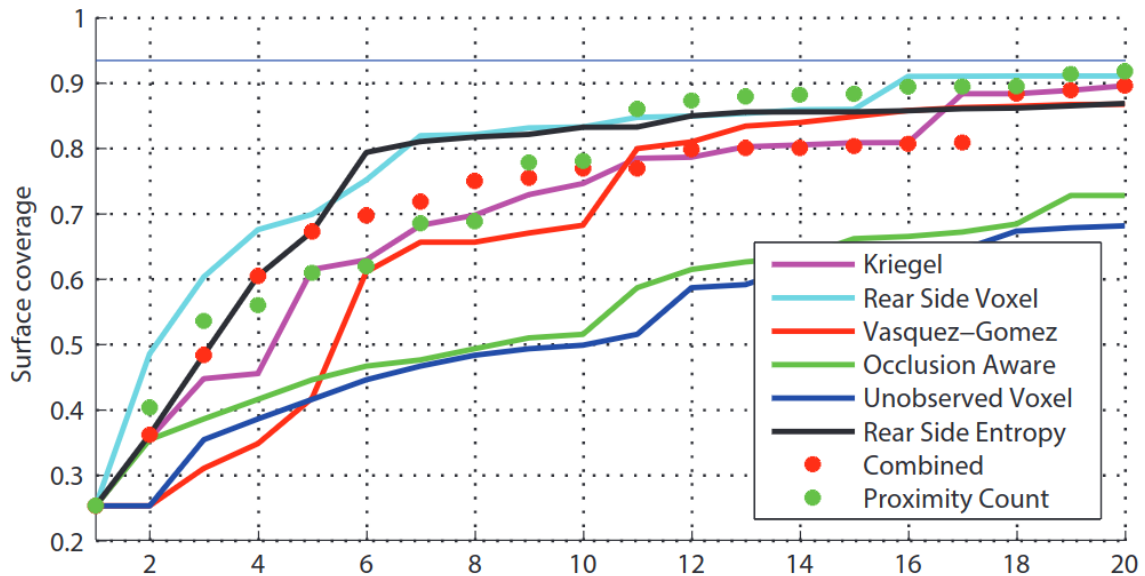
Slika 20. Rekonstruiran model čajnika [10]

Početni uvjet je stavljen tako da pri dolasku u početni položaj ne bude skeniran niti jedan voksela predmeta. Ovaj predmet je dosta manji nego prethodna dva s vidljivom površinom od 0.1 kvadratni metar i uz to širi u donjem dijelu nego u gornjem, pa u ovom slučaju filter zemlje i vokseli koji se ne vide od tla čine značajan udio u ukupnoj vidljivoj površini. Idealnim skeniranjem predmeta utvrđeno je da je maksimalna površina koja može biti skenirana iznosi oko 64% tako da će se dobiveni rezultati skalirati. Rezultati su prikazani grafički na slici 21.



Slika 21. Rezultati za model čajnika (y-os je postotak skenirane površine od ukupne površine predmeta, a x-os je broj iteracije)

Sa slike 21 je vidljivo da metode temeljene na okluziji (crna linija) i nepoznatim vokselima (crvena linija) daju visok stupanj pokrivenosti površine. Metoda 4 strane (siva linija) isto tako daje solidne rezultate u ovom slučaju s obzirom na to da nema problema s tim da robot skenira samog sebe. Naime, kod ostalih VI može se desiti da ruka skenira samu sebe pri prijelazu iz jednog pogleda u drugi, pa se onda nakon izvršenja programa moraju brojati vokseli koje su označeni kao okupirani a pripadaju konstrukciji robota i oduzimati od ukupnog broja zauzetih vokseli kako bi se dobio ispravan procent skeniranosti objekta. Metode koje se oslanjaju na pozadinske voksele (plava, zelena i žuta linija) ponovno ne daju dobre rezultate, odnosno uopće ne rade, što govori činjenica da je nasumična metoda (ljubičasta linija) imala nešto bolji rezultat. Na slici 22 su rezultati iz prethodno spomenutog rada:



Slika 22. Rezultati rekonstrukcije modela čajnika [10]

U eksperimentu nije korišteno 20 iteracija jer se za 10 iteracija predmet može odlično skenirati. Sve dalje od toga predstavlja nasumični odabir pogleda i tu nema više upotrebe algoritma. Gledajući slike 21 i 22 dolazi se do zaključka da je u ovom radu 10 iteracija bilo potrebno za sličan rezultat kao drugom radu nakon 20 iteracija. Najbolja formulacija na slici 21 ostvaruje 92.2% pokrivenosti dok na slici 22 ostvaruje 93%. Najgora formulacija na slici 21 ostvaruje 75% dok na slici 22 ostvaruje 68%.

Ono što je ipak najzanimljivije gledajući ove dvije analize je to što su se u oba rada kao najbolje pokazale različite metode. Na primjer, u ovom radu su najbolje VI temeljene na okluziji i nepoznatim vokselima sa 92.2% i 91.72% pokrivenosti. U drugom radu je prikazana suprotna situacija gdje su upravo ova dva algoritma najgora. Isto vrijedi za algoritme koji se oslanjaju na pozadinske voksele koji su se prilično dobro pokazali gledajući sliku 22.

Razlozi za ovo nisu jasno utvrđeni, ali pokazuje manu algoritama, a to je da se ne može sa sigurnošću znati kako će koji algoritam raditi za koje ponuđene poglede. U radu [10] sigurno su analizirani drugi pogledi u odnosu na ovaj eksperiment pa potpuna usporedba nije moguća. Prije korištenja bilo kojeg algoritma treba izvršiti detaljnu pripremu i analizu ponašanja na sustavu koji će biti sličan stvarnom sustavu prije skeniranja stvarnog sustava.

## 6. Zaključak

Rezultati pokazuju da su implementirane metode učinkovite za rekonstrukciju prostorno omeđenih a priori nepoznatih 3D modela objekata. Octomap biblioteka pokazala se kao fleksibilna i efikasna za mapiranje u stvarnom vremenu. Rezultati pokrivenosti objekta i vremena izvršavanja dosta su promjenjivi. Na ovo se nažalost ne može mnogo utjecati jer to ovisi o mogućnostima računala kao i o samoj Octomap biblioteci. Postoji mogućnost da se neki podaci ne stignu obraditi što utječe na idući i svaki naredni pogled, dok Octomap odlučuje hoće li nešto označiti kao okupiran vokselski ili ne. Buduća istraživanja trebaju uključivati testiranje metode na stvarnim robotskim sustavima koristeći jača računala, te proširenje na složenije scenarije rekonstrukcije kao i dodavanje mobilnog robota u sustav. Također treba analizirati odabir najboljih ponuđenih točaka te ograničenja za robota, primjerice, izbjegavanje prepreka. Algoritam implementiran u ovom radu mnogo ovisi o početnim točkama baš zbog toga što se nisu uvodila ograničenja na kretanje robota pa se može desiti da ruka padne ili da skenira samu sebe.

# Literatura

- [1] *A comparison of volumetric information gain metrics for active 3D object reconstruction*, J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza (Autonomous Robots 2017)
- [2] Wikipedia contributors, "Octree," Wikipedia, the free Encyclopedia, <https://en.wikipedia.org/wiki/Octree>, pristupljeno 25. Lipnja 2024.
- [3] Kinova Robotics, ros\_kortex, [https://github.com/Kinovarobotics/ros\\_kortex](https://github.com/Kinovarobotics/ros_kortex), pristupljeno 25. lipnja 2024.
- [4] Octomap development team, Octomap, <https://octomap.github.io/>, pristupljeno 25. Lipnja 2024.
- [5] Kinova Robotics, Gen3 Robots, <https://www.kinovarobotics.com/product/gen3-robots>, pristupljeno 25. Lipnja 2024.
- [6] Dawson-Haggerty et al, trimesh, <https://trimesh.org/>, Prosinac 2019.
- [7] GoogleResearch, Standpipe, <https://fuel.gazebosim.org/1.0/OpenRobotics/models/Standpipe>, Rujan 2023.
- [8] GoogleResearch, Melissa\_Doug\_Pound\_and\_Roll, [https://fuel.gazebosim.org/1.0/GoogleResearch/models/Melissa\\_Doug\\_Pound\\_and\\_Roll](https://fuel.gazebosim.org/1.0/GoogleResearch/models/Melissa_Doug_Pound_and_Roll), Rujan 2020.
- [9] GoogleResearch, Threshold\_Porcelain\_Teapot\_White, [https://fuel.gazebosim.org/1.0/GoogleResearch/models/Threshold\\_Porcelain\\_Teapot\\_White](https://fuel.gazebosim.org/1.0/GoogleResearch/models/Threshold_Porcelain_Teapot_White), Rujan 2020.
- [10] *An Information Gain Formulation for Active Volumetric 3D Reconstruction*, S. Isler, R. Sabzevari, J. Delmerico and D. Scaramuzza (ICRA 2016)

# Sažetak

## Aktivna rekonstrukcija 3D modela objekta primjenom robotske ruke

U ovom radu istražena je aktivna rekonstrukcija 3D modela objekta pomoću robotske ruke. Korištena je Kinova Gen 3 robotska ruka s LiDaR-om i ROS Kortex programskim paketom za upravljanje rukom. Za 3D mapiranje korištena je Octomap biblioteka. Implementirane su strategije koje odlučuju o najboljem idućem pogledu robotske ruke na temelju procjene nove volumetrijske informacije, kako bi se stvorio što detaljniji 3D model u što kraćem vremenu. Metode su testirane u simulacijskom okruženju i uspoređene na tri predmeta različitog oblika. Za različite načine računanja volumetrijskih informacija mjereno je vrijeme izvršavanja te kvaliteta konačnog modela, te je izvršena usporedba rezultata sa sličnim radom iz literature.

**Ključne riječi:** Kinova, ROS kortex, Octomap, kreiranje 3D mape voxela, odabir sljedećeg pogleda

# Summary

## **Active reconstruction of a 3D object model using a robotic arm**

In this paper, active reconstruction of a 3D model of an object using a robotic arm was investigated. Kinova Gen 3 robotic arm with LiDaR and ROS Kortex software package was used for reconstruction. The Octomap library was used for 3D mapping. Strategies have been implemented that decide on the next best view of the robotic arm based on the assessment of new volumetric information, in order to create the most detailed 3D model in the shortest possible time. The methods were tested in a simulation environment and compared on three objects of different shapes. For different ways of calculating volumetric information, the execution time and the quality of the final model were measured, and the results were compared with similar work from the literature.

**Keywords:** Kinova, ROS kortex, Octomap, 3D voxel map creation, next view selection