

# KLASIFIKACIJA OBJEKATA I ODREDIVANJE KORESPONDENCIJA DIJELOVA UNUTAR KLASA OBJEKATA NA DUBINSKIM SLIKAMA U SVRHU ROBOTSKE MANIPULACIJE

---

Đurović, Petra

Doctoral thesis / Disertacija

2020

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:538658>

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

*Download date / Datum preuzimanja:* **2024-12-24**

*Repository / Repozitorij:*

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



SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA

Petra Đurović

**KLASIFIKACIJA OBJEKATA I  
ODREĐIVANJE KORESPONDENCIJA  
DIJELOVA UNUTAR KLASA OBJEKATA  
NA DUBINSKIM SLIKAMA U SVRHU  
ROBOTSKE MANIPULACIJE**

Doktorska disertacija

Osijek, 2020.

Doktorska disertacija izrađena je na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija, Sveučilišta J. J. Strossmayera u Osijeku.

**Mentor:** Dr. sc. Robert Cupec, redoviti profesor

Disertacija ima 110 stranica.

Disertacija broj: 76

## **Naslov doktorskog rada**

- (HRV) Klasifikacija objekata i određivanje korespondencija dijelova unutar klasa objekata na dubinskim slikama u svrhu robotske manipulacije
- (ENG) Object classification and intra-class part correspondence determination in depth images for robot manipulation

## **Izvorni znanstveni doprinosi**

- Metoda vizualnog vođenja za SCARA robote koji koriste dubinsku kameru u boji kao jedini senzor.
- Metoda klasifikacije objekata na dubinskim slikama zasnovana na poravnanju konveksne ljuske.
- Metoda klasifikacije objekata na dubinskim slikama zasnovana na poliedarskim modelima klasa oblika.
- Metoda određivanja korespondencija dijelova unutar klasa objekata zasnovana na poliedarskim modelima klasa oblika.

### **Povjerenstvo za ocjenu doktorske disertacije**

- Dr. sc. Irena Galić, izvanredna profesorica, predsjednica, Sveučilište J. J. Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek
- Dr. sc. Robert Cupec, redoviti profesor, mentor, Sveučilište J. J. Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek
- Dr. sc. Ivan Petrović, redoviti profesor, član, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva Zagreb

### **Povjerenstvo za obranu doktorske disertacije**

- Dr. sc. Irena Galić, izvanredna profesorica, predsjednica, Sveučilište J. J. Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek
- Dr. sc. Robert Cupec, redoviti profesor, mentor, Sveučilište J. J. Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek
- Dr. sc. Ivan Petrović, redoviti profesor, član, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva Zagreb
- Dr. sc. Damir Filko, izvanredni profesor, član, Sveučilište J. J. Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek
- Dr. sc. Emmanuel Karlo Nyarko, izvanredni profesor, član, Sveučilište J. J. Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek

Datum obrane doktorske disertacije: 28. srpnja 2020.

# Zahvale

Zahvaljujem mentoru, prof.dr.sc. Robertu Cupecu, na povjerenju i neizmjerne pomoći i podršci tijekom doktorskog studija. Hvala na svakodnevnim smjericama, idejama, motivaciji i inspiraciji. Hvala na svim savjetima tijekom provedbe istraživanja i pisanja disertacije te na ukazanoj prilici za predstavljanje i prezentaciju objavljenih radova. Bila mi je iznimna čast raditi pod Vašim mentorstvom te se veselim budućoj suradnji!

Istraživački rad koji je rezultirao ovom doktorskom disertacijom financirala je Hrvatska zaklada za znanost projektom Napredna 3D percepcija za mobilne robotske manipulatore IP-2014-09-3155 pod vodstvom prof.dr.sc. Roberta Cupeca.

Doktorski studij financirala je Hrvatska zaklada za znanost projektom Plan razvoja karijera mladih istraživača – izobrazba novih doktora znanosti DOK-10-2015.

# Sadržaj

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Uvod</b>   | <b>1</b>  |
| <b>2</b> | <b>Vizualno vođenje za SCARA robote koji koriste dubinsku kameru u boji kao jedini senzor</b> | <b>5</b>  |
| 2.1      | Uvod . . . . .  | 6         |
| 2.2      | Istraživanja u ovom području . . . . .  | 8         |
| 2.3      | Pozicioniranje alata pomoću RGB-D kamere . . . . .  | 10        |
| 2.3.1    | Određivanje položaja markera RGB-D kamerom . . . . .  | 10        |
| 2.3.2    | Vizualno vođenje . . . . .  | 12        |
| 2.3.3    | <i>Hand-eye</i> umjeravanje za relativno pozicioniranje . . . . .                             | 15        |
| 2.4      | Planiranje hvatanja jednostavnih objekata pomoću računalnog vida . . . . .                    | 17        |
| 2.5      | Eksperimentalna evaluacija . . . . .  | 20        |
| 2.5.1    | Eksperimentalni postav . . . . .  | 21        |
| 2.5.2    | Pokusi vizualnog vođenja . . . . .  | 22        |
| 2.5.3    | Pokusi hvatanja . . . . .   | 23        |
| 2.6      | Zaključak . . . . .   | 25        |
| <b>3</b> | <b>Klasifikacija objekata na dubinskim slikama zasnovana na poravnanju konveksne ljuske</b>   | <b>26</b> |
| 3.1      | Uvod . . . . .  | 26        |
| 3.2      | Istraživanja u ovom području . . . . .  | 29        |
| 3.3      | Poravnanje konveksnih ljuski . . . . .  | 30        |
| 3.3.1    | Poravnanje sličnih oblika . . . . .   | 30        |
| 3.3.2    | Sličnost konveksnih ljuski . . . . .  | 31        |

|          |   |           |
|----------|---|-----------|
| 3.3.3    | Optimalna translacija i skala . . . . .   | 32        |
| 3.3.4    | Optimalna rotacija . . . . .  | 34        |
| 3.3.5    | Približno poliedarski prikaz . . . . .  | 37        |
| 3.4      | Klasifikacija objekata . . . . .  | 37        |
| 3.5      | Eksperimentalna evaluacija . . . . .  | 41        |
| 3.6      | Zaključak . . . . .   | 43        |
| <b>4</b> | <b>Klasifikacija objekata na dubinskim slikama zasnovana na poliedarskim modelima klasa oblika</b>                | <b>45</b> |
| 4.1      | Uvod . . . . .  | 46        |
| 4.2      | Istraživanja u ovom području . . . . .  | 46        |
| 4.3      | Model klase oblika . . . . .  | 47        |
| 4.3.1    | VolumeNet . . . . .   | 47        |
| 4.3.2    | Prostor oblika . . . . .  | 49        |
| 4.3.3    | Klasa oblika kao konveksni podskup prostora oblika . . . . .  | 52        |
| 4.4      | Mjere sličnosti oblika . . . . .  | 53        |
| 4.5      | Prilagodba VN modela oblaku točaka . . . . .  | 55        |
| 4.6      | Planiranje hvatanja na temelju VN modela . . . . .  | 57        |
| 4.7      | Određivanje VN strukture i deskriptora na osnovi skupa za učenje . . . . .  | 58        |
| 4.7.1    | Poravnanje uzoraka iz skupa za učenje s referentnim modelom . . . . .   | 59        |
| 4.7.2    | Optimalna VolumeNet struktura . . . . .   | 59        |
| 4.7.3    | Metamodel . . . . .   | 60        |
| 4.7.4    | Segmentacija 3D modela na konveksne, konkavne i toroidalne površine i njihovo pridruživanje . . . . .             | 61        |
| 4.7.5    | Odbacivanje irelevantnih stranica . . . . .   | 63        |
| 4.7.6    | Računanje deskriptora . . . . .   | 64        |
| 4.8      | Eksperimentalna evaluacija . . . . .  | 65        |
| 4.9      | Zaključak . . . . .   | 67        |
| <b>5</b> | <b>Određivanje korespondencija dijelova unutar klasa objekata zasnovano na poliedarskim modelima klasa oblika</b> | <b>69</b> |
| 5.1      | Uvod . . . . .  | 70        |
| 5.2      | Istraživanja u ovom području . . . . .  | 72        |



---

|          |  |            |
|----------|--|------------|
| 5.3      | Pregled predložene metode . . . . .                      | 74         |
| 5.3.1    | Detekcija komponenti . . . . .                           | 74         |
| 5.3.2    | Graf pridruživanja komponenti (CAG) . . . . .            | 76         |
| 5.4      | Vjerojatnost semantičke korespondencije . . . . .        | 76         |
| 5.4.1    | Sličnosti oblika, veličine i pozicije segmenta . . . . . | 77         |
| 5.4.2    | Sličnost susjedstva . . . . .                            | 78         |
| 5.5      | Pridruživanje segmenta i ciljne komponente . . . . .     | 81         |
| 5.6      | Izbor reprezentativnih objekata . . . . .                | 82         |
| 5.7      | Eksperimentalna evaluacija . . . . .                     | 83         |
| 5.7.1    | Rezultati . . . . .                                      | 86         |
| 5.7.2    | Diskusija rezultata . . . . .                            | 89         |
| 5.8      | Zaključak . . . . .                                      | 93         |
| <b>6</b> | <b>Zaključak</b>   | <b>95</b>  |
|          | <b>Literatura</b>  | <b>98</b>  |
|          | <b>Sažetak</b>   | <b>107</b> |
|          | <b>Abstract</b>  | <b>109</b> |
|          | <b>Životopis</b>   | <b>i</b>   |
|          | <b>Popis slika</b>                                       | <b>ii</b>  |
|          | <b>Popis tablica</b>                                     | <b>iv</b>  |

# 1

## Uvod

Pojava kvalitetnih, a cjenovno dostupnih robotskih manipulatora, senzora poput dubinskih kamera te hvataljki i mobilnih platformi, uzrokovala je prijelaz fokusa istraživanja i razvoja robotike sa područja klasične industrijske automatizacije na područje umjetne inteligencije u cilju razvoja autonomnih robota koji se samostalno lokaliziraju, kreću u nestrukturiranim okolinama, pritom izbjegavaju prepreke te obavljaju razne zadatke koji uključuju manipulaciju poznatim, ali i po prvi put viđenim objektima. Roboti takvih mogućnosti mogu, primjerice, asistirati u kućanstvima ili ustanovama poput bolnica i škola.

Uspješna redukcija cijene robotskih manipulatora može se postići uklanjanjem skupih enkodera iz zglobova, uslijed čega se, međutim, gubi informacija o položaju pojedinog članka i vrha alata robotske ruke. Uporabom drugih senzora manjih cijena, poput RGB-D kamera i implementacijom softverskih rješenja moguće je umjeravati robotsku ruku te robotsku manipulaciju planirati i nadzirati vizualnim vođenjem, na temelju vizualnih obilježja robotske ruke. RGB-D kamere, analogno očima čovjeka, osim boje, pružaju i trodimenzionalnu percepciju okoline robota te se primjenom naprednih algoritamskih rješenja za obradu dubinske slike uspješno analizira scena i dobiva informacija o položaju objekata na sceni. Informacija o položaju i međusobnim odnosima objekata može poslužiti za lokalizaciju robota, planiranje kretanja, planiranje hvatanja i manipulacije itd.

Preduvjet uspješne manipulacije objektom od interesa je njegova detekcija na sceni, određivanje njegovog relativnog položaja u odnosu na robota te određivanje točki hvatanja.

S obzirom da se autonomni robot kreće u nestrukturiranim okolinama, osim poznatih objekata, trebao bi biti u mogućnosti prepoznavati i prvi puta viđene objekte te ih, na temelju dovoljno sličnih značajki, svrstati u kategoriju s poznatim objektima. Radna operacija poput hvatanja bitno ovisi o svojstvima i dimenzijama hvataljke te dimenzijama predmeta, točnije dijela koji je predviđen za hvatanje. Zbog toga je bitno da robot bude u stanju prepoznati dijelove objekata relevantne za obavljanje zadane radne operacije. Da bi se neka radnja mogla definirati ne samo za točno određeni objekt već za veće skupove objekata, potrebno je definirati klase objekata. Klase objekata su skupovi objekata koji imaju neka zajednička svojstva ili svrhu, kao što su npr. posude, ormari, svjetiljke, boce, stolovi, stolice itd. Inteligentni sustavi prepoznaju i klasificiraju objekte na temelju modela izgrađenih postupkom učenja. Učenje robota se svodi na pokazivanje modela objekata pripadnika određenih klasa pri čemu je svakom modelu pridružena oznaka pripadnosti odgovarajućoj klasi. Učenjem, robot stvara bazu poznatih objekata. S obzirom da je klasificirani objekt sličan, ali ne identičan robotu poznatim objektima, njegovi su dijelovi približno jednakih dimenzija, oblika i smještaja na objektu, ali također je moguće da neki dijelovi nisu prisutni na određenom pripadniku neke klase ili da se uvelike razlikuju. Segmentacija objekta na dijelove i korespondencija dijelova objekata unutar iste klase omogućava da se radna operacija planirana za određeni dio referentnog objekta neke klase, prenese na ostale pripadnike te klase, uz potrebnu prilagodbu dimenzijama i orijentaciji svakog pojedinog objekta.

Vizualno vođenje, klasifikacija objekata i semantička segmentacija aktualne su teme u najnovijim istraživanjima na području robotike, umjetne inteligencije i računalnog vida. Vizualno vođenje spominje se u brojnim istraživanjima, ali često je namijenjeno industrijskim robotima, koji ni cjenovno niti dimenzijama nisu prikladni za svakodnevnu primjenu u kućanstvima. Primjena računalnog vida za robotsku manipulaciju podrazumijeva umjereni sustav kamere i robotske ruke, tj. poznavanje točnog položaja kamere u odnosu na robotsku ruku. Kada robot, kakav se razmatra u ovom doktorskom radu, traži objekte od interesa u svojoj okolini, kamera neprestano mijenja smjer te je potrebno često ponovno umjeravanje. Stoga je nužan brz i jednostavan proces umjeravanja. U nekim je predloženim metodama postupak umjeravanja dugotrajan i nerijetko zahtijeva kalibracijske panele, što je također neprikladno za često automatizirano umjeravanje u stvarnom vremenu.

Metode za klasifikaciju objekata na dubinskim slikama često se temelje na usporedbi lokalnih ili globalnih deskriptora značajki oblika te na primjeni dubokih neuronskih mreža.

Iako postižu relativno visoku točnost klasifikacije, postojeće metode uglavnom ne daju informaciju o položaju objekata na sceni.

Semantička segmentacija dijelova objekata te estimacija položaja dijelova objekata relativno je nov problem, ali pojava prikladnih označenih skupova 3D modela namijenjenih evaluaciji metoda prepoznavanja dijelova potiče i ukazuje na potrebu razvoja takvih metoda. Postojeće metode za semantičku segmentaciju zahtijevaju velike skupove za učenje, za čije je ručno označavanje potrebno uložiti puno napora i vremena.

U ovom se radu predlaže razvoj četiri metode, koje su opisane u četiri poglavlja. U poglavlju 2 predstavljena je metoda vizualnog vođenja za SCARA robote koji koriste RGB-D kameru kao jedini senzor. Opisana metoda rješava problem navođenja robotske ruke iznad točke od interesa. Umjeravanje ruke i kamere zasniva se na praćenju markera postavljenog u blizini vrha alata te na detekciji dominantne horizontalne ravnine, često prisutne u unutarnjim prostorima, na temelju koje se iterativno računaju potrebne promjene u zglobovima robotske ruke kako bi vrh alata došao u željenu poziciju, npr. iznad dijela objekta kojeg treba uhvatiti. Zbog svoje jednostavnosti, ova je metoda pogodna za često umjeravanje. Eksperimentalna evaluacija razvijene metode provodi se na stvarnom SCARA robotu nizom eksperimenata navođenja robotske ruke iznad točke ili objekta od interesa te mjerenjem točnosti pozicioniranja kao i pokusima hvatanja jednostavnih predmeta.

Poglavlje 3 opisuje novu metodu klasifikacije objekata na dubinskim slikama zasnovanu na poravnanju konveksne ljuske. Ova metoda podrazumijeva da su objekti pripadnici iste klase sličnog oblika te pronalazi optimalno poravnanje konveksne ljuske modela sa scene, kojeg se klasificira, s konveksnim ljuskama objekata u bazi modela. Klasa najbližijeg modela iz baze dodjeljuje se objektu na sceni. Ova metoda, osim klase, vraća i estimirani položaj objekta na sceni što omogućava planiranje hvatanja.

Metoda klasifikacije objekata na dubinskim slikama zasnovana na poliedarskim modelima klasa oblika, predstavljena u poglavlju 4, zasniva se na VolumeNet modelu klasa 3D oblika. U ovom je poglavlju predložena metoda učenja, tj. stvaranja VN modela na osnovu skupa uzoraka za učenje. Informacija o položaju, kao i parametri oblika omogućavaju planiranje hvatanja robotskim manipulatorom. Obje metode klasifikacije eksperimentalno su ispitane na referentnim skupovima RGB-D slika objekata te je uspješnost ispravnog klasificiranja uspoređena s drugim metodama klasifikacije.

Poglavlje 5 opisuje metodu određivanja korespondencija dijelova unutar klasa objekata

zasnovanu na poliedarskim modelima klasa oblika. Osnovna ideja predložene metode jest pridruživanje semantičke oznake dijelovima objekata koji pripadaju istoj klasi, na temelju malog broja označenih reprezentativnih objekata. Informacija o vrsti dijela objekta, njegovom položaju i dimenzijama omogućava planiranje akcije hvatanja - analogno hvatanju referentnog oblika. Evaluacija uspješnosti ispitana je na referentnom skupu objekata s označenim dijelovima i uspoređena s postojećim metodama.

U ovom se radu, dakle, razvijaju metode koje bi, integrirane u jedan sustav, zajedno s hardverskim dijelovima – robotskom rukom i kamerom, činile funkcionalnu cjelinu. Tako razvijeni robotski sustav mogao bi klasificirati objekt na sceni snimljenoj RGB-D kamerom, estimirati njegov položaj te ga dijeliti na semantički smislene cjeline. Potom bi dijelu predviđenom za hvatanje mogao odrediti položaj i točke u kojima je potrebno postaviti hvataljku robotske ruke. Naposljetku, vizualnim bi se vođenjem usmjeravala robotska ruka te bi se izvršavalo hvatanje. Iako realizacija i eksperimentalna evaluacija takvog cjelovitog robotskog sustava nije obuhvaćena ovim radom, predložene metode daju doprinos rješavanju pojedinih problema koji su nezaobilazni kod realizacije takvih sustava.

# 2

## Vizualno vođenje za SCARA robote koji koriste dubinsku kameru u boji kao jedini senzor

Prvi je doprinos ovog doktorskog rada metoda vizualnog vođenja za SCARA robote koji koriste dubinsku kameru u boji kao jedini senzor. Ovaj je doprinos objavljen u časopisnom članku (Đurović et al., 2017a), a ovo poglavlje disertacije opisuje razvijenu metodu vizualnog vođenja s jednostavnim *hand-eye* umjeravanjem u dva koraka namijenjenu robotskim sustavima u SCARA konfiguraciji. Opisana je i jednostavna metoda planiranja hvatanja zasnovana na računalnom vidu. Opisane su metode dizajnirane za robote niske cijene bez enkodera i drugih propriocepcijskih senzora, upravljane kamerom, pri čemu je pozicioniranje alata ostvareno vizualnim vođenjem zasnovanim na praćenju markera i informacije o 3D geometriji scene dobivene RGB-D kamerom. Umjeravanje se zasniva na identifikaciji dominantne horizontalne ravnine u vidnom polju kamere i na pretpostavci da su sve osi robota okomite na tu ravninu. Osim parametara ravnine, potreban je samo jedan rotacijski pokret zglobov ramena kako bi se provelo umjeravanje između kamere i robotske ruke te omogućilo vizualno vođenje. Planiranje hvatanja zasniva se na graničnim okvirima (engl. *bounding box*) detektiranih objekata na RGB-D slici, koji pružaju dostatnu informaciju za pozicioniranje robotske hvataljke iznad predmeta, pravilnu orijentaciju hvataljke te širinu njezina otvora. Metode su eksperimentalno ispitane na pravoj robotskoj ruci te je na temelju mjerenja točnosti pozicioniranja i uspješnosti hvatanja analizirana njihova učinkovitost.

## 2.1 Uvod

Usprkos stalnom godišnjem rastu proizvodnje robota, većina robota i dalje se koristi u industriji ([ifr, Pristup: 2020-06-01](#)). Stoga su mnogi istraživački timovi ([Kim et al., 2014](#); [Larouche i Zhu, 2014](#); [Đurović et al., 2017b](#)) bili motivirani za pronalaženje robotskih rješenja niskih cijena kako bi učinili robote pristupačnijima u svakodnevnoj uporabi. Istraživanje predstavljeno u ovom poglavlju doprinosi tome cilju predlaganjem metode za vizualno upravljanje robotskom rukom niske cijene. Predstavljeno istraživanje obuhvaća umjeravanje robotske ruke i kamere (engl. *hand-eye calibration*) u svrhu vizualnog vođenja (engl. *visual servoing*) i hvatanja (engl. *grasping*) jednostavnih konveksnih objekata.

Razmatra se robotska ruka niske cijene, koja se zasniva na koračnim motorima bez apsolutnih enkodera ili drugih propriocepcijskih senzora za mjerenje kutova zglobova, koja se oslanja samo na vizualnu informaciju. Nedostatak apsolutnih enkodera podrazumijeva da pozicioniranje alata nije moguće primjenom inverzne kinematike jer apsolutni kutevi zglobova nisu poznati. Stoga se predlaže relativno pozicioniranje alata zasnovano na računalnom vidu. Odredišna pozicija alata određena je detekcijom objekta od interesa na slici robotskog radnog prostora snimljenoj RGB-D kamerom, dok je trenutna pozicija alata određena lokalizacijom markera, montiranog pri vrhu alata, softverom za praćenje markera. Predložena metoda vizualnog vođenja računa promjene u zglobovima potrebne kako bi se alat premjestio iz trenutne u odredišnu poziciju. Metoda je dizajnirana za robote u SCARA konfiguraciji, često korištene u robotskoj manipulaciji, osobito za zadatke sastavljanja ili premještanja (engl. *pick-and-place*) ([Craig, 2005](#)). U takvoj konfiguraciji, trenutna i odredišna pozicija predstavljene su točkama na dvijema kružnicama različitih polumjera, sa središtem u osi ramenog zgloba. Udaljenost između alata i osi ramenog zgloba podešava se promjenom kuta u zglobu lakta, dok se promjenom kuta ramenog zgloba alat pomiče po kružnici dok ne dosegne željenu odredišnu poziciju. Zbog nesavršenosti motora u zglobovima robotske ruke i nesigurnosti određivanja položaja osi ramenog zgloba u odnosu na kameru, pozicioniranje alata iterativno se ponavlja dok se ne postigne željena pozicija unutar zadane tolerancije, što se mjeri sustavom robotskog vida.

Položaj osi ramenog zgloba u odnosu na kameru, koji zahtijeva predložena metoda vizualnog vođenja, ostvaren je novom, jednostavnom i brzom *hand-eye* metodom umjeravanja u dva koraka. S obzirom da je metoda vizualnog vođenja zasnovana na udaljenosti alata od osi ramenog zgloba i njegovu poziciju na kružnici sa središtem u toj osi, kamera je smještena

u konfiguraciji *eye-to-hand*, pri čemu je kamera postavljena fiksno u odnosu na os ramenog zgloba. Takva je konfiguracija prikladna za manje robotske ruke na kojima je zbog dimenzija ruke i kamere nezgodno montirati kameru pri vrhu alata. Osim toga, ova je konfiguracija tipična za antropomorfne robote kao i za biološke sustave poput čovjeka ili životinja, koji također imaju sustav vizije smješten visoko iznad zemlje što omogućava široki pregled okoline. Iz istog je razloga ovakva konfiguracija prikladna za robotske manipulatore koji koriste kameru za lokalizaciju, detekciju objekata od interesa u okolini robota i manipulaciju tim objektima.

U razmatranoj SCARA konfiguraciji robota sve su osi zglobova paralelne gravitacijskoj osi te se pretpostavlja da objekti od interesa leže na horizontalnoj ravnini, koja se u ovoj disertaciji naziva potporna ravnina (engl *supporting plane*). Predložena *hand-eye* metoda umjeravanja identificira potpornu ravninu u vidnom području kamere te određuje orijentaciju osi ramenog zgloba kao vektor okomit na tu ravninu. Osim informacija o potpornoj ravnini, predložena metoda umjeravanja zahtijeva samo jedan rotacijski pokret ramenim zglibom. Uz pretpostavku da se kut zgloba lakta ne mijenja, rameni zglob pomiče alat duž kružnice sa središtem u osi ramenog zgloba. Uz poznatu promjenu kuta ramenog zgloba, može se odrediti središte te kružnice u odnosu na kameru. Os ramenog zgloba određuje se kao pravac okomit na potpornu ravninu, koji prolazi kroz središte kružnice. Jednostavnost ove metode čini je pogodnom za često umjeravanje.

Kako bi se ostvario cjelokupni sustav za pozicioniranje alata SCARA robota zasnovan na računalnom vidu, razvijena je i metoda za planiranje hvatanja. Predložena metoda planiranja hvatanja sastoji se od detekcije jednostavnih konveksnih objekata na RGB-D snimci scene, kreiranja graničnog okvira tih objekata te računanja određene pozicije alata te orijentacije i širine otvora hvataljke na temelju parametara graničnih okvira. Orijehtacija alata određena je na temelju oblika i orijentacije objekta od interesa. Točnije, najmanje zakrivljeno područje površine objekta, približno okomito na potpornu ravninu smatra se prikladnim mjestom hvata. Područje male zakrivljenosti površine objekta detektirano je segmentacijom te površine na približno ravninske segmente. Planiranje hvatanja predloženo u ovom radu ima nekoliko nedostataka: (i) pretpostavlja se da alat ima samo jedan stupanj slobode, rotaciju, te da se predmeti hvataju odozgo, (ii) moguće je hvatanje samo konveksnih objekata i (iii) ni u slučaju konveksnih objekata, stabilan hvat nije garantiran. Međutim, klasa objekata na koje se predložena metoda može primijeniti je i dalje vrlo rasprostranjena te je



jasna njezina prednost u odnosu na druge, općenitije, ali složenije metode koje su opisane u poglavlju 2.2.

Dvije su metode opisane u ovom poglavlju. Prva je nova metoda vizualnog vođenja za SCARA robote bez apsolutnih enkodera koji koriste kameru kao jedini senzor. Metoda koristi informaciju o osi ramenog zgloba koja je dobivena jednostavnom metodom *hand-eye* umjeravanja. Druga je metoda za planiranje hvatanja, prikladna za jednostavne konveksne objekte detektirane na RGB-D slikama. Predložene metode su eksperimentalno evaluirane nizom pokusa vizualnog vođenja i hvatanja objekata vizualno upravljanom robotskom rukom niske cijene u SCARA konfiguraciji.

Struktura ovog poglavlja je sljedeća. U potpoglavlju 2.2 dan je pregled istraživanja iz područja robotike niske cijene, vizualnog vođenja, *hand-eye* umjeravanja i planiranja hvatanja. Predložena metoda vizualnog vođenja opisana je u potpoglavlju 2.3, a metoda planiranja hvatanja u potpoglavlju 2.4. Eksperimentalna evaluacija i rezultati izneseni su u potpoglavlju 2.5. Zaključak je napisan u potpoglavlju 2.6.

## 2.2 Istraživanja u ovom području

U ovom je potpoglavlju dan pregled objavljenih radova iz područja cjenovno dostupne (engl. *low-cost*) robotike, *hand-eye* umjeravanja, vizualnog vođenja i planiranja hvatanja zasnovanog na računalnom vidu.

**Cjenovno dostupni roboti upravljani vizijom:** Cjenovno dostupan, samostalno izrađen, Pieper robotski manipulator sa šest stupnjeva slobode upravljani kamerom u *eye-in-hand* konfiguraciji predložen je u radu (Larouche i Zhu, 2014). U članku (Đurović et al., 2017b), predstavljen je robotski sustav upravljani kamerom koji se sastoji od cjenovno dostupnog robotskog manipulatora s 4 stupnja slobode i RGB-D kamere. Sustav nema enkodere te se zasniva na vizualnom vođenju kojem prethodi umjeravanje ruke i kamere u dva jednostavna koraka.

**Umjeravanje robotske ruke i kamere i vizualno vođenje:** U radovima (Larouche i Zhu, 2014; van Delden i Hardy; Lippiello et al., 2005; Qiao et al., 2014; Xu et al., 2008; Miseikis et al., 2016; Heller et al., 2014; Strobl i Hirzinger, 2006; Boby i Saha, 2016; Freeman et al., 2007) predložene su metode umjeravanja robotske ruke i kamere u *eye-in-hand* konfiguraciji. U radu (Qiao et al., 2014), predložena je *eye-in-hand* metoda umjeravanja koja se sastoji od malog broja koraka. Laser montiran pri vrhu alata projicira svjetlosnu

ravninu za potrebe umjeravanja. Metoda koja istovremeno provodi *eye-in-hand* umjeravanje kao i umjeravanje intrinzičnih parametara kamere u isto vrijeme predložena je u (Xu et al., 2008). Ova metoda ovisi o fiksnoj točki koja mora biti vidljiva tijekom umjeravanja. Točka je smještena u radni prostor robota, a ne na samog robota. U mobilnoj robotskoj manipulaciji, ovo ograničenje čini metodu neprikladnom za često umjeravanje jer robot mijenja položaj i na taj način se mijenja i njegov radni prostor. Osim toga, ova metoda zahtijeva najmanje sedam pokreta robotskom rukom, što je sporo i neprikladno za često umjeravanje u stvarnom vremenu. Slično tomu, metoda koja ujedno provodi umjeravanje ruke i kamere u *eye-to-hand* konfiguraciji i određuje intrinzične parametre kamere predložena je u (Miseikis et al., 2016). Metoda pritom koristi kalibracijski panel, šahovsku ploču A4 formata koja je postavljena pri vrhu alata. Zbog veličine panela, ova metoda nije prikladna za manje robotske ruke i često automatsko umjeravanje. Složenija *eye-in-hand* metoda umjeravanja, koja koristi globalnu polinomsku optimizaciju predložena je u (Heller et al., 2014). Ova metoda zahtijeva kalibracijski panel u radnom prostoru robota, što je neprikladno za mobilne robote koji imaju potrebu za čestim umjeravanjem. U (Larouche i Zhu, 2014) implementirano je vizualno vođenje samostalno izrađene robotske ruke. Razmatrana je *eye-in-hand* konfiguracija, a određeni položaj alata određen je na 2D slici fotogrametrijom. Vizualno vođenje za apsolutno pozicioniranje predloženo je u (Đurović et al., 2017b). Umjeravanje ruke i kamere sastoji se od dva rotacijska pokreta koji prolaze kroz tri točke koje formiraju trokut dostatan za izračun položaja koordinatnog sustava robota u odnosu na koordinatni sustav kamere. Međutim, u ovom je postupku teško postići visoku točnost zbog malog broja mjerenja tijekom izračuna. Osim toga, ova metoda zahtijeva da se robot nalazi u određenom inicijalnom položaju prije provođenja umjeravanja ruke i kamere, a to nije moguće automatizirati bez apsolutnih enkodera.

**Planiranje hvatanja:** Neka istraživanja određuju poziciju objekta za hvatanje na 2D slici, na temelju kontura objekta (Speth et al., 2008; Xu et al., 2008). Osim stereo kamere, (Marton et al., 2010) koristi laserske skenere koji određuju 3D poziciju objekta na sceni. Informacija o 3D geometriji scene je korisna u robotskoj manipulaciji, što je i dokazano rezultatima istraživanja u području računalnog vida (Aldoma et al., 2012; Papazov i Burschka, 2010; Hinterstoisser et al., 2012; Mueller et al., 2014; Detry et al., 2013; Twardon i Ritter, 2015). Kako bi se postigle performanse u stvarnom vremenu, (Kragic et al., 2001) uvode bazu objekata koje je moguće uhvatiti te se *off-line* računaju sva uspješna hvatanja za

određeni objekt. Taj je objekt predstavljen CAD modelom ili 3D skenom objekta. U radu (Lopez-Damian et al., 2005), predstavljena je implementacija planiranja hvatanja nekonvexnih objekata dekompozicijom objekata unutar alata za planiranje kretanja (Simeon et al., 2001). Metoda koja planira hvatanje bazirana na graničnim okvirima predmeta predstavljena je u (Xue i Dillmann, 2011). Skup graničnih okvira različitih svojstava poput veličine i orijentacije stvoren je *off-line* i spremljen u bazu. Za zadani granični okvir objekta detektiranog na sceni, sličan se granični okvir pronalazi u bazi te se, ako je to moguće, provodi hvatanje s unaprijed postavljenom konfiguracijom hvataljke. Optimizacija konfiguracije hvataljke upotrijebljena je u metodi planiranja hvatanja koju predstavljaju (Borst et al., 2002). U slučaju kompleksnih zadataka planiranja hvatanja, prijedlozi hvatanja evaluirani su unutar simulatora, od kojih je jedan od najpoznatijih Graspit! (Miller i Allen, 2004) te ga koriste (Kragic et al., 2001; Xue i Dillmann, 2011; Marton et al., 2010). U radu (Kragic et al., 2001), Graspit! se koristi u integraciji vizije u stvarnom vremenu i *online* planiranja hvatanja. Nakon što se izvede stabilno hvatanje objekta, prati se trajektorija uhvaćenog objekta tijekom njegova kretanja do ciljnog mjesta. U odnosu na navedene metode, prednosti metode predložene u ovom poglavlju ovog doktorskog rada jesu jednostavnost, brzina i računaska učinkovitost.

## 2.3 Pozicioniranje alata pomoću RGB-D kamere

U ovom je poglavlju opisana predložena metoda pozicioniranja vrha robotskog alata temeljena na vizualnom vođenju kojem prethodi hand-eye umjeravanje. Predloženi koncept sastoji se od određivanja trenutnog i odredišnog položaja vrha alata u odnosu na os koja prolazi ramenim zglobovom robota na temelju računalnog vida te promjene kuta zglobova dok ne dođe do preklapanja tih dvaju položaja unutar zadane tolerancije.

### 2.3.1 Određivanje položaja markera RGB-D kamerom

Razvijene metode dizajnirane su za robotski manipulator na čijoj je ruci smješten marker. Središte markera služi za određivanje trenutnog položaja vrha alata robota. S obzirom da je u razmatranom robotskom sustavu korištena RGB-D kamera, dostupne su dvije vrste mjerenja udaljenosti središta markera i kamere: udaljenost izmjerena na RGB slici pomoću softvera za praćenje markera te udaljenost izmjerena dubinskim senzorom. Softver za praćenje markera detektira marker na RGB slici te računa njegov položaj u odnosu na korodinatni sustav

kamere na temelju veličine markera na slici i stvarne unaprijed zadane veličine markera. Ovaj je položaj definiran koordinatama  $(x_{RGB}, y_{RGB}, z_{RGB})$ . Dubinski senzor RGB-D kamere dodjeljuje vrijednost dubine točkama na slici. Vrijednost dubine točke na slici je vrijednost njezine  $z$  koordinate u koordinatnom sustavu kamere, pri čemu je  $z$ -os tog koordinatnog sustava paralelna s optičkom osi kamere. Vrijednost dubine  $z_d$  dodijeljena točki na slici koja predstavlja centar markera jest alternativna mjera  $z$  koordinate centra markera. U ovoj je metodi provedena fuzija dviju mjera  $z$  koordinate centra markera,  $z_{RGB}$  i  $z_d$ , uzimajući u obzir nesigurnost oba mjerenja.

Kako bi se estimirala nesigurnost softvera za praćenje markera, marker se pomiče uzduž vertikalne linije pokretanjem translacijskog zgloba robota. Pravac regresije ortogonalne udaljenosti (engl. *orthogonal distance regression*, ODR) aproksimiran je skupom točaka koje predstavljaju položaje centra markera mjerene softverom za praćenje markera. Za svaki položaj centra markera, razlika između njegove  $z$  koordinate i  $z$  koordinate točke, na optičkoj osi koja prolazi kroz centar markera, najbliže ODR pravcu, predstavlja pogrešku mjerenja koordinate  $z_{RGB}$ . Varijanca ove pogreške,  $\sigma_{RGB}^2$ , predstavlja nesigurnost mjerenja koordinate  $z_{RGB}$ .

U svrhu estimacije nesigurnosti koordinate  $z_d$ , u vidno polje kamere smještena je radna površina te su točke koje pripadaju toj ravnini detektirane na dubinskoj slici. Nesigurnost koordinate  $z_d$  estimirana je statističkom analizom devijacije tih točaka u odnosu na ravninu aproksimiranu tim točkama metodom najmanjih kvadrata. Razmatrana ravnina identificirana je standardnim RANSAC algoritmom (Fischler i Bolles, 1981). Svaka točka te ravnine, projicirana je na ravninu određenu metodom najmanjih kvadrata duž optičke osi koja prolazi tom točkom. Razlika između  $z$  koordinata svake točke i njezine projekcije predstavlja pogrešku mjerenja. Varijanca tih mjerenja,  $\sigma_d^2$ , predstavlja mjeru nesigurnosti koordinate  $z_d$ . Postupke određivanja varijanci  $\sigma_{RGB}^2$  i  $\sigma_d^2$  dovoljno je provesti samo jednom.

Optimalna  $z$  koordinata računa se pomoću mjerenja  $z_{RGB}$  i  $z_d$  te njihovih varijanci  $\sigma_{RGB}^2$  i  $\sigma_d^2$ , na sljedeći način

$$z = \frac{\frac{z_d}{\sigma_d^2} + \frac{z_{RGB}}{\sigma_{RGB}^2}}{\frac{1}{\sigma_d^2} + \frac{1}{\sigma_{RGB}^2}}. \quad (2.1)$$

Ova se  $z$  koordinata koristi pri korekciji drugih dviju koordinata centra markera, skaliranjem koordinata skalirajućim faktorom

$$s = \frac{z}{z_{RGB}}. \quad (2.2)$$

Uz pretpostavku da je marker potpuno vidljiv, opisana metoda računa 3D poziciju markera u odnosu na koordinatni sustav kamere označen s  ${}^C p_M$ . Izračun 3D pozicije markera opisan je sljedećim koracima.

1. RGB-D kamera snima sliku.
2. Centar markera identificiran je na RGB slici odgovarajućim programom računalnog vida koji daje njegove  $x_{RGB}$ ,  $y_{RGB}$  i  $z_{RGB}$  koordinate.
3. Koordinate centra markera na dubinskoj slici su određene pomoću koordinata na RGB slici. Kako bi se smanjio utjecaj mjernog šuma izračunava se vrijednost prosječne dubine susjednih točaka kvadrata oko centra markera te se  $z_d$  koordinata centra markera u odnosu na kameru zamjenjuje tom vrijednošću.
4. Pomoću jednadžbe (2.2) računa se faktor skaliranja,  $s$ .
5. Konačna 3D pozicija markera računa se množenjem svake koordinate  $x$ ,  $y$  i  $z$ , izračunate u koraku 2, s faktorom skaliranja  $s$ .

### 2.3.2 Vizualno vođenje

U nastavku rada, koriste se sljedeće oznake:  ${}^B t_A$  predstavlja translacijski vektor koji definira položaj koordinatnog sustava  $S_A$  u odnosu na koordinatni sustav  $S_B$ . Položaj točke  $A$  u odnosu na koordinatni sustav  $B$  označen je s  ${}^B p_A$ . Nadalje, u ovom se poglavlju koordinatni sustav robota označava s  $R$ , kamere sa  $C$ , markera s  $M$ , a  $G$  predstavlja koordinatni sustav graničnog okvira objekta.

Postoje dvije uobičajene konfiguracije SCARA robota: jedna u kojoj je prvi zglob translacijski, a druga dva rotacijska i druga konfiguracija u kojoj su prva dva zgloba rotacijska, a treći translacijski. U ovom se radu razmatra prva konfiguracija, ali ista je metoda primjenjiva na obje konfiguracije. Svrha vizualnog vođenja je izračun potrebnih promjena kuteva zglobova kako bi se marker iz trenutnog položaja postavio u željeni odredišni položaj. Trenutni je položaj markera određen mjerenjem opisanim u potpoglavlju 2.3.1, a odredišni je položaj određen metodom za planiranje hvatanja opisanom u poglavlju 2.4.

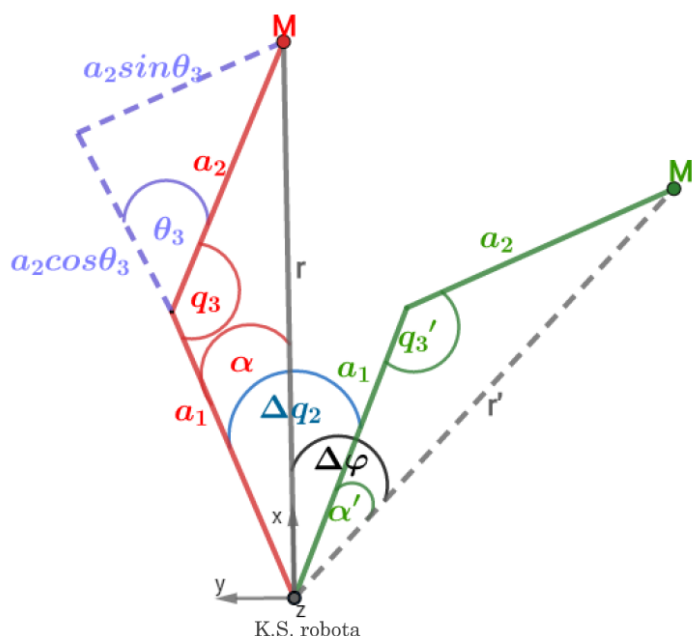
Predložena metoda vizualnog vođenja zahtijeva položaj osi ramenog zgloba u odnosu na koordinatni sustav kamere definiran jediničnim vektorom  ${}^C z_R$ , koji predstavlja orijentaciju

osi, i vektorom  ${}^C t_R$ , koji predstavlja poziciju referentne točke na promatranoj osi u odnosu na koordinatni sustav kamere.

Vektori  ${}^C z_R$  i  ${}^C t_R$  određeni su postupkom hand-eye umjeravanja opisanim u poglavlju 2.3.3. Vizualno vođenje se može izvoditi više puta s istim parametrima  ${}^C z_R$  i  ${}^C t_R$  sve dok ne postoji potreba za ponovnim umjeravanjem.

Pretpostavimo geometriju konfiguracije SCARA robota prikazanu na slici 2.1, pri čemu  $a_1$  i  $a_2$  predstavljaju duljine članaka robotske ruke. Kako bi se objasnila razmatrana metoda vizualnog vođenja, uveden je koordinatni sustav robota smješten u ishodištu osi ramenog zgloba, pri čemu na toj osi leži  $z$ -os koordinatnog sustava. Preostale dvije osi definirane su ovisno o trenutnoj poziciji markera, pri čemu je  $x$ -os orijentirana prema markeru, kao što je prikazano na slici 2.1. U svakom korekcijskom koraku vizualnog vođenja,  $x$  i  $y$ -os koordinatnog sustava robota su redefinirane.

Neka je  $M$  trenutna, a  $M'$  odredišna pozicija markera. Vizualno vođenje računa potrebnu promjenu u vertikalnom položaju,  $\Delta z$ , te potrebne promjene u kutevima rotacijskih zglobova,  $\Delta q_2$  i  $\Delta q_3$ . Promjena u vertikalnom položaju ostvarena je pomicanjem prvog, translacijskog, zgloba. Promjene u rotacijskim zglobovima računaju se na temelju geometrije prikazane na slici 2.1.



Slika 2.1: Robotska ruka u trenutnom (crveno) i odredišnom (zeleno) položaju

Potrebna promjena u translacijskom zglobo,  $\Delta z$ , neovisna je o preostalim varijablama

rotacijskih zglobova. Promjena  $\Delta z$  računa se na sljedeći način

$$\Delta z = z' - z \quad (2.3)$$

za što su potrebne samo trenutna,  $z$ , i odredišna  $z'$  vrijednost  $z$ -koordinate centra markera u odnosu na korodinatni sustav robota. Pri tome,

$$z = {}^C z_R^\top ({}^C p_M - {}^C t_R), \quad (2.4)$$

a  $z'$  se računa analogno

$$z' = {}^C z_R^\top ({}^C p'_M - {}^C t_R). \quad (2.5)$$

Potrebna promjena u zglobu lakta,  $\Delta q_3$ , računa se na sljedeći način

$$\Delta q_3 = q'_3 - q_3, \quad (2.6)$$

pri čemu  $q_3$  predstavlja trenutni kut zgloba lakta, a računa se na temelju standardne konfiguracije planarnih robotskih manipulatora ([Craig, 2005](#); [Kucuk i Bingul, 2006](#))

$$q_3 = \text{atan2}(s_2, c_2) \quad (2.7)$$

pri čemu su

$$c_2 = \frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1 a_2}, \quad (2.8)$$

$$s_2 = \pm \sqrt{1 - c_2^2}, \quad (2.9)$$

a  $x$  i  $y$  su koordinate centra markera. Kut  $q'_3$  predstavlja kut zgloba u odredišnoj poziciji te se računa analogno (2.7).

Promjena potrebna u ramenom zglobu,  $\Delta q_2$ , računa se na sljedeći način:

$$\Delta q_2 = \alpha - \alpha' + \Delta \varphi \quad (2.10)$$

pri čemu  $\Delta \varphi$  predstavlja kut između vektora  $r$  i  $r'$ , prikazanih na slici 2.1 dok se  $\alpha$  računa na sljedeći način

$$\alpha = \text{asin} \frac{a_2 \sin q_3}{\|r\|}, \quad (2.11)$$

a  $\alpha'$  se računa analogno.

Vektor  $r$ , koji povezuje os ramenog zgloba i trenutnu poziciju markera računa se na

sljedeći način:

$$r = {}^C p_M - {}^C t_R - z \cdot {}^C z_R, \quad (2.12)$$

a vektor  $r'$ , koji povezuje os ramenog zgloba i određenu poziciju markera, računa se analogno kao  $r$ .

Ovaj se algoritam iterativno ponavlja sve dok marker ne dostigne određenu poziciju unutar zadane tolerancije. Tolerancija predstavlja najveću dozvoljenu udaljenost između zadane i dostignute pozicije. Nije poželjno da je tolerancija postavljena na nulu zbog utjecaja mjernog šuma, zračnosti i ograničene preciznosti robota koje sprječavaju da robot dostigne točno zadanu poziciju, zbog čega bi program za vizualno vođenje završio u beskonačnoj petlji.

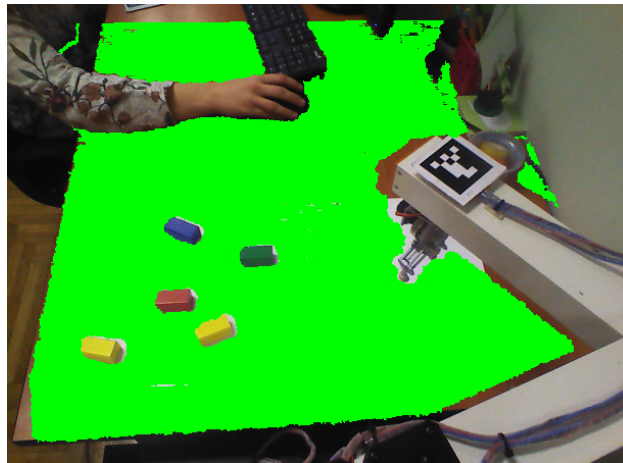
Točnost pozicioniranja razmatranog robotskog sustava ovisi o točnosti određivanja pozicije markera mjerene vizijom, određivanja ciljne točke na koju treba dovesti vrh alata i o točnosti određivanja osi ramenog zgloba u odnosu na kameru koja je estimirana na temelju detekcije dominantne (potporne) ravnine, kao što je objašnjeno u potpoglavlju 2.3.3. Pogreška u određivanju pozicije markera daje pogrešnu informaciju o trenutnom položaju vrha alata. Ciljna točka u ovom je radu predstavljena središtem određivanja markera postavljenog u radni prostor robota, ili središtem jednostavnog konveksnog objekta namijenjenog za hvatanje. Pogrešno određena pozicija određivanja markera ili pogrešno detektiran objekt i njemu pripadajuće središte daju pogrešnu informaciju o ciljnom položaju vrha alata te utječu na točnost pozicioniranja. Nadalje, pozicija vrha alata se izračunava na temelju pozicije središta markera i orijentacije osi ramenog zgloba uz pretpostavku da je marker montiran na robotsku ruku tako da je pravac na kojem leže njegovo središte i vrh alata paralelan s osi ramenog zgloba. To znači da pogreška estimacije osi ramenog zgloba doprinosi pogrešci određivanja pozicije vrha alata jer rezultira netočnim izračunom promjene kuta  $\Delta q_2$ . Točnost određivanja referentne točke  ${}^C t_R$  ne utječe na točnost pozicioniranja, ali utječe na broj iteracija vizualnog vođenja. Bolja estimacija  ${}^C t_R$  rezultira manjim brojem iteracija.

### 2.3.3 *Hand-eye* umjeravanje za relativno pozicioniranje

Parametri osi ramenog zgloba,  ${}^C z_R$  i  ${}^C t_R$ , potrebni za vizualno vođenje opisano u potpoglavlju 2.3.2, određuju se postupkom umjeravanja opisanim u ovom poglavlju. Razvijena metoda umjeravanja određuje orijentaciju osi ramenog zgloba na temelju detekcije potporne ravnine.



Pozicija ove osi definirana je proizvoljnom referentnom točkom na osi ramenog zgloba koja je određena na temelju samo jednog rotacijskog pokreta ramenog zgloba. Potporna ravnina može se estimirati RANSAC algoritmom, na sljedeći način. Prvo se nasumično odaberu tri točke na RGB-D slici te se odrede parametri ravnine koja prolazi tim trima točkama. Sve točke na slici koje pripadaju toj ravnini unutar zadane tolerancije predstavljaju konsenzus skup. Ovaj se postupak ponavlja zadani broj puta te se odabiru parametri ravnine s najvećim konsenzus skupom. Konačno, odabrani konsenzus skup aproksimiran je ravninom određenom metodom najmanjih kvadrata. Primjer potporne ravnine, detektirane na RGB-D slici, prikazan je slikom 2.2, na kojoj su zelenom bojom označene točke koje pripadaju toj ravnini. Pod pretpostavkom da je os ramenog zgloba robota okomita na potpornu ravninu, orijentacija normale detektirane potporne ravnine identična je orijentaciji osi ramenog zgloba  ${}^C z_R$ . U ovom radu, riječ normala označava jedinični vektor okomit na neku površinu.



Slika 2.2: Detekcija potporne ravnine označene zelenom bojom

Razmotrimo gibanje robota koje premješta marker iz početne točke  $M$  u konačnu točku  $M'$ , pri čemu se samo rameni zglob rotira za poznatu vrijednost kuta rotacije,  $\Delta q_2$ . Rotacija točke  $M$  oko osi koja prolazi točkom definiranom vektorom  ${}^C t_R$ , pri čemu je orijentacija osi definirana vektorom  ${}^C z_R$ , opisana je jednadžbom:

$$R({}^C z_R, \Delta q_2) \cdot ({}^C p_M - {}^C t_R) = {}^C p_{M'} - {}^C t_R, \quad (2.13)$$

pri čemu  $R({}^C z_R, \Delta q_2)$  predstavlja rotacijsku matricu koja opisuje rotaciju oko osi  ${}^C z_R$  za kut  $\Delta q_2$ . Vektor  ${}^C t_R$  računa se rješavanjem jednadžbe (2.13) za  ${}^C p_M$  i  ${}^C p_{M'}$  dobivene softverom za praćenje markera. Predložena metoda *hand-eye* umjeravanja sastoji se sljedećih koraka:

1. Na dubinskoj slici dobivenoj kamerom svakoj se točki dodijele 3D koordinate u koor-

dinatnom sustavu kamere.

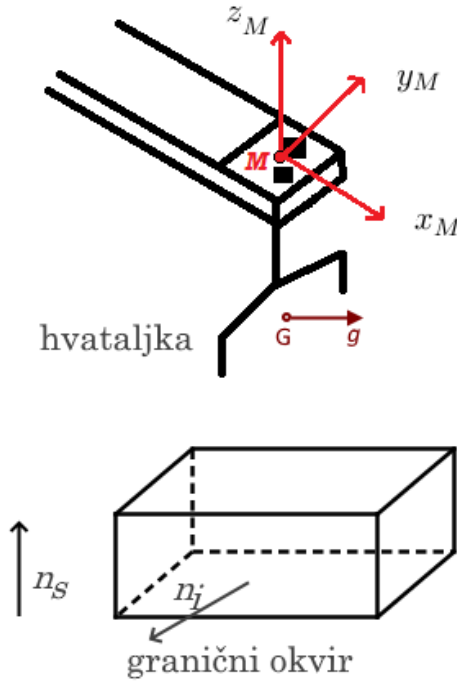
2. Standardnim RANSAC algoritmom računaju se parametri dominantne ravnine na sceni te se definira  $z$ -os koordinatnog sustava robota u odnosu na koordinatni sustav kamere kao jedinični vektor okomit na dominantnu ravninu.
3. Na snimljenoj slici detektira se marker smješten pri vrhu alata. Središte markera predstavlja točku  ${}^C p_M$ .
4. Izvodi se rotacijski pokret ramenog zgloba za proizvoljni kut rotacije  $\Delta q_2$ , u pozitivnom smjeru.
5. Ponovno se uzima slika snimljena kamerom te se pronalazi novo središte markera  ${}^C p_{M'}$ .
6. Ishodište koordinatnog sustava robota,  ${}^C t_R$ , u odnosu na koordinatni sustav kamere računa se rješavanjem jednadžbe (2.13).

## 2.4 Planiranje hvatanja jednostavnih objekata pomoću računalnog vida

U ovom je poglavlju opisana metoda za planiranje hvatanja koja se zasniva na graničnim okvirima objekata detektiranim na RGB-D slici. Pretpostavlja se da hvataljka hvata objekte samo odozgo, što je tipično za SCARA robote. Kako bi se izvršilo uspješno hvatanje, potrebno je odrediti potrebnu orijentaciju hvataljke, njezinu poziciju iznad objekta te širinu otvora hvataljke. Predložena je metoda planiranja hvatanja ograničena na jednostavne konveksne objekte. Objekti od interesa detektiraju se na RGB-D slici u radnoj okolini robota pomoću metode predložene u radu (Cupec et al., 2017). Ukratko, RGB-D slika je segmentirana u ravninske segmente te su susjedni segmenti grupirani u objekte na temelju kriterija konveksnosti. Na taj je način rezultat ove metode jedan ili više objekata predstavljenih ravninskim segmentima. Razmatrajući hvatanje odozgo te uz pretpostavku da objekti leže na potpornoj ravnini, smatra se da slabo zakrivljena površina objekta, orijentirana pod strmim kutom u odnosu na potpurnu ravninu, predstavlja stabilnu točku hvatanja. Vektor hvatanja,  $g$ , koji definira orijentaciju hvataljke, kao što je prikazano na slici 2.3, okomit je na normalu potporne ravnine i na normalu jednog ravninskog segmenta objekta te se računa na sljedeći način

$$g = \frac{n_s \times n_i}{\|n_s \times n_i\|} \quad (2.14)$$

gdje  $n_s$  predstavlja normalu potporne ravnine, a  $n_i$  normalu  $i$ -tog ravninskog segmenta objekta. Ravninski segment pomoću kojeg se računa vektor  $g$  odabire se tako da jednadžba (2.14) ima minimalnu nesigurnost orijentacije.



Slika 2.3: Koordinatni sustav markera i orijentacija alata

Nesigurnost vektora  $g$  ovisi o nesigurnosti  $n_i$ , kao i o njezinoj orijentaciji. Nesigurnost normale  $n_i$  estimirana je postupkom opisanim u radu (Cupec et al., 2013). Izračunava se matrica kovarijance  $\Sigma_p$  svih točaka koje pripadaju razmatranom ravninskom segmentu. Svojstveni vektor koji odgovara najmanjoj svojstvenoj vrijednosti matrice kovarijance  $\Sigma_p$  predstavlja normalu ravninskog segmenta, dok preostale druge dvije svojstvene vrijednosti opisuju područje u ravnini ravninskog segmenta pokriveno točkama tog segmenta. Te su dvije vrijednosti veće za veće ravninske segmente koji predstavljaju područje površine objekta male zakrivljenosti. Nesigurnost orijentacije ravninskog segmenta određena je sljedećom jednadžbom

$$n_i = \hat{n}_i + M_i s_i \quad (2.15)$$

pri čemu je  $n_i$  prava vrijednost normale ravninskog segmenta, a  $\hat{n}_i$  je njezina mjerena vrijednost. Matrica  $M$  ima dva stupca koji predstavljaju svojstvene vektore koji odgovaraju dvama najvećim svojstvenim vrijednostima matrice kovarijance  $\Sigma_p$ , a  $s_i$  je vektor od dvije komponente koji predstavlja odstupanje  $n_i$  od  $\hat{n}_i$  u dva smjera okomita na  $\hat{n}_i$ .

Matrica kovarijance  $\Sigma_{n_i}$ , koja predstavlja razdiobu vektora odstupanja  $s_i$ , jest dijagonalna matrica čiji elementi na dijagonali su približno obrnuto proporcionalni dvijema većima svojstvenim vrijednostima matrice  $\Sigma_p$  (Cupec et al., 2013). Na taj način, veći ravninski segmenti imaju manju nesigurnost normale. Nesigurnost vektora  $g$  može se estimirati propagacijom nesigurnosti normale  $n_i$ . Matrica kovarijance  $\Sigma_g$  koja opisuje nesigurnost vektora  $g$  računa se sljedećom jednadžbom

$$\Sigma_g = \frac{dg}{ds_i} \cdot \Sigma_{n_i} \cdot \left(\frac{dg}{ds_i}\right)^\top \quad (2.16)$$

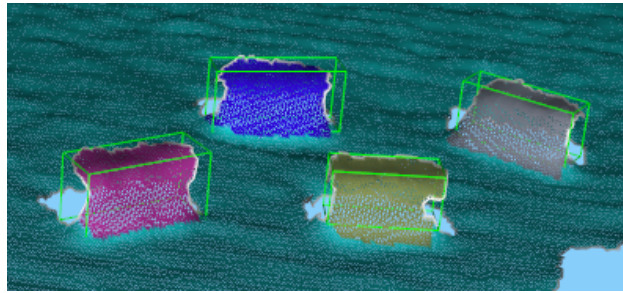
pri čemu  $\frac{dg}{ds_i}$  predstavlja Jacobian, koji se računa supstitucijom (2.15) u (2.14) te parcijalnom derivacijom dobivenoga vektora po komponentama od  $s_i$ .

Konačno, mjera nesigurnosti orijentacije vektora  $g$  računa se projekcijom  $\Sigma_g$  u smjeru okomitom na  $g$  i  $n_s$ . Ova se projekcija računa jednadžbom

$$\sigma_g = n_i^\top \Sigma_g u \quad (2.17)$$

gdje  $n_i$  predstavlja jedinični vektor okomit na  $g$  i  $n_s$ . Vrijednost  $\sigma_g$  računa se za svaki ravninski segment objekta, a konačni vektor  $g$  se računa pomoću normale ravninskog segmenta s pripadajućom najmanjom vrijednosti  $\sigma_g$ .

Stabilan je hvat određen računanjem graničnog okvira objekta od interesa, čije su stranice poravnane s vektorima  $n_s$ ,  $g$  i  $n_i$ . Primjer detektiranih objekata na RGB-D slici u radnom prostoru robota i njihovi granični okviri prikazani su na slici 2.4.



Slika 2.4: Primjeri graničnih okvira detektiranih objekata, označeni zelenim linijama

Osnovna ideja predložene metode zasniva se na činjenici da ako pravac koji spaja točke hvatanja prolazi kroz centar mase objekta i ako je približno okomit na normalu površine u točkama hvatanja, hvat će biti stabilan. Pretpostavlja se da je centar mase objekta blizu centra mase njegova graničnoga okvira. U tom slučaju, točke hvatanja definirane su na način da pravac koji ih spaja prolazi centroidom graničnog okvira. Parametri ravnina graničnog okvira pružaju dovoljno informacija za približni izračun centra mase objekta, a točka  ${}^C p_G$

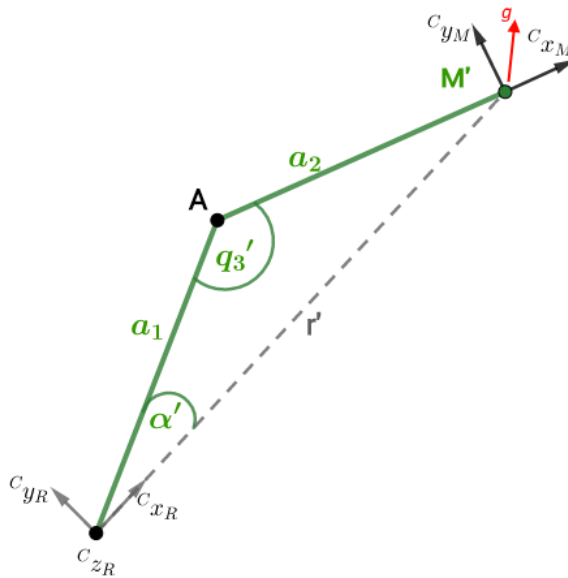
predstavlja određenu poziciju središta markera u postupku vizualnog vođenja, tj. središnju točku između prstiju hvataljke.

Pretpostavlja se da je hvataljka na početku operacije hvatanja orijentirana tako da je pravac na kojemu leže prsti hvataljke okomit na smjer drugog članka robotske ruke, označenog na slici 2.5 s  $a_2$ , koji je paralelan s  $x$ -osi koordinatnog sustava markera  ${}^C x_M$ . Nakon pozicioniranja robotske ruke iznad određene točke, hvataljka se rotira za kut  $q_4$ , koji predstavlja kut između  ${}^C x_M$  i  $g$ , kao što je prikazano na slici 2.5. Vektor  ${}^C x_M$  se računa kao jedinični vektor paralelan pravcu koji spaja središte markera  $M$  i točku  $A$  na osi trećeg zgloba.

Pozicija točke  $A$  u odnosu na koordinatni sustav kamere računa se na sljedeći način

$${}^C p_A = {}^C t_R + a_1 \cdot ({}^C x_R \cdot \cos\alpha' + {}^C y_R \cdot \sin\alpha') - z' \cdot {}^C z_R, \quad (2.18)$$

pri čemu su  $\alpha'$  i  $z'$  objašnjeni u potpoglavlju 2.3.2.



Slika 2.5: Koordinatni sustav robota i alata u poziciji  ${}^C p_{M'}$

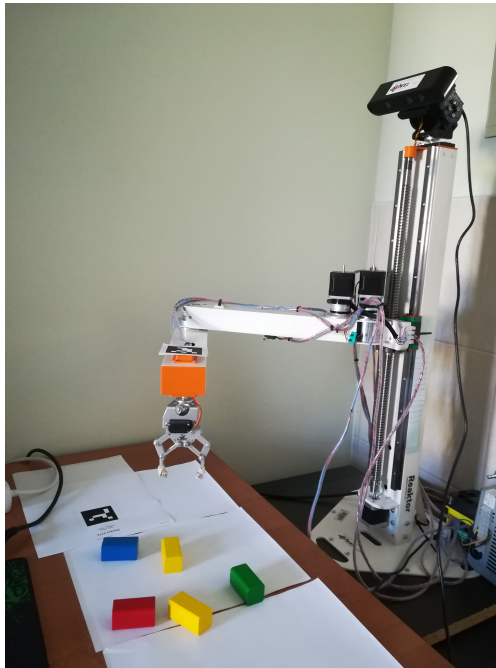
Na ovaj su način, dakle, definirane pozicija (2.18) i orijentacija (2.14) hvataljke, kut rotacije hvataljke  $q_4$  i širina otvora hvataljke koja se računa kao udaljenost između dviju stranica graničnoga okvira, paralelnih s vektorom  $g$ .

## 2.5 Eksperimentalna evaluacija

U ovom je poglavlju opisana eksperimentalna analiza predložene metode za vizualno vođenje i planiranje hvatanja.

### 2.5.1 Eksperimentalni postav

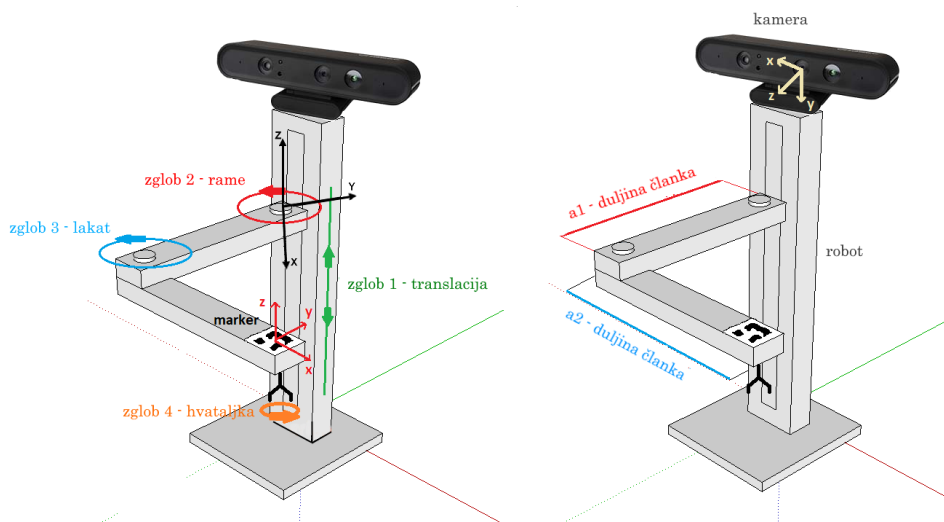
Robotski sustav za koji je razvijen predloženi algoritam sastoji se od robotske ruke u SCARA konfiguraciji, RGB-D kamere, makera za praćenje te upravljačkog programa. Predloženi pristup je ispitan robotskom rukom VICRA (VIsion Controlled Robot Arm), koja je prikazana na slici 2.6. VICRA se sastoji od jednog translacijskog i tri rotacijska zgloba. Prva su tri zgloba, koja pozicioniraju vrh alata, pokretana koračnim motorima, dok je četvrti zglob, koji definira orijentaciju alata, pokretan DC servo motorom. Prvi translacijski zglob omogućava vertikalni doseg od otprilike 0.6 m, dok rotacijski zglobovi omogućuju horizontalni doseg od, također, otprilike 0.6 m. Masa robotske ruke je otprilike 12 kg, što je čini pogodnom za postavljanje na mobilnu platformu. Robot je pokretan mikroupravljačem baziranom na platformi Arduino, a komunikacija s računalom ostvarena je pomoću USB protokola.



Slika 2.6: VICRA - robotska ruka u SCARA konfiguraciji

RGB-D kamera, postavljena na *pan-tilt* glavu na vrhu robota, promatra radni prostor robota. Upotrijebljena je kamera niske cijene, Orbec Astra S ([ast](#), [Pristup: 2020-06-01](#)), optimizirana za primjenu unutar manjeg doseg od 0.35 do 2.5 m, što je prikladno za manje robote, gdje je kamera relativno blizu objekta od interesa.

Hvataljka je postavljena kao alat te je, zavisno o pokusu, zamjenjiva laserom. Pozicioniranje alata ostvareno je praćenjem markera postavljenog pri vrhu alata, tako da mu središte leži na osi četvrtog zgloba, kao što je prikazano na slici 2.7. Detekcija markera i estimacija



Slika 2.7: Vizualno navođen robotski sustav u SCARA konfiguraciji

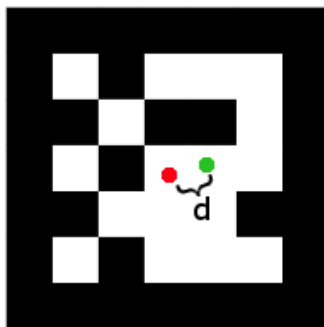
njegovog položaja implementirana je u ArUco biblioteci za proširenu stvarnost ([aru](#), [Pristup: 2020-06-01](#)) zasnovanoj na biblioteci Open CV ([Kaehler i G, 2016](#)).

Cijena cjelokupnog sustava je približno €3500. Komercijalna bi cijena takvog robota bila i manja, jer je razmatrani robot prototip te je u cijenu uključen i njegov razvoj.

## 2.5.2 Pokusi vizualnog vođenja

Razvijeni je algoritam eksperimentalno ispitan kako bi se utvrdila točnost vizualnog vođenja na temelju predložene metode umjeravanja. U tu je svrhu hvataljka zamijenjena laserskim pokazivačem. Osim markera pri vrhu alata, središte drugog markera, postavljenog u radni prostor robota, predstavljalo je odredišnu poziciju. Robotska je ruka trebala pozicionirati laserski pokazivač u blizini odredišne pozicije. Nakon što je pozicioniranje završeno, udaljenost  $d$  između centra markera i svjetlosne točkice koju projicira laser ručno je mjerena. Primjer je prikazan na slici 2.8, gdje crvena točkica predstavlja središte markera, a zelena točkicu lasera.

Imajući na umu zračnost u zglobovima lakta i ramena, kompenzacijski pokret je uključen svaki puta kada zglob mijenja smjer rotacije. Također, na početku provedbe eksperimenta, učinjen je inicijalni pokret, rotacija u pozitivnom smjeru oba zgloba. Na taj je način osigurano da je poznat inicijalni smjer rotacije kako bi se pravilno kompenzirala zračnost. Sljedeći je eksperiment proveden pet puta. Svaki je puta kamera zarotirana, kako bi se promijenio



Slika 2.8: Udaljenost  $d$  između centra markera (crvena točka) i točke lasera (zeleno točka)

relativni odnos između koordinatnih sustava kamere i robota te je provedena predstavljena metoda umjeravanja. Nakon toga, marker čiji centar predstavlja određenu poziciju robotske ruke, postavljen je u 15 različitih pozicija u radnom prostoru robota. Provedeno je vizualno vođenje čiji su rezultati predstavljeni u tablici 2.1. Na ovaj su način ispitane točnost i ponovljivost predložene metode.

Tablica 2.1: Eksperimentalni rezultati vizualnog vođenja

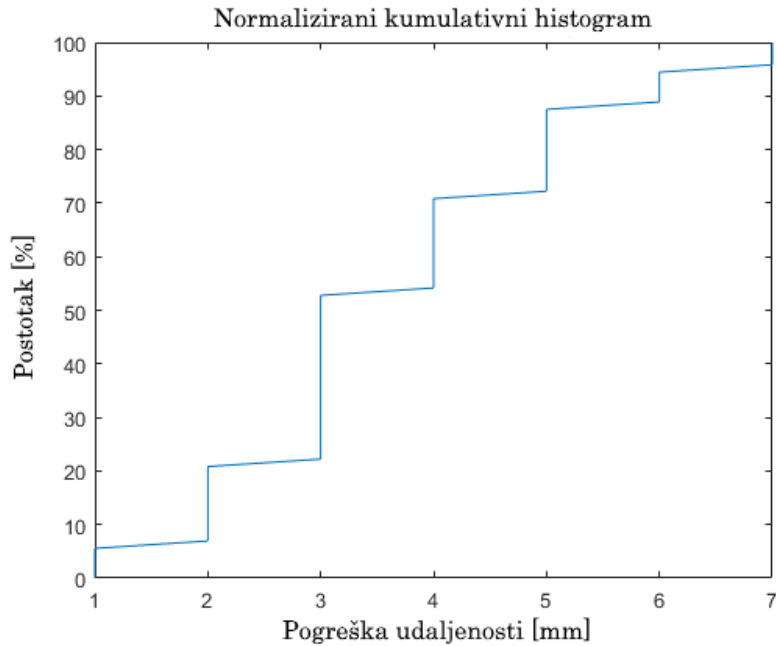
|  | Exp.1 | Exp.2 | Exp.3 | Exp.4 | Exp.5 | Prosjek |
|--|-------|-------|-------|-------|-------|---------|
| <b>Prosječna udaljenost<br/>15 mjerenja, <math>\bar{d}</math> [mm]</b> | 3.47  | 3.00  | 4.00  | 3.87  | 3.93  | 3.65    |

Na slici 2.9, prikazan je normalizirani kumulativni histogram, na kojem  $x$ -os predstavlja udaljenost u milimetrima, a  $y$ -os predstavlja postotak eksperimenata za koje je udaljenost bila manja od odgovarajuće vrijednosti na  $x$ -osi. Kao što se vidi, 87.67% provedenih eksperimenata ima pogrešku udaljenosti manju od 5 mm. Najveći utjecaj na točnost pozicioniranja ima nesigurnost estimacije  $z$ -osi korodinatnog sustava robota. Pogreška pozicioniranja proporcionalna je razlici u visini između markera pri vrhu robotskog alata i markera koji predstavlja objekt od interesa. U ovoj je evaluaciji ta visina bila otprilike 200 mm, što rezultira pogreškom u pozicioniranju od 5 mm, ako je estimacija  $z$ -osi pogrešna za  $1.43^\circ$ .

### 2.5.3 Pokusi hvatanja

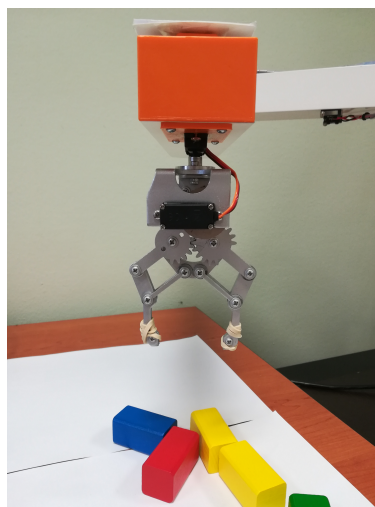
Kako bi se procijenila mogućnost primjene predložene metode za planiranje hvatanja provedeno je 12 serija pokusa. Svaka se serija pokusa sastojala od *hand – eye* umjeravanja nakon kojeg je provedeno pet pokusa hvatanja. Tijekom svake operacije hvatanja, objekt je postavljen na horizontalnu ravninu u radnom prostoru robota, kao što je prikazano na





Slika 2.9: Normalizirani kumulativni histogram pogreške pozicioniranja alata

slici 2.6, detektiran je na sceni te su izračunati njegov centroid, koji predstavlja određenu poziciju za vizualno navođenje, i kut rotacije hvataljke potreban za uspješno hvatanje. Ovaj je izračun objašnjen u potpoglavlju 2.4. Kut rotacije hvataljke upotrijebljene u pokusima definiran je u intervalu  $q_4 \in [0, \pi]$ . Slika 2.10 predstavlja početnu poziciju hvataljke, kada je  $q_4 = 0$ . Vizualno vođenje navodi robotsku ruku iznad objekta od interesa te se izvodi operacija hvatanja. Objekt je konačno prenesen na određenu poziciju u radnom području robota predstavljenu markerom.



Slika 2.10: Inicijalni položaj hvataljke

Pokus je smatran uspješnim ako je objekt pravilno detektiran, ako je robotska ruka po-

zicionirana iznad objekta, hvataljka se pravilno rotirala te uspješno uhvatila objekt koji je tada podignut i prenesen na određenu destinaciju. Rezultati su prikazani u tablici 2.2. Od ukupno 60 pokusa hvatanja, četiri su bila neuspješna zbog pogreške pri prepoznavanju objekta, što nije doprinos ovog rada. Stoga su ti pokusi izuzeti iz statističke obrade rezultata. Tri su pokusa hvatanja bila neuspješna zbog nedovoljne preciznosti vizualnog vođenja. U tim je pokusima objekt nepravilno uhvaćen te je zbog toga ispao iz hvataljke. Ostatak je pokusa bio uspješan.

Tablica 2.2: Rezultati pokusa hvatanja

|              | Uspješno | Neuspješno |
|--------------|----------|------------|
| Broj pokusa  | 53       | 3          |
| Postotak [%] | 94.64    | 5.36       |

## 2.6 Zaključak

U ovom je poglavlju opisan robotski sustav upravljani pomoću računalnog vida, koji koristi samo vizualnu informaciju kako bi pozicionirao vrh alata i hvatao jednostavne objekte. S obzirom da se radi o robotu bez enkodera u zglobovima, komercijalna cijena takvog sustava može biti dosta niska. Opisani je sustav zasnovan na vizualnom vođenju koji koristi novu, brzu *hand-eye* metodu umjeravanja. Brzo vrijeme izvođenja predložene metode od velike je važnosti kada je potrebno često umjeravanje. Prikazani rezultati provedenih pokusa pokazuju da je postignuta točnost pozicioniranja prikladna za manipulaciju objektima za koje je dovoljna točnost od 7 mm. Pokusi hvatanja bili su uspješni u 95% slučajeva.

Točnost pozicioniranja značajno ovisi o točnosti određivanja dominantne potporne ravnine, pri čemu se pogreška u pozicioniranju povećava linearno s udaljenošću ciljne točke hvatanja na sceni i markera pri vrhu alata. Točnost pozicioniranja bi se, dakle, mogla poboljšati drugačijim dizajnom robotske ruke kod kojega bi ta udaljenost bila manja. Osim toga, zapažena je netočna registracija RGB i dubinske slike snimljene spomenutom kamerom. S obzirom da predložena metoda ovisi o informaciji sa RGB i dubinske slike, te je njihovo precizno poravnanje važno, ova se pogreška može eliminirati uporabom drugog senzora, koji pruža točniju registraciju RGB i dubinske slike ili primjenom algoritma umjeravanja koji bi osigurao optimalnu registraciju.

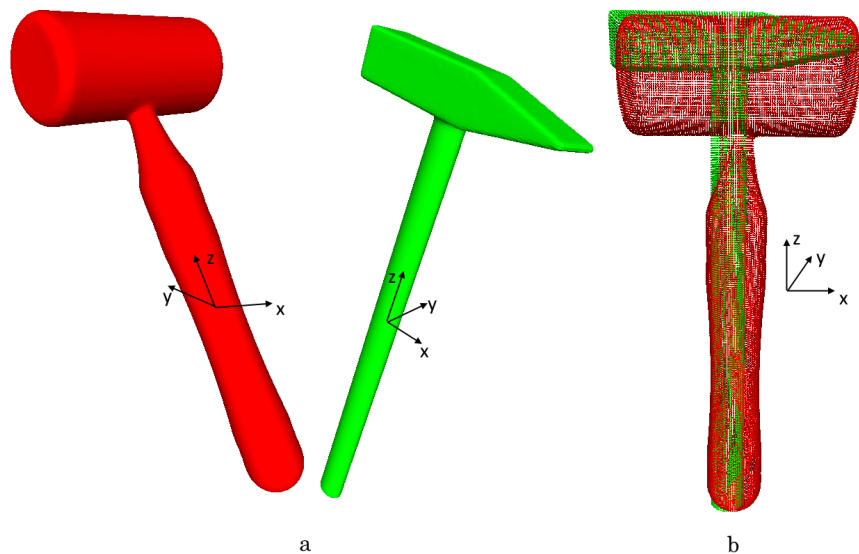
# 3

## Klasifikacija objekata na dubinskim slikama zasnovana na poravnanju konveksne ljuske

Kako bi se radne akcije robota, poput hvatanja predloženog u prethodnom poglavlju, osim za jednostavne konveksne objekte primijenile i za složenije, prvi puta viđene objekte, potreban je razvoj metoda klasifikacije koje estimiraju položaj klasificiranog objekta. U ovom je poglavlju, stoga, predstavljen drugi doprinos ovog doktorskog rada, metoda klasifikacije objekata na dubinskim slikama zasnovana na poravnanju konveksne ljuske. Klasifikacija objekata neizostavan je dio naprednog robotskog vida, a primjenjuje se često u svrhu autonomne robotske manipulacije. S obzirom da objekti pripadnici iste klase nerijetko imaju sličan oblik, u radu ([Đurović et al., 2018](#)) predložena je metoda za poravnanje objekata (engl. *object alignment*) koja se zasniva na sličnosti njihovih konveksnih ljuski (engl. *convex hull*). Svrha razvijene metode je generiranje hipoteza za klasifikaciju objekata i određivanje položaja klasificiranog objekta. Predloženi se pristup sastoji od generiranja prijedloga poravnanja i evaluacije tih prijedloga metodom koja se često primjenjuje u algoritmima za detekciju objekata. Predložena metoda ispitana je na referentnom skupu podataka te postiže točnost usporedivu s rezultatima četiri suvremene metode za klasifikaciju objekata.

### 3.1 Uvod

Prepoznavanje objekata od interesa važna je mogućnost inteligentnih robota koji djeluju u nestrukturiranim okolinama. Dva su osnovna problema vezana uz prepoznavanje objekata,



Slika 3.1: Dva slična oblika (a) i njihovo poravnanje (b).

kojima se opširno bave u literaturi, a to su prepoznavanje objekata točno definiranog oblika i klasifikacija objekata. Prvi se problem svodi na prepoznavanje objekta od interesa na sceni, pri čemu je dostupan točan 3D model tog objekta. U ovom se radu taj problem naziva detekcija instanci oblika (engl. *shape instance detection*). Moguće je da algoritam prepoznavanja istovremeno prepozna više ciljnih objekata različitog oblika. U tom slučaju algoritam koristi bazu objekata koja sadrži 3D modele svih ciljnih objekata. Drugi problem, klasifikacija objekata, proširuje problem prepoznavanja na objekte čiji točni model nije dostupan, ali pripadaju jednoj od klasa koju je sustav naučio prepoznavati. Učenje se odvija analiziranjem modela objekata pripadnika iste klase, koji se nalaze u skupu za učenje, pri čemu je svakom modelu pridružena oznaka pripadnosti određenoj klasi. Na osnovi učenja, algoritam treba biti u stanju prepoznati objekte koji nisu u skupu za učenje, ali su mu slični. Primjer dva objekta različitog oblika koji pripadaju istoj klasi prikazan je na slici 3.1.

U ovom se radu razmatra klasifikacija objekata na dubinskim slikama, npr. na slikama snimljenima RGB-D kamerom. Neke od najuspješnijih metoda za detekciju instanci oblika sastoje se od izračuna deskriptora lokalnih značajki, generiranja hipoteza objekata baziranih na sparivanju deskriptora, evaluacije hipoteza poravnanjem modela i scene te određivanjem koliko je dobro scena opisana pojedinom hipotezom. Većina metoda za klasifikaciju objekata koje se zasnivaju na neuronskim mrežama bazirana je na sparivanju deskriptora predstavljenih histogramima određenih geometrijskih značajki, boje ili teksture. S druge strane,

metoda klasifikacije predstavljena u ovom poglavlju oslanja se na metodologiju svojstvenu detekciji instanci oblika. Tijekom detekcije instanci oblika, evaluacija hipoteza zasniva se na projiciranju modela na scenu i mjerenju koliko je točaka scene blizu projiciranih točaka modela (Aldoma et al., 2012, 2016; Papazov i Burschka, 2010). Dodatno, tijekom evaluacije hipoteza penaliziraju se točke modela koje zaklanjaju vidljive točke scene. Ovakav je postupak moguć samo ako je model ispravno poravnan s relevantnim podskupom točaka scene. Poravnanje nije važno samo u rješavanju problema detekcije ciljnih objekata. Kako bi se obavila radnja robotske manipulacije, koja obuhvaća planiranje hvatanja, važno je odrediti točan položaj objekta na sceni u odnosu na koordinatni sustav robota. Kako su objekti pripadnici iste klase uglavnom sličnih oblika, pretpostavlja se da su mjesta na površini objekta relevantna za obavljanje neke operacije (npr. hvatanje i prenošenje objekta) približno na sličnim pozicijama u odnosu na neki referentni koordinatni sustav. To omogućuje da se neka radna operacija robota definirana na jednoj ili više instanci iz neke klase objekata primijeni na druge pripadnike iste klase. Ovo se može postići poravnanjem ciljnog objekta detektiranog na sceni s referentnim objektom dane klase te planiranjem operacije koju treba izvršiti nad ciljnim objektom na temelju operacije definirane za referentni objekt.

U ovom se poglavlju predlaže nova metoda za efikasno poravnanje objekata. Predložena je metoda primijenjena u algoritmu za klasifikaciji objekata, gdje se prethodno neviđeni objekti poravnavaju s modelima iz baze objekata na temelju sličnosti njihove konveksne ljuske te se odabire poravnanje za koje je sličnost između ispitivanog (engl. *query*) objekta i modela iz baze poravnog s razmatranim objektom najveća. Baza objekata predstavlja skup 3D modela objekata pripadnika jedne ili više ciljnih klasa. Poravnanja se evaluiraju postupkom koji se često koristi pri detekciji instanci oblika. Prednost predložene metode je što se razmatranom objektu, osim pripadnosti nekoj klasi, određuje i položaj u odnosu na referentni model iz baze objekata.

Ovo se poglavlje sastoji od sljedećih dijelova. Potpoglavlje 3.2 donosi pregled istraživanja u području klasifikacije objekata. U potpoglavlju 3.3 predložena je metoda poravnanja objekata zasnovana na sličnosti njihovih konveksnih ljuski. Nadalje, u potpoglavlju 3.4 predložena je metoda klasifikacije objekata koja se sastoji od dva koraka, generiranje prijedloga poravnanja i evaluacije tih prijedloga. Eksperimentalna evaluacija predloženih metoda dana je u potpoglavlju 3.5. Konačan pregled postignutih rezultata i zaključak doneseni su u zadnjem potpoglavlju 3.6.

## 3.2 Istraživanja u ovom području

Općenito, klasifikacija objekata provodi se ekstrakcijom značajki nakon koje slijedi primjena klasifikatora. U računalnom vidu, dugo su bili dominantni ručno dizajnirani deskriptori značajki, zasnovani na histogramima, poput SHOT (Salti et al., 2014; Wohlkinger et al., 2012), Fast Point Feature Histogram (FPFH) (Khan et al., 2016; Rusu et al., 2009), Shape Distributions (Osada et al., 2002), Shape Distributions on Voxel Surfaces (Wohlkinger i Vincze, 2011b) i Ensemble of Shape Functions (ESF) (Wohlkinger i Vincze, 2011a; Patten et al., 2016). To su ili globalni deskriptori koji opisuju cijeli objekt ili deskriptori široke primjene koji se koriste kao globalni (Wohlkinger et al., 2012; Khan et al., 2016). Lokalni deskriptori, od kojih su neki opisani u (Guo et al., 2016), rijetko se koriste za klasifikaciju kakva se razmatra u ovom poglavlju ovog doktorskog rada.

U svrhu klasifikacije, razvijaju se i tehnike učenja značajki, poput (Bo et al., 2012; Blum et al., 2012). U novije vrijeme sve su više zastupljene metode zasnovane na neuronskim mrežama (Qi et al., 2016; Brock et al., 2016; Klovov i Lempitsky, 2017; Qi et al., 2017a,b; Gupta et al., 2016; Dai et al., 2017), koje rješavaju problem klasifikacije na velikom broju skupova podataka. Kako bi klasifikacija objekata imala primjenu u robotici, važno je da osim prepoznavanja klase, metoda ispravno estimira položaj klasificiranog objekta na sceni. Navedene metode uglavnom kao rezultat daju samo klasu, dok metoda predložena u ovom poglavlju, osim određivanja klase, uspješno estimira i položaj objekta.

Oba spomenuta pristupa, ručno dizajnirani deskriptori i neuronske mreže, imaju neki način provedbe poravnanja objekata. Većina ručno dizajniranih deskriptora invarijantni su na translaciju i rotaciju. Globalni deskriptori (Osada et al., 2002; Wohlkinger i Vincze, 2011b,a) zasnovani su na histogramima te su stoga invarijantni na translaciju, rotaciju i skalu. Za metode bazirane na nenadgledanom učenju poput (Yu et al., 2011; Bo et al., 2011; Coates i Ng, 2011; Blum et al., 2012) nije potpuno jasno koju razinu invarijantnosti na krute transformacije postižu te to očito ovisi o skupu podataka za učenje i ispitivanje. Za metode koje se zasnivaju na dubokim neuronskim mrežama, invarijantnost na uobičajene transformacije detaljno je opisana u (Jaderberg et al., 2015). Spominju se dva važna pristupa ovom problemu, a to su (i) ukomponirati invarijantnost u samu arhitekturu mreže, ili (ii) pažljivo pripremiti skupove za učenje i ispitivanje kako bi se umanjili negativni utjecaji neporavnanih modela. Primjer klasifikacije dubokom neuronskom mrežom pri kojoj se invarijantnost na rotaciju postiže pripremom skupa za učenje dan je u radu (Filipović et al., 2018). Iako se

ova dva pristupa koriste zajedno kako bi se poboljšale performanse, veliki su napori koji se ulažu u pripremu skupova za učenje i ispitivanje u svrhu poravnanja.

Više istraživačkih timova bavi se klasifikacijom manjih kućanskih objekata kojima robot može manipulirati (Wohlkinger i Vincze, 2011b,a; Wohlkinger et al., 2012; Bo et al., 2012; Blum et al., 2012; Asif et al., 2015; Dai et al., 2017; Patten et al., 2016; Hernández et al., 2016; Khan et al., 2016). Postoji i više skupova podataka koji sadrže RGB-D slike malih kućanskih objekata, od koji su dostupni *Large Dataset of Object Scans* (Choi et al., 2016), *3DNet* (Wohlkinger et al., 2012), *SUN RGB-D* (Song et al., 2015) i *RGB-D Object Dataset* (Lai et al., 2011). *ScanNet* (Dai et al., 2017) sadrži scene unutarnjih prostora te se također može svrstati u ovu grupu. Nedavni pregled dostupnih RGB-D skupova može se pronaći u (Firman, 2016). Za specifičnu primjenu predstavljenu ovim poglavljem doktorskog rada, upotrijebljen je *3DNet* (Wohlkinger et al., 2012) jer sadrži 3D modele objekata koji pripadaju različitim klasama u skupu za učenje te označene RGB-D slike objekata pripadnika istih klasa u skupu za ispitivanje. Modeli u *3DNet*-u nisu poravnani s gravitacijskom osi, tj. ne leže na horizontalnoj površini. Prednost je metode predstavljene u ovom poglavlju što ne zahtijeva takvo poravnanje te je prikladna za ispitivanje na *3DNet*-u i sličnim skupovima.

### 3.3 Poravnanje konveksnih ljuski

#### 3.3.1 Poravnanje sličnih oblika

Neka su  $O$  i  $O'$  dva skupa 3D točaka koji predstavljaju površine dvaju objekata. Sličnost ova dva skupa točaka mjeri se pomoću Chamferove udaljenosti (Barrow et al., 1977)

$$\Upsilon(O, O') = \sum_{p \in O} \min_{p' \in O'} \|p - p'\|^2. \quad (3.1)$$

Uz pretpostavku da  $O$  predstavlja kruto tijelo, rotacija i translacija ovog skupa ne bi trebala mijenjati sličnost između tog skupa i skupa  $O'$ . Nadalje, povećanje odnosno smanjenje tijela predstavljenog skupom točaka  $O$ , na način da se sve koordinate tih točaka pomnože istom vrijednošću također ne bi trebalo mijenjati mjeru sličnosti između skupova  $O$  i  $O'$ . Takva transformacija skupa točaka se u ovom radu naziva skaliranje. Funkcija troška (3.1) je prikladna mjera sličnosti između oblika  $O$  i  $O'$  samo ako su ispravno poravnani. Stoga,

definirana je druga mjera sličnosti na koju ne utječu translacija, rotacija i skaliranje.

$$\mathfrak{S}(O, O') = \sum_{p \in O} \min_{p' \in T^*(O')} \|p - p'\|^2, \quad (3.2)$$

pri čemu

$$T^* = \operatorname{argmin}_T f(O, T(O')). \quad (3.3)$$

Simbol  $T$  označava krutu transformaciju skupa točaka, a  $T^*$  predstavlja optimalnu transformaciju koja poravnava  $O$  i  $O'$ . U ovom je radu predložena efikasna metoda za određivanje suboptimalne transformacije za poravnanje,  $T^*$ , koja se zasniva na poravnanju konveksnih ljuski  $O$  i  $O'$ .

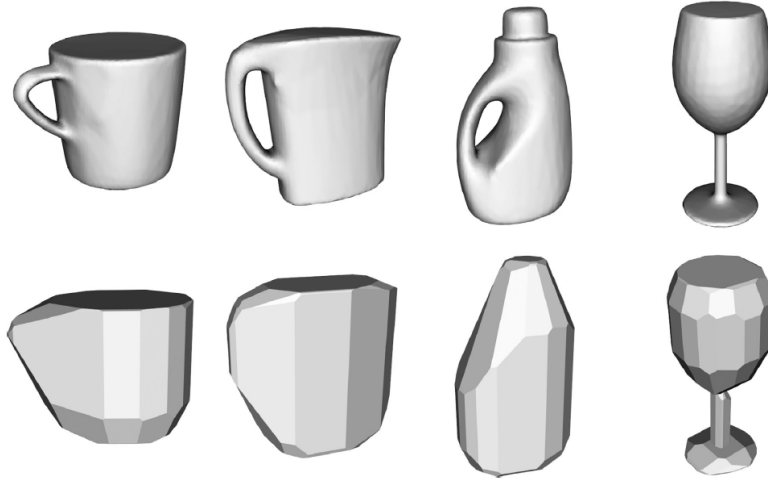
### 3.3.2 Sličnost konveksnih ljuski

Kako bi se postiglo efikasno poravnanje konveksnih ljuski dva objekta, koristi se CTI (engl. *Convex Template Instance*) deskriptor (Nyarko et al., 2018). Ideja ovog deskriptora je aproksimacija jednog konveksnog poliedra drugim poliedrom čije normale stranice pripadaju konačnom skupu jediničnih vektora. Neka je  $\mathbb{A}$  skup različitih jediničnih vektora  $a_i \in \mathbb{R}^3, i = 1, \dots, n$ . Ovaj skup vektora naziva se konveksni predložak (engl. *convex template*) koji definira skup svih konveksnih poliedara u 3D,  $\Theta$ , takvih da svaka stranica poliedra  $K \in \Theta$  je okomita na neki od vektora  $a_i \in \mathbb{A}$ . Poliedar koji pripada tom skupu naziva se instanca konveksnog predloška, CTI (engl. *convex template instance*). CTI nije u potpunosti konveksna ljuska objekta, ali je njezina približna aproksimacija. Svaka stranica poliedra  $K \in \Theta$  leži na ravnini definiranoj jednadžbom

$$a_i^\top p = d_i, \quad (3.4)$$

gdje  $a_i \in \mathbb{A}$  predstavlja normalu ravnine,  $p \in \mathbb{R}^3$  jest vektor koji predstavlja poziciju 3D točke u odnosu na određeni koordinatni sustav, a  $d_i$  je udaljenost od ravnine u odnosu na ishodište tog koordinatnog sustava. Dakle, svaki poliedar  $K \in \Theta$  je potpuno definiran skupom jediničnih vektora  $\mathbb{A}$  i vektorom  $d = [d_1, d_2, \dots, d_n]$ . Nadalje, neka je  $O$  skup 3D točaka  $p \in \mathbb{R}^3$  i neka je  $K(O)$  najmanji poliedar  $K \in \Theta$  takav da  $O \subseteq K(O)$ . Vektor  $d(O)$  koji opisuje CTI  $K(O)$  može se izračunati na sljedeći način. Neka je  $P$  matrica koja se sastoji od vektora točaka  $p \in O$  koji predstavljaju stupce matrice  $P$ , i neka je  $A = [a_1|a_2|\dots|a_n]$ .





Slika 3.2: 3D modeli (gornji redak) i odgovarajući CTI-evi (donji redak). Prva tri CTI-a predstavljaju cijeli oblik prva tri objekta, dok je četvrti objekt, čaša, predstavljena s tri CTI-a za svaki konveksni segment.

Tada je  $d(O)$  određen sljedećim izrazom

$$X = A^T P \quad (3.5)$$

i formiranjem vektora  $d(O)$  čije su komponente najveće vrijednosti redaka matrice  $X$ . Vektor  $d(O)$  naziva se CTI deskriptorom oblika  $O$ . Ovaj je deskriptor predložen u (Nyarko et al., 2018), gdje se koristi za prepoznavanje voća. Nekoliko primjera konveksnih oblika i odgovarajućih CTI-eva prikazani su na slici 3.2. Prikazani CTI-evi kreirani su za konveksni predložak koji se sastoji od  $n = 66$  jediničnih vektora približno jednoliko distribuiranih po jediničnoj kugli.

Reprezentacija skupa točaka njihovim CTI-evima,  $K(O)$ , dozvoljava usporedbu njihovih oblika uspoređivanjem pridruženih deskriptora  $d(O)$ . Euklidska udaljenost CTI deskriptora  $d$  i  $d'$  (3.6) predstavlja mjeru sličnosti dva CTI-a

$$e(d, d') = \|d - d'\|^2 \quad (3.6)$$

Međutim, analogno mjeri sličnosti oblika opisanoj u potpoglavlju 3.3.1, euklidska udaljenost između dva CTI deskriptora prikladna je mjera sličnosti samo ako su CTI-evi pravilno poravnani. Metoda za poravnanje konveksnih ljuski, tj. CTI-eva, predložena je u nastavku.

### 3.3.3 Optimalna translacija i skala

Pretpostavimo da je optimalna rotacija između dva skupa točaka,  $O$  i  $O'$  poznata. U tom slučaju, bez smanjenja općenitosti, možemo pretpostaviti da su  $O$  i  $O'$  već poravnani u

odnosu na tu optimalnu rotaciju te da je potrebno odrediti samo optimalnu translaciju i skalu. Neka je  $O''$  skup točaka dobiven skaliranjem skupa  $O'$  skalirajućim faktorom  $s$ . Lako je dokazati da je

$$d(O'') = sd(O') . \quad (3.7)$$

Neka je  $O''$  skup točaka dobiven translacijom  $O'$  translacijskim vektorom  $t$ . Tada je

$$d(O'') = d(O') + A^\top t . \quad (3.8)$$

Posljedično, ako je  $O''$  dobiven skaliranjem  $O'$  skalirajućim faktorom  $s$  i translacijom dobivenog skupa točaka translacijskim vektorom  $t$ , tada je

$$d(O'') = Qu, \quad (3.9)$$

gdje je

$$Q = [d(O')|A^\top] , \quad (3.10)$$

i  $u = [s|t^\top]$ . Optimalne translacija i skala određene su jednadžbom

$$u^* = \operatorname{argmin}_u e(d(O), Qu) \quad (3.11)$$

pri čemu je  $e$  određen jednadžbom (3.6). Ovo je standardna optimizacija metodom najmanjih kvadrata čije je rješenje

$$u^* = (Q^\top Q)^{-1} Q^\top d(O) . \quad (3.12)$$

Predstavljena analiza je osnova za definiranje mjere sličnosti konveksnih ljuski, invarijantne na translaciju i skalu. Ta je mjera sličnosti

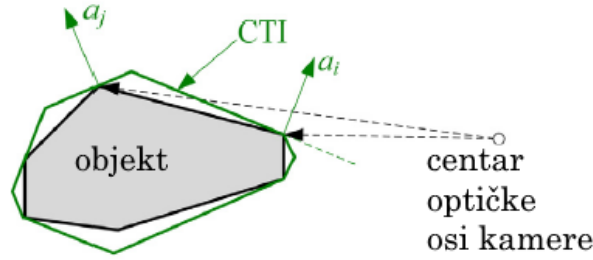
$$e_{ts}(d, d') = \|d - Qu^*\|^2 , \quad (3.13)$$

pri čemu su  $Q$  i  $u^*$  definirani jednadžbama (3.10) i (3.12). Supstitucijom (3.12) u (3.13), dobivena je sljedeća formulacija predložene mjere sličnosti

$$e_{ts}(d, d') = \|(I - Q(Q^\top Q)^{-1} Q^\top)d\|^2 , \quad (3.14)$$

gdje je  $I$  jedinična matrica.

U slučaju snimanja objekta 3D kamerom iz jednog pogleda, samo je dio njegove površine dobivenim oblakom točaka vidljiv. S obzirom da svaka komponenta CTI deskriptora pripada jednoj stranici poliedra, samo one komponente CTI deskriptora koje pripadaju stranicama



Slika 3.3: Poliedar (sivo) i CTI (zeleno) generiran na osnovi tog poliedra. Stranica  $i$  vidljiva je kameri, dok stranica  $j$  nije.

čije su normale orijentirane prema kameri uzimaju se u obzir. Te se komponente nazivaju vidljive komponente CTI deskriptora. Primjer je dan na slici 3.3. Kako bi se odredilo je li komponenta CTI deskriptora vidljiva, koristi se sljedeći kriterij. Ako komponenta CTI deskriptora, definirana u odnosu na koordinatni sustav kamere s ishodištem u njenom optičkom centru, ima negativnu vrijednost, onda je ta komponenta vidljiva. Vidljivi dio,  $d^v$ , CTI deskriptora  $d$  računa se pomoću

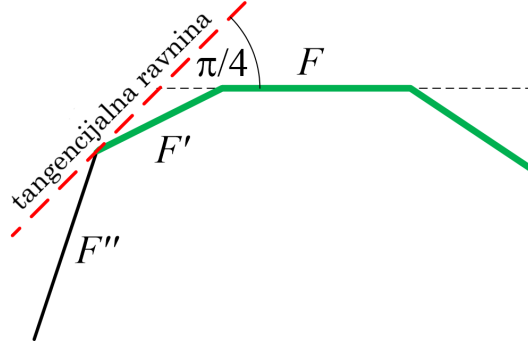
$$d^v = Zd \quad (3.15)$$

pri čemu je  $Z$  matrica vidljivosti s  $n_v$  redaka i  $n$  stupaca, čiji su elementi  $z_{ij} = 1$  ako  $i$ -ti element od  $d^v$  odgovara  $j$ -tom elementu od  $d$  i  $z_{ij} = 0$  u suprotnom. Samo vidljive komponente CTI deskriptora se mogu sparivati. U tom slučaju, koristi se matrica  $Q^v = ZQ$  umjesto matrice  $Q$  te  $d^v$  umjesto  $d$  u računanju optimalnog translacijskog vektora prikazanog jednadžbom (3.12). Dobiva se sljedeća formulacija mjere sličnosti

$$e_{ts}^v(d^v, d'^v) = \|(I - Q^v(Q^{v\top}Q^v)^{-1}Q^{v\top})d^v\|^2, \quad (3.16)$$

### 3.3.4 Optimalna rotacija

U ovom je potpoglavlju predstavljena metoda za određivanje rotacije koja poravnava dvije konveksne ljuske. Predložen postupak sastoji se od generiranja skupa prijedloga rotacija i odabiranja onog za koji je funkcija troška (3.14) najmanja. Prijedlozi rotacija generirani su prikazivanjem oblika skupom ravninskih segmenata i poravnanjem dominantnih ravninskih segmenata. Pretpostavimo da se poravnavaju poliedri, tj. oblici koji se sastoje od poligonalnih površina, koje se u ovom radu nazivaju stranicama, odijeljenih ravnim bridovima. Dva se oblika  $O$  i  $O'$  mogu poravnati tako da im se poravnaju odgovarajući koordinatni sustavi definirani njihovim stranicama. Neka je  $L$  koordinatni sustav generiran od para ravina, pri



Slika 3.4: 2D prikaz površine poliedra. Susjedstvo  $\mathcal{N}_{\pi/4}(F)$  stranice  $F$  označeno je zelenim linijama. Tangencijalna ravnina između stranica  $F' \in \mathcal{N}_{\pi/4}(F)$  i  $F'' \notin \mathcal{N}_{\pi/4}(F)$  predstavljena je crvenom isprekidanom linijom.

čemu na prvoj ravnini leži jedna stranica  $F$ , a druga ravnina je tangencijalna površini u susjedstvu  $\mathcal{N}_{\pi/4}(F)$  te stranice.

Susjedstvo  $\mathcal{N}_{\pi/4}(F)$  stranice  $F$  predstavlja povezani skup stranica koji sadrži  $F$  sa svojstvom da je kut između svake stranice  $F' \in \mathcal{N}_{\pi/4}(F)$  i  $F$  manji ili jednak  $\pi/4$ . Dvodi-menzionalni prikaz površine poliedra sa stranicom  $F$  i njezinim susjedstvom  $\mathcal{N}_{\pi/4}(F)$  dan je na slici 3.4. Tangencijalna ravnina na bridovima između susjednih stranica  $F'$  i  $F''$ , pri čemu  $F' \in \mathcal{N}_{\pi/4}(F)$  i  $F'' \notin \mathcal{N}_{\pi/4}(F)$  služi za definiranje koordinatnog sustava  $L$ . Postoji beskonačno velik broj tangencijalnih ravnina na bridovima između dviju stranica poliedra. S obzirom da je kut između  $F' \in \mathcal{N}_{\pi/4}(F)$  i  $F$  manji ili jednak  $\pi/4$  te da je kut između  $F'' \notin \mathcal{N}_{\pi/4}(F)$  i  $F$  veći od  $\pi/4$ , postoji tangencijalna ravnina na zajedničkom bridu stranica  $F'$  i  $F''$ , takva da je kut između te ravnine i ravnine na kojoj leži  $F$  točno  $\pi/4$ , kao što je prikazano na slici 3.4. Ova tangencijalna ravnina, zajedno s ravninom na kojoj leži  $F$ , služi za određivanje koordinatnog sustava  $L$  pri čemu normala stranice  $F$  predstavlja  $z$ -os tog koordinatnog sustava, a  $x$ -os je dobivena vektorskim produktom normala tih dviju ravnina. Pretpostavlja se da su stranicama poliedra dodijeljene normale koje su orijentirane od unutrašnjosti poliedra prema van te da su normalne tangencijalnih ravnina također tako orijentirane. Time su predznaci  $x$  i  $z$ -osi jednoznačno određeni.

Na ovaj je način definirano više koordinatnih sustava  $L_1, L_2, \dots, L_m$  za svaki oblik  $O$ , pri čemu su kao stranica  $F$  korištene samo relevantne stranice. Relevantne stranice su najveća stranica razmatranog poliedra i sve ostale stranice tog poliedra čija je veličina najmanje 4% veličine najveće stranice. Ukupni broj koordinatnih sustava po obliku ovisi o broju relevantnih stranica. Nadalje, s obzirom da postoji više kombinacija relevantnih stranica i ostalih stranica u susjedstvu, više je koordinatnih sustava generirano za svaku relevantnu

stranicu. Kako bi se ukupni broj koordinatnih sustava održao relativno malim, i samim time skratilo vrijeme izvođenja algoritma, samo je ograničeni broj tangencijalnih ravnina generiran od parova stranica  $(F', F'') \in \mathcal{N}_{\pi/4}(F)$  izabran za određenu relevantnu stranicu  $F$ , s obzirom na sljedeći kriterij. Krajnje točke zajedničkog brida stranica  $F'$  i  $F''$  projicirane su ortogonalno na presjek između ravnine na kojoj leži stranica  $F$  i tangencijalne ravnine definirane s  $F'$  i  $F''$ . Udaljenost  $\lambda$  između tako dobivenih točaka koristi se kao mjera na temelju koje su parovi stranica rangirani. Samo najviše rangiran par za određenu relevantnu stranicu i svi parovi čija je vrijednost  $\lambda$  najmanje 20% najveće vrijednosti  $\lambda$  razmatraju se za definiranje koordinatnih sustava  $L_i$ .

Optimalna rotacija između dva oblika  $O$  i  $O'$  pronalazi se definiranjem više koordinatnih sustava  $L_1, L_2, \dots, L_m$  za  $O$  te  $L'_1, L'_2, \dots, L'_{m'}$  za  $O'$ , računanjem CTI deskriptora u odnosu na svaki koordinatni sustav i uspoređivanjem dobivenih deskriptora. CTI deskriptor opisan u potpoglavlju 3.3.2 izračunava se na temelju koordinata točaka u odnosu na neki koordinatni sustav. Neka je  $d(O; L_i)$  CTI deskriptor skupa točaka  $O$  izračunat za svaki koordinatni sustav  $L_i$ . Nadalje, neka je  ${}^O R_{L_i} \in \text{SO}(3)$  rotacijska matrica koja definira orijentaciju  $L_i$  u odnosu na koordinatni sustav oblika  $O$ . Transformirane koordinate točke  ${}^{L_i}p$  dobivene pomoću

$${}^{L_i}p = {}^O R_{L_i}^\top \cdot p, p \in O, \quad (3.17)$$

koriste se za izračun komponenti deskriptora  $d(O; L_i)$ . Svaki par koordinatnih sustava  $(L_i, L'_j)$  predstavlja prijedlog poravnanja. Ti se prijedlozi evaluiraju funkcijom troška (3.14). Prijedlozi poravnanja s manjim troškom smatraju se bliže optimalnom poravnanju. Ako je poravnanje između koordinatnih sustava  $L_i$  i  $L'_j$  predstavljeno optimalnom rotacijom koja poravnava  $O'$  sa  $O$ , tada je optimalna transformacija koja poravnava ova dva oblika definirana homogenom matricom transformacije

$${}^O T_{O'}^{(i,j)} = {}^O T_{L_i} \quad {}^{L_i} T_{L'_j} \quad {}^{O'} T_{L'_j}^{-1}, \quad (3.18)$$

pri čemu

$${}^O T_L = \left[ \begin{array}{c|c} {}^O R_L & 0 \\ \hline 0 & 1 \end{array} \right], \quad (3.19)$$

$${}^{L_i} T_{L'_j} = \left[ \begin{array}{c|c} sI & t \\ \hline 0 & 1 \end{array} \right]. \quad (3.20)$$

Faktor skaliranja  $s$  i translacijski vektor  $t$  u (3.20) računaju se pomoću jednadžbe (3.12)

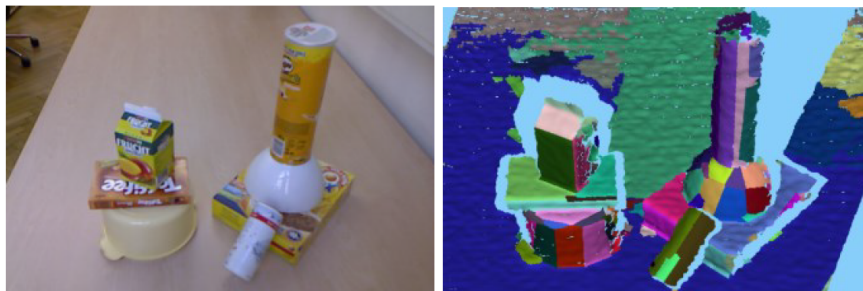
### 3.3.5 Približno poliedarski prikaz

U potpoglavlju 3.3.4 pretpostavljeno je da su oblici koji se poravnavaju poliedri. Međutim, predloženi postupak može se primijeniti i na objekte proizvoljnog oblika predstavljene 3D oblakom točaka ili gustom mrežom trokuta (engl. *dense triangular mesh*). Dubinska slika dobivena 3D senzorom poput RGB-D kamere ili LiDAR-a jednostavno se može transformirati u gustu mrežu trokuta (Holz i Behnke, 2012). U ovom je istraživanju primijenjen postupak predložen u (Cupec et al., 2017) koji segmentira gustu mrežu trokuta u ravninske segmente. Ova metoda, bazirana na širenju područja (engl. *region growing*), određuje granice između susjednih segmenata algoritmom pretraživanja koji favorizira granice poravante s pravcom presjeka ravnina na kojima leže susjedni ravninski segmenti. Na taj je način postignuta segmentacija, koja se može smatrati približno poligonalnim prikazom ulazne guste mreže trokuta. Točke u kojima se susreću tri susjedna segmenta predstavljaju dobru aproksimaciju vrhova (engl. *vertex*) poliedra, te su ovom radu nazivaju vrhovima. Oni se koriste za učinkoviti izračun CTI deskriptora jednadžbom (3.5), pri čemu je matrica  $P$  formirana samo od vrhova. S obzirom da je broj vrhova puno manji od ukupnog broja točaka ulazne guste mreže trokuta računsko vrijeme je značajno skraćeno. Primjer slike u boji i pripadajuće dubinske slike sa izdvojenim ravninskim segmentima dan je na slici 3.5 (Cupec et al., 2017).

## 3.4 Klasifikacija objekata

Poravnanje 3D oblika predstavljeno u potpoglavlju 3.3 predviđeno je za primjenu u klasifikaciji objekata prikazanih dubinskim slikama koristeći bazu 3D modela objekata, pri čemu svaki model pripada jednoj klasi. Ispitivani objekt predstavljen je s više konveksnih ljsuki definiranih u odnosu na različite koordinatne sustave, kao što je objašnjeno u potpoglavlju 3.3.4. Svaka od tih konveksnih ljsuki predstavljena je CTI deskriptorima, definiranim u potpoglavlju 3.3.2. Skup  $\mathbb{A}$  jediničnih vektora  $a_i$  upotrijebljenih pri definiranju CTI deskriptora dobiven je uniformnim uzorkovanjem jedinične kugle. U ovom radu korišten je skup od 66 takvih vektora. Međutim, pri usporedbi 3D modela sa objektom detektiranim na dubinskoj slici, samo vektori  $a_i$  koji su orijentirani prema kameri, razmatraju se pri izračunu funkcije troška (3.16), jer oni predstavljaju vidljive dijelove površine objekta.

3D modeli u bazi modela predstavljeni su deskriptorima  $d'$  koji se računaju koristeći



Slika 3.5: Slika u boji (lijevo) i pripadajuća dubinska slika (desno). Svijetlo plavom bojom označeni su dijelovi koji nisu vidljivi iz trenutnog pogleda kamere.

cjelokupni skup  $\mathbb{A}$ . Svakom je deskriptoru razmatranog objekta ili objekta iz baze pridružena pripadna rotacijska matrica  ${}^O R_L$ , opisana u potpoglavlju 3.3.4.

Predložena metoda klasifikacije sastoji se od dva koraka:

1. Generiranje prijedloga poravnanja između razmatranog objekta i svih modela iz baze i njihovo rangiranje na temelju funkcije troška (3.16).
2. Odabir unaprijed određenog broja najviše rangiranih prijedloga poravnanja i njihova evaluacija izračunom mjere sličnosti između poravnanih oblika. Donošenje konačne odluke o klasi razmatranog objekta ovisno o prijedlogu poravnanja s najvećom mjerom sličnosti.

U prvom je koraku trošak sparivanja (3.16) izračunat za svaki par  $(d_i, d'_j)$ , gdje  $d_i$  predstavlja  $i$ -ti deskriptor razmatranog oblika, a  $d'_j$  predstavlja  $j$ -ti deskriptor modela iz baze. Treba napomenuti da simbol  $d_i$  u potpoglavlju 3.3.2 predstavlja  $i$ -tu komponentu deskriptora, dok u ovom potpoglavlju predstavlja cijeli deskriptor.

Izabire se unaprijed zadani broj  $n_H$  najviše rangiranih parova  $(d_i, d'_j)$  za svaku klasu objekata te se za svaki od njih računa transformacija za poravnanje  ${}^{O'} T_{O'}^{(i,j)}$  jednadžbama (3.18)-(3.20), gdje su rotacijske matrice  ${}^O R_L$  pridružene deskriptorima  $d_i$  i  $d'_j$  upotrijebljene u (3.19), dok su faktor skaliranja  $s$  i translacijski vektor  $t$  u (3.20) dobiveni pomoću jednadžbe (3.12).

Kako bi se eliminali redundantni prijedlozi poravnanja, za svaku je klasu definiran referentni objekt te je transformacija koja poravnava razmatrani objekt s odgovarajućim referentnim objektom izračunata za svaki prijedlog poravnanja. S obzirom da prijedlog poravnanja predstavlja transformaciju  ${}^{O'} T_{O'}^{(i,j)}$ , koja poravnava model  $O'$  iz baze s razmatranim objektom  $O$ , kompozicija te transformacije i transformacije  ${}^{O'} T_{O'}^{(i,j)}$  koja poravnava model  $O'$  sa referentnim modelom iz iste klase  $O'_r$  daje transformaciju  ${}^{O'} T_{O'_r}^{(i,j)}$  koja poravnava referentni

model  $O'_r$  s razmatranim oblikom  $O$ . Ova se transformacija koristi kako bi se odlučilo je li razmatrani prijedlog poravnanja redundantan ili ne. Ako je transformacija  ${}^O T_{O'_r}^{(i,j)}$  pridružena prijedlogu poravnanja slična transformaciji već pridruženoj drugom prijedlogu poravnanja, tada se prijedlog s većim troškom sparivanja odbacuje kao redundantan. Sličnost između dvije transformacije  $T$  i  $T'$  mjeri se računanjem relativne rotacije na sljedeći način. Iz transformacijskih matrica se izdvoje rotacijske matrice  $R$  i  $R'$ . Izračuna se rotacijska matrica koja predstavlja relativnu orijentaciju dva prijedloga poravnanja  $R'' = R \cdot R'^{\top}$ .  $R''$  se može prikazati kao rotacija za neki kut oko neke osi (engl. *angle-axis representation*). Taj kut predstavlja razliku orijentacije dva prijedloga poravnanja. Ako je kut te rotacije veći ili jednak unaprijed određenom pragu  $\theta_{max}$ , tada se prijedlozi poravnanja smatraju sličnima. Za objekte koji nisu simetrični, jedna je orijentacija po klasi uglavnom dostatna pa bi se za takve objekte mogao razmatrati samo najviše rangirani prijedlog poravnanja, dok bi se ostali mogli odbaciti. U tom slučaju, prag  $\theta_{max}$  bi se mogao postaviti na maksimalnu moguću vrijednost. Međutim, za objekte simetričnih oblika, pridruživanje konveksnih ljuski često rezultira netočnim prijedlozima poravnanja kod kojih je objekt rotiran za  $\pi$  rad u odnosu na ispravnu orijentaciju. Npr. na temelju konveksne ljuske cilindrične čaše nije moguće odrediti je li okrenuta naopako. Stoga je izabran prag  $\theta_{max} = \pi/2$ . Referentni model je izabran za svaku klasu tijekom treninga, poravnanjem svih modela iste klase sa svima koristeći postupak opisan u potpoglavlju 3.3 i računajući trošak sparivanja tih poravnanja. Model čija je suma troškova poravnanja sa svim ostalim modelima iste klase najmanja, izabran je kao referentni model te klase.

U drugom su koraku prijedlozi poravnanja izračunati u prvom koraku korišteni pri klasifikaciji razmatranog objekta. Svaki prijedlog poravnanja,  $h$ , evaluira se računanjem mjere sličnosti između razmatranog oblika i modela s kojim je poravnan. Ova se mjera sastoji od tri dijela, analogno onoj predstavljenoj u (Aldoma et al., 2012). Prvi dio naziva se utemeljenost hipoteze (engl. *hypothesis support*), koji mjeri koliko dobro poravnani model opisuje točke scene koje predstavljaju razmatrani objekt. Računa se projiciranjem poravnalog modela na sliku uporabom tehnike  $z$ -spremnika (engl. *z-buffer*). Neka skup  $Z_h$  predstavlja vidljive točke modela projiciranog na sliku. Utemeljenost hipoteze računa se dohvaćajući skup točaka  $Z_h$  i računanjem sume

$$\Omega(h) = \sum_{p \in Z_h} \omega(p), \quad (3.21)$$



gdje je  $\omega(p)$  doprinos svake točke  $p$  iz skupa  $Z_h$  sigurnosti hipoteze. Računa se na sljedeći način

$$\omega(p) = \begin{cases} (1 - \frac{\|p - p'\|_2}{\rho_e}) \max(n_p^\top n_{p'}, 0), & \|p - p'\|_2 \leq \rho_e \\ 0, & \text{inače} \end{cases} \quad (3.22)$$

gdje  $p'$  predstavlja točku slike u 9-susjedstvu točke  $p$ , koja je najbliža točki  $p$ ,  $n_p$  i  $n_{p'}$  su normale lokalnih površina u točkama  $p$  i  $p'$ , a  $\rho_e$  je parametar koji zadaje korisnik. Ako je udaljenost točke modela najbližoj točki scene veća od  $\rho_e$ , tada ta točka modela nije pridružena niti jednoj točki razmatranog objekta te ne povećava vrijednost poravnanja.

Drugi dio mjere sličnosti računa se projiciranjem modela na scenu i prebrojavanjem točaka modela koje zaklanjaju točke scene. Pretpostavlja se da razmatrani objekti nisu prozirni. Stoga, točke modela transformirane na scenu ne bi smjele biti bliže kameri od točaka scene s jednakim koordinatama na slici. Skup točaka koji krši ovu pretpostavku, tj. koji zaklanja točke slike, označava se s  $\Phi_h$ .

Treći dio jest udaljenost poravnalog modela od potporne horizontalne ravnine. Pretpostavka je da je objekt smješten na površini potporne ravnine. Stoga su penalizirani prijedlozi poravnanja za koje je poravnani model djelomično ili u potpunosti ispod potporne ravnine ili lebdi iznad nje. Ovo se postiže pronalaženjem najnižeg vrha transformiranog modela u odnosu na potpornu ravninu i računanjem udaljenosti  $f_{SP}(h)$  te točke i potporne ravnine. Iako ova pretpostavka predstavlja ograničenje predložene metode, situacije u kojima objekti leže na horizontalnim površinama poput stola ili poda česte su u svakodnevnim okruženjima poput kućanstva, što i predstavlja primjenu ovog istraživanja. Ova je pretpostavka također korištena u metodama prepoznavanja objekata predstavljenim u (Aldoma et al., 2016; Cupec et al., 2020).

Ukupna vrijednost podudaranja sa scenom (engl. *scene fitting score*), dana je izrazom

$$\Psi(h) = \Omega(h)(1 - \gamma^2 f_{SP}^2(h)) - \lambda |\Phi_h|, \quad (3.23)$$

pri čemu su težine  $\gamma$  i  $\lambda$  korisnički definirane. Prijedlog poravnanja s najvećom vrijednošću podudaranja sa scenom odabire se kao konačni rezultat predložene metode za klasifikaciju objekata te je razmatranom objektu dodijeljena klasa modela s kojim je poravnan.

## 3.5 Eksperimentalna evaluacija

Predstavljena metoda implementirana je koristeći biblioteku Point Cloud Library (Rusu i Cousins, 2011) u programskom jeziku C++ te je ispitana na 3DNet skupu podataka (Wohlkinger et al., 2012). Ovaj se skup podataka sastoji od skupa 351 modela pripadnika jedne od 10 klasa. Ispitni skup sastoji se od 1652 scene snimljene RGB-D kamerom. Na 40 scena postojala je greška te nisu razmatrane tijekom evaluacije, tako da je klasifikacija objekata provedena na 1612 ispitnih (testnih) scena. S obzirom da predložena metoda za segmentaciju zahtijeva uniformno uzorkovane modele, modeli iz baze su predobrađeni u programu Meshlab (Cignoni et al., 2008).

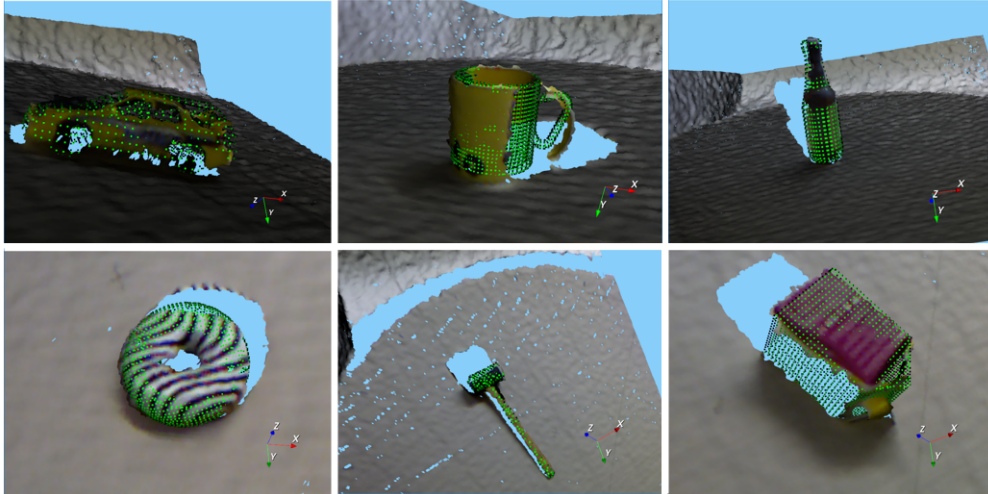
Tijekom faze učenja, modeli su predstavljeni CTI deskriptorima, kao što je objašnjeno u potpoglavlju 3.3. Više je koordinatnih sustava definiranih za svaki model koristeći postupak opisan u potpoglavlju 3.3.4 te je za svaki koordinatni sustav određen deskriptor kao što je opisano u potpoglavlju 3.3.2. Baza modela sastoji se od ukupno 36052 deskriptora.

Za svaku ispitnu scenu kreirani su CTI deskriptori koristeći isti postupak. S obzirom da segmentacija razmatranog objekta od pozadine nije tema ovog istraživanja, pretpostavlja se da je segmentacijska maska, koja pridružuje svaku točku scene ili s razmatranim objektom ili s pozadinom, dostupna za svaku ispitnu scenu. Ovo se pretpostavlja i u metodama s kojima se predloženi postupak uspoređuje. Deskriptori razmatranog objekta uspoređuju se sa svim deskriptorima svih modela iz baze računanjem troška podudaranja (3.16). Za svaku klasu,  $n_H$  najboljih podudaranja je izabrano te je za svaki od njih izračunata transformacija poravnanja, kao što je opisano u potpoglavlju 3.4. Ovi prijedlozi poravnanja su rangirani na temelju vrijednosti podudaranja sa scenom (3.23) te se odabire najbolje rangirani prijedlog poravnanja. Primjeri najviše rangiranih prijedloga poravnanja za šest ispitnih scena prikazani su na slici 3.6, gdje su modeli iz baze poravnani s razmatranim objektom prikazanim zelenim oblacima točaka. Konačno rješenje predložene metode sadrži klasu modela pridruženu odabranom prijedlogu poravnanja te pripadajuću transformaciju poravnanja. Vrijednosti parametara koje se koriste u prikazanim pokusima dane su u tablici 3.1. Vrijeme izvedbe neoptimiziranog algoritma po sceni iznosi 3.43 [s].

Kvantitativni rezultati klasifikacije objekata postignuti predloženom metodom dani su u zadnjem stupcu tablice 3.2 gdje su označeni sa CHASFS (Convex Hull Alignment + Scene Fitting Score), zajedno s rezultatima druge četiri metode na istom skupu podataka Wohlkinger et al. (2012). Te su metode zasnovane na deskriptorima: Viewpoint Feature Histogram

Tablica 3.1: Parametri klasifikacije objekata

| $n_H$ | $\rho_e$ | $\gamma$ | $\lambda$ |
|-------|----------|----------|-----------|
| 15    | 10 mm    | 0.6      | 30        |



Slika 3.6: Primjeri ispravno poravnanih modela s razmatranim objektima

(VFH) (Rusu et al., 2010), Clustered Viewpoint Feature Histogram (OVFH) (Aldoma et al., 2011), the Ensemble of Shape Functions (ESF) (Wohlkinger i Vincze, 2011a; Patten et al., 2016) i SHOT deskriptor (Salti et al., 2014; Wohlkinger et al., 2012). Predloženi je postupak klasifikacije dao najbolje rezultate za dvije klase i ukupno najbolji rezultat. Treba napomenuti da predložena metoda pretpostavlja da objekti leže na horizontalnim površinama, dok ostale metode nemaju tu pretpostavku.

Zanimljiv je rezultat i uspješnost klasifikacije postignute samo na temelju poravnanja CTI-a, prikazan u predzadnjem stupcu tablice 3.2, označen sa CHA. Ovi su rezultati postignuti nakon prvog koraka procedure opisane u potpoglavlju 3.4 i uzimajući klasu modela s najmanjim troškom pridruživanja (3.16) kao konačan rezultat, bez računanja vrijednosti podudaranja sa scenom. Ovaj se postupak može zvati klasifikacija na temelju sličnosti CTI deskriptora. Iako tako postignuti rezultati nisu među najboljima prikazanima u tablici 3.2, usporediva je uspješnost postignuta za neke klase. Na primjer, za boce i aute, rezultati postignuti ovom metodom su drugi najbolji. Međutim, kako bi se razlikovali objekti pripadnici različitih klasa sličnih konveksnih ljuski, drugi je korak klasifikacije nužan.

Ovaj je zaključak potvrđen matricama konfuzije, prikazanima slikom 3.7. Prva matrica predstavlja rezultate postignute klasifikacijom na temelju sličnosti konveksnih ljuski (CHA), a druga matrica predstavlja rezultate dobivene cjelokupnom metodom (CHASF). Vidljivo

Tablica 3.2: Rezultati klasifikacije objekata [%]

|                | VFH | CVFH       | ESF        | SHOT       | CHA | CHASFS    |
|----------------|-----|------------|------------|------------|-----|-----------|
| jabuka         | 64  | 95         | <b>99</b>  | 93         | 10  | 91        |
| banana         | 97  | <b>99</b>  | 89         | 73         | 79  | 95        |
| boca           | 84  | 87         | 94         | <b>99</b>  | 96  | 93        |
| zdjela         | 97  | 97         | 88         | <b>100</b> | 88  | 99        |
| auto           | 87  | 91         | <b>98</b>  | 95         | 97  | 95        |
| krofna         | 78  | <b>96</b>  | <b>96</b>  | 90         | 54  | 86        |
| čekić          | 93  | 98         | <b>100</b> | 97         | 95  | 97        |
| šalica         | 98  | <b>100</b> | <b>100</b> | 95         | 96  | 94        |
| tetrapak       | 5   | 5          | 91         | 74         | 69  | <b>96</b> |
| toaletni papir | 20  | 11         | 61         | 58         | 41  | <b>84</b> |
| UKUPNO         | 72  | 78         | 92         | 87         | 73  | <b>93</b> |

|          | jabuka | banana | boca | zdjela | auto | krofna | čekić | šalica | tetrapak | t. papir |
|----------|--------|--------|------|--------|------|--------|-------|--------|----------|----------|
| jabuka   | 10     | 0      | 0    | 80     | 0    | 8      | 0     | 2      | 0        | 0        |
| banana   | 0      | 79     | 3    | 0      | 1    | 0      | 17    | 0      | 0        | 0        |
| boca     | 0      | 0      | 96   | 0      | 2    | 0      | 2     | 0      | 0        | 0        |
| zdjela   | 0      | 0      | 6    | 88     | 4    | 2      | 0     | 0      | 0        | 0        |
| auto     | 0      | 0      | 2    | 0      | 97   | 0      | 0     | 1      | 0        | 0        |
| krofna   | 0      | 0      | 2    | 44     | 0    | 54     | 0     | 0      | 0        | 0        |
| čekić    | 0      | 1      | 3    | 0      | 1    | 0      | 95    | 0      | 0        | 0        |
| šalica   | 1      | 0      | 0    | 1      | 1    | 0      | 0     | 96     | 0        | 1        |
| tetrapak | 0      | 0      | 7    | 0      | 23   | 0      | 1     | 0      | 69       | 0        |
| t. papir | 0      | 0      | 2    | 0      | 0    | 0      | 0     | 57     | 0        | 41       |

a

|          | jabuka | banana | boca | zdjela | auto | krofna | čekić | šalica | tetrapak | t. papir |
|----------|--------|--------|------|--------|------|--------|-------|--------|----------|----------|
| jabuka   | 91     | 0      | 1    | 8      | 0    | 0      | 0     | 0      | 0        | 0        |
| banana   | 0      | 95     | 0    | 0      | 0    | 0      | 5     | 0      | 0        | 0        |
| boca     | 0      | 0      | 93   | 0      | 4    | 0      | 1     | 0      | 2        | 0        |
| zdjela   | 0      | 0      | 0    | 99     | 0    | 0      | 1     | 0      | 0        | 0        |
| auto     | 0      | 0      | 2    | 0      | 95   | 0      | 0     | 0      | 3        | 0        |
| krofna   | 0      | 0      | 0    | 14     | 0    | 86     | 0     | 0      | 0        | 0        |
| čekić    | 1      | 0      | 0    | 0      | 1    | 0      | 97    | 0      | 1        | 0        |
| šalica   | 0      | 1      | 1    | 0      | 1    | 0      | 2     | 94     | 0        | 1        |
| tetrapak | 0      | 0      | 1    | 0      | 1    | 0      | 1     | 1      | 96       | 0        |
| t. papir | 0      | 0      | 0    | 0      | 0    | 0      | 0     | 16     | 0        | 84       |

b

Slika 3.7: Matrica konfuzije nakon poravnanja objekata (a) i nakon cjelokupne klasifikacije objekata (b)

je da su klase koje se najčešće miješaju CHA metodom jabuke i zdjele, jer njihovi vidljivi dijelovi na većini scena imaju slične konveksne ljuske. Međutim, drugi korak klasifikacije, koji uzima u obzir cijeli oblik objekta, rješava taj problem.

### 3.6 Zaključak

U ovom je poglavlju istražena mogućnost uporabe metode evaluacije hipoteza, koja se uglavnom koristi za detekciju instanci oblika, u klasifikaciji objekata. Postupak zahtijeva ispravno poravnanje između razmatranog objekta i modela objekata koji pripadaju određenim klasama. Predložen je novi algoritam za poravnanje 3D oblika zasnovan na konveksnim ljuskama. Razmatrana je metoda evaluirana na 3DNet podatkovnom skupu pri čemu postiže usporedive rezultate sa četiri značajne metode razvijene za istu primjenu. Potrebno je na-

pomenuti da baza modela razmatranog podatkovnog skupa sadrži objekte koji su vrlo slični objektima na sceni, u ispitnom podskupu, što je ključno za uspješnu primjenu ovog postupka. Uspješnost metode veća je od 90% za osam od deset klasa, ali i dalje postoji prostor za poboljšanja. Većina pogrešne klasifikacije događa se u slučajevima kada je vidljivi dio razmatranog objekta sličan za objekte pripadnike različitih klasa. Ovo bi se moglo riješiti klasifikacijom iz više pogleda. Nadalje, nedostatak metode jest što računaska složenost linearno raste s brojem modela u bazi. Primjena indeksiranja doprinijela bi rješenju tog problema. Konačno, klasifikacija na složenim scenama s okluzijom je i dalje izazovan problem.

U budućnosti je moguće primijeniti predloženu metodu za planiranje hvatanja gdje bi željeno hvatanje bilo definirano za referentni model određene klase te transformirano na objekt na sceni na temelju dobivenog položaja objekta predloženom metodom. Optimizacijskom tehnikom hvatanje bi se moglo prilagoditi za razmatrani objekt, pri čemu bi se ograničavanjem optimizacije na malo lokalno susjedstvo ostvarila računaska efikasnost planiranja hvatanja.

# 4

## Klasifikacija objekata na dubinskim slikama zasnovana na poliedarskim modelima klasa oblika

U ovom je poglavlju opisan treći doprinos ovog doktorskog rada, klasifikacija objekata na dubinskim slikama zasnovana na poliedarskim modelima klasa oblika. U poglavlju 2 predložen je začetak ideje određivanja parametara oblika u svrhu robotske manipulacije, a u ovom je poglavlju opisano unaprjeđenje takve metode. Metoda na kojoj se zasniva predložena klasifikacija, objavljena u (Cupec i Đurović, 2018), predstavlja novi pristup za modeliranje 3D klasa oblika koji se zasniva na aproksimaciji 3D oblika poliedrima dobivenim unijama ili presjecima konveksnih i konkavnih skupova. Modeli klasa oblika stvoreni su na temelju uzoraka za trening u obliku 3D mreže trokuta. Ovi se modeli koriste za klasifikaciju objekata na dubinskim snimkama i određivanje njihovog položaja i parametara oblika. Ova svojstva omogućuju adaptaciju robotske manipulacije definirane za referentni model neke klase objekata na ostale pripadnike iste klase. Za razliku od metode planiranja hvatanja, predložene u zaključku poglavlja 3, zasnovane na lokalnom pretraživanju površine klasificiranog objekta, pristup predložen u ovom poglavlju određuje točne parametre hvatanja, poput položaja hvataljke i širine razmaka između prstiju hvataljke. Primjena predložene metode ispitana je na odgovarajućem ispitnom skupu podataka te ostvaruje najbolje rezultate u usporedbi s četiri suvremena deskriptora 3D oblika namijenjena za klasifikaciju objekata.

## 4.1 Uvod

U budućnosti se očekuje rad robota u dinamičkim i nestrukturiranim okolinama te manipulacija objektima koje vide po prvi puta, na temelju znanja o sličnim objektima stečenima tijekom procesa učenja. Robot koji zna kako provesti određenu akciju s objektom predstavnikom neke klase objekata, trebao bi znati tu akciju prilagoditi ostalim objektima pripadnicima iste klase. Kako bi se ostvarila ta sposobnost, sustav robotskog vida osim klase objekta, treba prepoznati i njegove dijelove relevantne za obavljanje nekog postavljenog zadatka. Iako je već predložen velik broj metoda koje rješavaju problem klasifikacije objekata, većina takvih metoda ne određuje položaj objekta i njegove parametre koji su korisni u robotskoj manipulaciji. U svrhu manipulacije, potreban je parametarski model oblika. U ovom se radu predlaže novi postupak za modeliranje klasa 3D oblika, koji se zasniva na aproksimaciji oblika poliedrima opisanim deskriptorima čije komponente odgovaraju stranicama poliedra. U radu (Cupec i Đurović, 2018) predlaže se novi parametarski model klasa 3D oblika, naziva VolumeNet, koji dopušta veliku raznolikost oblika unutar klase te uspostavljanje korespondencija između semantički odgovarajućih dijelova površine objekata iste klase. Doprinos ove disertacije, opisan u ovom poglavlju je primjena VolumeNet-a u klasifikaciji objekata uz ostvarene rezultate usporedive s drugim uspješnim metodama. Predlaže se metoda za kreiranje modela klasa 3D oblika na temelju skupa uzoraka za učenje.

## 4.2 Istraživanja u ovom području

Problem klasifikacije objekata u jednu od prethodno naučenih klasa objekata istražuje velik broj timova. Pregled pristupa ovom problemu predloženih u literaturi dan je u poglavlju 3. Metoda predložena u ovom poglavlju ima sličnosti s metodama koje se zasnivaju na generiranju 3D deskriptora oblika obradom 3D snimke objekta na sceni te primjeni prikladnog klasifikatora da se tome deskriptoru dodijeli najvjerojatnija klasa. Međutim, navedeni postupci, kao i danas sve češće primjenjivane metode zasnovane na neuronskim mrežama, ne pružaju informacije o položaju objekta niti o parametrima oblika, koji su potrebni za prilagodbu robotske akcije na zadani razmatrani objekt. U tu svrhu, prikladniji su pristupi koji prilagođavaju parametarski predložak modela tako da se podudara s ispitivanim objektom. Takvi pristupi su primarno razvijeni za rekonstrukciju 3D scena (Huang et al., 2015) iz 2D slika ili oblaka točaka (Nan et al., 2012). Metoda predložena u (Nan et al., 2012) detektira

objekte iz nekoliko klasa, primarno stolice i stolove, na 3D skenovima složenih scena unutar njih prostora. Ulazni oblak točkaka je nadsegmentiran u glatke dijelove. Tada je primijenjen postupak širenja područja (engl. *region growing*), koji nakuplja susjedne dijelove u veće segmente. Nakon što je taj postupak završen, deformabilni predložak je postavljen na dobiveni segment. Dijelovi predloška se prilagođavaju snimljenom oblaku točkaka tako da se minimizira euklidsku udaljenost od točkaka segmenta do prilagođenog predloška. Proces prilagodbe oblika jedne klase traje prosječno 10 sekundi. Algoritam pretpostavlja da je objekt na sceni orijentiran prema gore te da je proces podudaranja s predloškom inicijaliziran poravnanjem predloška u toj orijentaciji sa segmentom. Nadalje, nije jasno kako je model predloška parametriziran. Postupak koji je predložen u ovom poglavlju dozvoljava proizvoljnu orijentaciju razmatranog objekta u 3D prostoru. Predloženi postupak stvara model klase oblika na temelju skupa za učenje koji se sastoji od 3D modela objekata zadane klase koji također mogu biti proizvoljno orijentirani. U radu (Mescheder et al., 2019) predložen je postupak za 3D rekonstrukciju koji se zasniva na kontinuiranoj 3D funkciji popunjenosti (engl. *occupancy function*). Funkcija svakoj točki prostora pridružuje vrijednost, ovisno o tome pripada li razmatrana točka unutrašnjosti ili površini objekta ili ne pripada objektu. Metoda opisana u ovom poglavlju zasniva se na VolumeNet modelu predloženom u (Cupec i Đurović, 2018), koji definira implicitnu funkciju oblika, koja također dodjeljuje vrijednosti svim točkama u prostoru, a kao i funkcija popunjenosti, ima vrijednost 0 na površini razmatranog objekta.

## 4.3 Model klase oblika

### 4.3.1 VolumeNet

Trodimenzionalni oblik može se predstaviti funkcijom  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$  koja ima vrijednost 0 za sve točke na površini, pozitivne vrijednosti za sve točka izvan i negativne vrijednosti za sve točke unutar razmatrane površine. Ova se funkcija u ovom doktorskom radu naziva *implicitna funkcija oblika* (engl. *implicit shape function*) (ISF).

Svaki konveksni poliedar  $O \in \mathbb{R}^3$  može se definirati kao presjek poluravnina predstavljena skupom jednadžbi

$$a_i^\top p \leq d_i, \quad i = 1, 2, \dots, n. \quad (4.1)$$

pri čemu  $p \in \mathbb{R}^3$ ,  $a_i$  je jedinični vektor koji predstavlja normalu  $i$ -te ravnine, a  $d_i$  predstavlja



udaljenost te ravnine od ishodišta koordinatnog sustava u kojem je predstavljena točka  $p$ . Oblik poliedra dobiven presjekom  $n$  poluravnina može se opisati sljedećom ISF

$$f(p) = \max_i(a_i^\top p - d_i). \quad (4.2)$$

Analogno tomu, unija poluravnina predstavljena skupom jednadžbi (4.1) je ili  $\mathbb{R}^3$  ili komplement konveksnog skupa opisan sljedećom ISF

$$f(p) = \min_i(a_i^\top p - d_i). \quad (4.3)$$

Treba napomenuti da se široka klasa oblika može aproksimirati do proizvoljne preciznosti poliedrima dobivenima presjecima i unijama konveksnih poliedara i komplementima konveksnih poliedara (Ricci, 1973).

Neka su  $f_1(p)$  i  $f_2(p)$  ISF kojima su opisana dva poliedra. Obik dobiven presjekom ili unijom ta dva poliedra može se opisati sljedećom ISF

$$f(p) = o \cdot \max(o \cdot f_1(p), o \cdot f_2(p)), \quad (4.4)$$

pri čemu je parametar  $o$  jednak 1 u slučaju presjeka, tj.  $-1$  u slučaju unije.

U ovom se radu predlaže model klasa oblika koji se zasniva na jedinstvenom opisu svih poliedara dobivenih kombinacijom unija i presjeka konveksnih poliedara i komplementa konveksnih poliedara. Ovaj se model naziva *VolumeNet*. *VolumeNet* (VN) je funkcija definirana s

$$f_i(p; d) = \begin{cases} a_i^\top p - d_i, & i \leq n \\ o_{i-n} \max_{j \in J_i}(o_{i-n} f_j(p; d)), & n < i \leq r, \end{cases} \quad (4.5)$$

$$f(p; d) = f_r(p; d) \quad (4.6)$$

pri čemu  $r$  predstavlja ukupan broj čvorova,  $J_i$  predstavlja skup indeksa dodijeljenih funkciji  $f_i$  tako da

$$\forall j \in J_i, \quad j < i,$$

$o_{i-n} \in \{-1, 1\}$  su parametri dodijeljeni svim čvorovima s indeksima  $n < i \leq r$  i  $d = [d_1, d_2, \dots, d_n]^\top$ . Naziv *VolumeNet* dolazi od strukture funkcije  $f$  koja je slična umjetnim neuronskim mrežama (engl. *neural network*) te zato što VN opisuje volumen u 3D prostoru dodijeljivanjem vrijednosti  $f(p; d) \leq 0$  svim točkama koje pripadaju tom volumenu.

Struktura VN potpuno je opisana matricom  $A$ , vektorom  $o$  i matricom  $W$ . Matrica  $A$  je  $n \times 3$  matrica koju čine vektori  $a_i$ , gdje je  $i$ -ti red matrice  $A$  jednak  $a_i^\top$  za  $i = 1, \dots, n$ . Vektor  $o$  sastoji se od parametara  $o_i, i = 1, \dots, r - n$ . Matrica  $W$  je  $r \times r$  matrica čiji elementi  $w_{ij}$  mogu imati vrijednost 0 ili 1. Element  $w_{ij} = 1$  ako je  $j \in J_i$ , a inače  $w_{ij} = 0$ .

Dvodimenzionalni primjer VN prikazan je na slici 4.1. Predstavljeni VN sastoji se od  $r = 10$  čvorova, pri čemu svaki čvor predstavlja funkciju  $f_i(p; d)$  definiranu s (4.6). Ulazni čvorovi ovog VN-a su dvije koordinate točke  $p = [x, y]^\top$ , dok je izlaz iz VN-a predstavljen vrijednošću ISF dodijeljenoj točki  $p$ . Postoje dva tipa čvorova, *čvorovi stranica*, sa indeksima  $i \leq n$  i *čvorovi operacija*, s indeksima  $n < i \leq r$ . Svaki čvor stranice odgovara stranici poliedra prikazanog razmatranim VN-om. On zapravo definira poluravninu ograničenu ravninom na kojoj leži jedna stranica poliedra. Svaki operacijski čvor predstavlja presjek ili uniju između volumena predstavljenih dvama čvorovima. Ako  $i$ -ti operacijski čvor predstavlja presjek, tada  $o_i = 1$ , inače  $o_i = -1$ . Podskup od  $\mathbb{R}^2$  definiran svakim čvorom prikazan je na slici 4.1 iznad tog čvora.

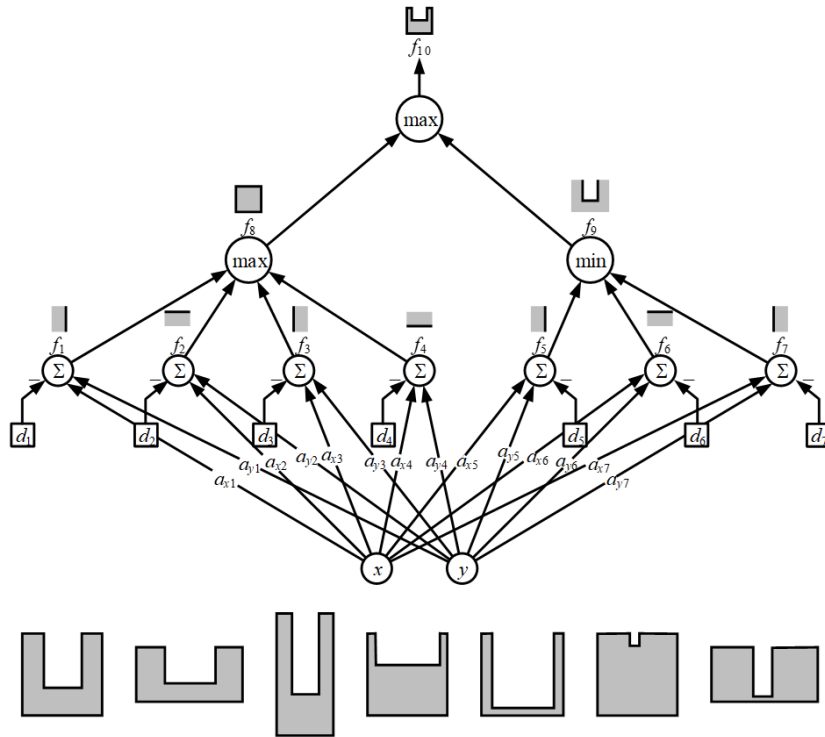
Ukupan broj čvorova, veze između njih definirane matricom  $W$ , operacije definirane vektorom  $o$  i orijentacije stranica definirane matricom  $A$  predstavljaju *strukturu* VN-a, dok parametri  $d_i$  predstavljaju *variable*. Nekoliko instanci klase 2D oblika predstavljenih VN-om prikazanim na slici 4.1 prikazano je na dnu slike 4.1. Sve instance oblika dobivene su jednakim parametrima  $a_1 = a_5 = [1, 0]^\top, a_2 = a_6 = [0, 1]^\top, a_3 = a_7 = [-1, 0]^\top, a_4 = [0, -1]^\top$  i različitim parametrima  $d_i$  za svaki oblik. Stoga, VN predstavlja parametarski model klase oblika, gdje je svaka instanca te klase oblika određena vektorom  $d = [d_1, \dots, d_n]^\top$ , koji se u ovom radu naziva *VN deskriptor*.

Radi jednostavnosti objašnjenja, VN model na slici 4.1 ilustriran je 2D primjerom. Međutim, osnovna je motivacija za razvoj VN-a modeliranja klasa 3D oblika. Pet primjera 3D oblika modeliranih VN-ima prikazano je na slici 4.2.

### 4.3.2 Prostor oblika

Prostor svih VN deskriptora  $d$  jest  $n$ -dimenzionalni prostor, u kojem svaka točka predstavlja jedan oblik. Taj se prostor u ovom radu naziva *prostor oblika*.

Neka je klasa oblika određena VN-om i neka je  $d$  deskriptor instance  $O$  ove klase oblika. Analogno objašnjenju opisanom u potpoglavlju 3.4, može se pokazati da se skaliranjem



Slika 4.1: Primjer VolumeNet strukture i nekoliko instanci 2D oblika klase predstavljene tom strukturom



Slika 4.2: Skup 3D modela, od kojih svaki pripada klasi oblika (gornji redak) i njihovi pripadajući 3D volumeni modelirani VN-om (donji redak).

instance  $O$  faktorom  $s$  i translacijom dobivenog oblika vektorom  $t$  mijenja njezin deskriptor

$$d' = sd + At, \quad (4.7)$$

pri čemu je matrica  $A$  definirana u potpoglavlju 4.3.1. Skup svih vektora  $d'$  dobivenih jednadžbom (4.7) za  $s \in \mathbb{R}$  i  $t \in \mathbb{R}^3$  predstavlja 4-dimenzionalni potprostor u  $n$ -dimenzionalnom prostoru oblika, koji sadrži sve oblike dobivene translacijom  $O$  za proizvoljni vektor  $t$  i translacijom proizvoljnim faktorom  $s$ . Ovaj potprostor može se predstaviti i jednadžbom analognoj (3.9)

$$d' = Qu. \quad (4.8)$$

U ovom poglavlju, matrica  $Q$  predstavlja ortonormiranu bazu, što je mala razlika u odnosu na (3.9). Matrica  $Q$  dobivena je iz matrice  $[A|d]$  na sljedeći način. Od prvog stupca matrice  $[A|d]$  napravi se jedinični vektor. Svaki sljedeći,  $k$ -ti vektor-stupac matrice  $[A|d]$  projicira se na sve prethodne vektor-stupce skalarnim produktom. Duljinom dobivenog vektora pomnoži se svaki prethodni vektor te se rezultat oduzme od  $k$ -tog vektora. Ono što ostane predstavlja komponentu  $k$ -tog vektora okomitu na sve prethodne vektor-stupce. Tako nastali stupci čine stupce matrice  $Q$ . Prva tri elementa vektora  $u$  predstavljaju poziciju instance klase u prostoru, a četvrti njezinu veličinu. Sve instance opisane jednadžbom (4.8) za neku konstantnu vrijednost četvrtog elementa vektora  $u$  predstavljaju isti oblik na svim mogućim pozicijama u 3D prostoru. Nadalje, dvije instance opisane jednadžbom (4.8) za dvije različite vrijednosti četvrtog elementa vektora  $u$  predstavljaju instance istog oblika, različitih veličina. Jednadžba (4.8) može se zapisati na sljedeći način

$$d' = Q_{1:3}u_{1:3} + Q_4u_4, \quad (4.9)$$

gdje je  $Q_{1:3}$  matrica koja predstavlja prva tri stupca matrice  $Q$ , a  $Q_4$  je njezin zadnji stupac. Znači, bilo koji VN deskriptor se može rastaviti na translacijsku komponentu  $Q_{1:3}u_{1:3}$  i komponentu oblika  $Q_4u_4$ .

Sličnost dva oblika  $O$  i  $O'$  mjeri se euklidskom udaljenošću između  $d$  i potprostora određenog jednadžbom (4.8). Ta udaljenost, određena s

$$e_{ts}(O, O') = \min_u \|d - Qu\|_2^2 \quad (4.10)$$

predstavlja mjeru sličnosti između dva oblika, invarijantnu na translaciju i skalu, odnosno

na poziciju i veličinu.

Neka je  $M$  ortonormirana matrica dimenzija  $n \times n$ , čija su prva tri stupca jednaka matrici  $Q_{1:3}$ . Svaki deskriptor  $d$  iz prostora oblika možemo prikazati pomoću matrice  $M$  i vektora  $q$  na sljedeći način

$$d = Mq. \quad (4.11)$$

Svakom vektoru  $q$  odgovara jedan vektor  $d$ , pri čemu prva tri elementa vektora  $q$  opisuju poziciju, a ostali oblik, tj. deskriptori dobiveni jednadžbom (4.11) za vektore  $q$  koji imaju različita prva tri elementa, dok su im ostali elementi isti, predstavljaju isti oblik na različitim pozicijama u 3D prostoru.

### 4.3.3 Klasa oblika kao konveksni podskup prostora oblika

VN prostor oblika sadrži sve instance neke klase oblika. Međutim, osim toga može sadržavati oblike koji ne pripadaju toj klasi. Stoga se uvodi dodatno ograničenje u definiciju modela klase oblika. Klasa oblika definira se kao konveksni podskup prostora oblika koji predstavlja konveksnu ljusku deskriptora uzoraka za učenje. Prvo se računaju VN deskriptori skupa uzoraka za učenje, tj. 3D modela koji predstavljaju instance zadane klase. Metoda za određivanje VN deskriptora željenog 3D modela, predložena je u potpoglavlju 4.7. Na početku stvaranja modela klase oblika, svi deskriptori su centrirani tako da im je oduzeta translacijska komponenta na sljedeći način

$$\tilde{d} = (I - Q_{1:3}Q_{1:3}^\top)d. \quad (4.12)$$

Nadalje, svi su deskriptori normalizirani na duljinu jediničnog vektora

$$\bar{d} = \frac{\tilde{d}}{\|\tilde{d}\|} \quad (4.13)$$

Nakon toga, računa se centroid  $d_c$  svih centriranih i normaliziranih deskriptora  $\bar{d}$ . Oblik opisan s  $d_c$  u ovom se radu naziva *centralni oblik* klase oblika. Oduzimanjem komponente u smjeru  $d_c$  od svih deskriptora  $\bar{d}$ , dobiveni su vektori  $\hat{d}$ ,

$$\hat{d} = \bar{d} - d_c, \quad (4.14)$$

koji predstavljaju varijacije oblika unutar razmatrane klase u odnosu na centralni oblik.

Klase oblika koje predstavljaju stvarne oblike, poput kutija, valjaka, šalica, auta, voća itd., obično imaju značajno manji broj stupnjeva slobode od broja dimenzija njihovog prostora oblika. Na primjer, u ovom je istraživanju definiran VN za opis konveksnih oblika koristeći 66 vektora  $a_i$  dobivenih uniformnim uzorkovanjem jedinične sfere, slično kao što je u poglavlju 3 definiran konveksni predložak. U oba slučaja, prostor oblika ima  $n = 66$  dimenzija. S druge strane, varijacije unutar klase svih valjaka mogu se opisati samo jednim stupnjem slobode - omjerom između radijusa i visine valjka. Kako bi se odredio niže-dimenzionalni potprostor u kojem, unutar zadane tolerancije, leže svi oblici razmatrane klase, primjenjuje se postupak analize glavnih komponentata (engl. *principal component analysis*) (PCA). Na ovaj se način smanjuje računaska složenost, ali i osigurava da se prilikom klasifikacije objekata na dubinskim slikama snimljenima iz jednog pogleda na temelju vidljivih dijelova objekta, rekonstruira nevidljivi. Ovaj se potprostor u ovom radu naziva *latentni prostor*. Latentni je prostor kreiran primjenom PCA na vektore  $\hat{d}$ , uzimajući  $m'$  glavnih komponenti s varijancom većom od predefiniranog praga i formirajući matricu  $M$  koja predstavlja ortonormiranu bazu. Prva tri stupca matrice  $M$  predstavljaju matricu  $Q_{1:3}$ , četvrti stupac predstavlja  $d_c$ , dok ostalih  $m'$  stupaca predstavljaju osi izabranih glavnih komponentata. Dobiveni latentni prostor opisan je jednadžbom

$$d = Mq \quad (4.15)$$

pri čemu je  $q \in \mathbb{R}^m$ ,  $m = m + 4$ . Vektor  $q$  nazivamo latentni vektor. Svi vektori  $\hat{d}$  projicirani su na latentni prostor pomoću izraza

$$q = M^\top \hat{d}, \quad (4.16)$$

te se računa konveksna ljuska  $C$  dobivenih latentnih vektora. Konačno, cjelokupni model klase oblika definiran je VN strukturom  $(A, o, W)$ , matricom  $M$  i podskupom  $C \subset \mathbb{R}^m$ . Ovaj se model koristi u određivanju najbližije instance određene klase za neki zadani oblik, kao što je opisano u potpoglavlju 4.5.

## 4.4 Mjere sličnosti oblika

Predloženi pristup koristi tri mjere sličnosti između dva oblika: (i) sličnost konveksnih ljuski, (ii) Chamferovu udaljenost i (iii) utemeljenost hipoteze.

Za zadanu  $n_H \times 3$  matricu  $A_H$ , skup točaka  $O$  može se predstaviti CTI deskriptorom.

Sličnost dva oblika  $O$  i  $O'$  mjeri se sličnošću njihovih CTI deskriptora primjenom mjere (3.14). S obzirom da se udaljenost CTI deskriptora može se vrlo efikasno izračunati, ova se mjera sličnosti koristi za pročišćavanje velikog skupa početnih hipoteza, kao što je opisano u potpoglavlju 4.5.

U ovom je istraživanju upotrijebljena verzija Chamferove udaljenosti dana jednadžbom (3.1). Ova je mjera korištena pri usporedbi konačnog skupa 3D točaka  $O$  s instancom klase oblika predstavljene VN-om. U tom slučaju,  $O'$  je skup svih točaka na površini razmatrane instance oblika, koji je definiran s

$$O' = \{p \in \mathbb{R}^3 | f(p; d') = 0\}, \quad (4.17)$$

gdje je  $f$  funkcija definirana u potpoglavlju 4.3.1, a  $d'$  je VN deskriptor te instance oblika. Uz pretpostavku da je točka  $p$  na površini oblika  $O'$  ili izvan tog oblika, vrijednost  $f(p; d')$  može se upotrijebiti kao mjera udaljenosti točke  $p \in O$  i njezina najbližeg susjeda u  $O'$ . Treba napomenuti da  $f(p; d')$  nije nužno euklidska udaljenost točke  $p$  od površine  $O'$ , ali može poslužiti kao dobra mjera udaljenosti koja se može efikasno izračunati. Zbog toga se u algoritmu predloženom u ovom poglavlju, umjesto Chamferove udaljenosti (3.1) koristi njezina aproksimacija. Ovako definirana udaljenost prikladna je za minimizaciju silaznim gradijentom.

Treća mjera sličnosti koja se koristi u ovom poglavlju jest utemeljenost hipoteze. Neka su  $O$  i  $O'$  dva skupa 3D točaka koja predstavljaju dva oblika. Sličnost ta dva oblika može se mjeriti brojem točaka iz skupa  $O$  koji imaju dovoljno bliskog susjeda u skupu  $O'$  i obratno, pri čemu je doprinos svake točke veći, što je manja udaljenost do njezina najbližeg susjeda u drugom skupu. Ovaj se kriterij sličnosti može izraziti funkcijom

$$\Omega(O, O') = \sum_{p \in O} \max_{p' \in O'} \omega(p - p') + \frac{|O|}{|O'|} \sum_{p' \in O'} \max_{p \in O} \omega(p - p'), \quad (4.18)$$

pri čemu je funkcija  $\omega$  upotrijebljena za izračun doprinosa svake točke ukupnoj vrijednosti  $\Omega$ . U tu se svrhu može primijeniti

$$\omega(x) = \max\left(1 - \frac{\|x\|_2}{\rho_e}, 0\right) \quad (4.19)$$

pri čemu je  $\rho_e$  korisnički definiran parametar. Slična mjera sličnosti koristi se i u (Aldoma et al., 2016). Vrijednost podudaranja sa scenom je najtočnija od mjera sličnosti navedenih

u ovom potpoglavlju, ali zahtijeva najviše računskog vremena te se zato koristi u konačnoj evaluaciji hipoteza, nakon što je početni skup hipoteza već pročišćen.

Osim spomenutih mjera sličnosti, pri evaluaciji hipoteza također se koristi i penalizacija. Penalizira se broj odudarajućih vrijednosti (engl. *outlier*),  $\Phi(O, O')$ , tj. broj točaka iz skupa  $O$  koje nemaju susjeda u  $O'$  bližeg od  $\rho_e$  i obratno.

## 4.5 Prilagodba VN modela oblaku točaka

Razmatra se klasa oblika definirana s VN-om, matricom  $M$  i koveksnom ljuskom  $C$ , pri čemu je VN definiran matricama  $A$ ,  $o$  i  $W$ , kao što je opisano u potpoglavlju 4.3.1, dok su  $M$  i  $C$  opisane u potpoglavlju 4.3.3. Nadalje, pretpostavka je da je razmatrani objekt na sceni predstavljen 3D oblakom točaka  $O$  dobiven 3D senzorom. Metoda opisana u ovom potpoglavlju određuje instancu klase oblika koja se podudara s oblakom točaka  $O$  te translaciju, rotaciju i skalu koje poravnavaju ovu instancu s  $O$ . Ova metoda je predložena u (Cupec i Đurović, 2018), a sastoji se od tri koraka:

1. Generiraju se početne hipoteze za različite orijentacije.
2. Odabire se  $n_{hyp}$  najboljih početnih hipoteza. Za svaku se određuju orijentacija, latentni vektor koji minimizira funkciju troška te se računa vrijednost podudaranja sa scenom.
3. Najbolja hipoteza odabire se kao konačno rješenje.

Početne su hipoteze generirane uzorkovanjem 3D rotacijske grupe  $SO(3)$  i poravnanjem CTI-a centralnog oblika klase s  $O$ . Uniformno uzorkovanje  $SO(3)$  izvodi se definiranjem skupa koordinatnih sustava koji predstavljaju kandidate za orijentacije instanci klasa oblika u odnosu na koordinatni sustav kamere. Osi  $z$  ovih koordinatnih sustava generiraju se jednolikim uzorkovanjem jedinične sfere. Na taj način dobiveno je 66 osi. Za svaku od ovih  $z$ -osi, generirano je 16 osi  $x$  s jednoliko raspodijeljenim orijentacijama. Kao rezultat, ukupno je definirano 1056 kandidata koordinatnih sustava. Ishodišta ovih koordinatnih sustava podudaraju se s ishodištem koordinatnih sustava kamere. Sve točke  $p \in O$  transformirane su u svaki potencijalni koordinatni sustav te je za svaki postavljena hipoteza  $h^{(1)}$ . Hipoteza  $h^{(1)} = (R^{(1)}, q^{(1)})$  se sastoji od rotacijske matrice  $R^{(1)}$ , koja opisuje orijentaciju instance klase



oblika u odnosu na koordinatni sustav kamere, i latentnog vektora

$$q^{(1)} = \begin{bmatrix} M^\top d_c + \begin{bmatrix} u \\ 0 \end{bmatrix} \end{bmatrix} \quad (4.20)$$

koji minimizira udaljenost konveksnih ljusti

$$e_{ts} = \|d^v - d_c^v - Q^v u\|_2^2 \quad (4.21)$$

između  $O$  i centralnog oblika definiranog deskriptorom  $d_c$  razmatrane klase. Vektor  $u$  za koji je gornji izraz minimalan računa se na sljedeći način

$$u = (Q^{v\top} Q^v) Q^{v\top} (d^v - d_c^v). \quad (4.22)$$

Ove početne hipoteze su rangirane prema udaljenosti konveksnih ljustaka te je  $n_{hyp}$  hipoteza s najmanjom udaljenošću prosljeđeno u idući korak. U pokusima predstavljanim u potpoglavlju 4.8,  $n_{hyp} = 50$ .

U sljedećem se koraku, za svaku hipotezu generiranu u prvom koraku, izračunavaju optimalna rotacijska matrica  $R^{(2)}$  i latentni vektor  $q^{(2)}$  minimiziranjem funkcije troška

$$\mathfrak{S}(R, q) = E(O, V(R, q)) + \alpha \Upsilon(O, V(R, q)), \quad (4.23)$$

gdje  $\alpha$  predstavlja predefinirani težinski faktor, a  $V(R, q)$  označava skup točaka dobivenih rotacijom skupa točaka  $O'$ , definiranog s (4.17) za  $d' = Mq$ , koristeći rotacijsku matricu  $R$ . Minimizacija (4.23) provodi se Levenberg-Marquardt algoritmom ograničenim na  $C$ , gdje su  $R^{(1)}$  i  $q^{(1)}$  upotrijebljeni pri inicijalizaciji. Nakon optimizacije, za dobiveni  $R^{(2)}$  i  $q^{(2)}$ , izračunava se vrijednost podudaranja sa scenom definirana s

$$\Psi(R, q) = \Omega(O, V(R, q)) - \beta \cdot \Phi(O, V(R, q)) - \gamma \cdot \Delta_{SP}(V(R, q)), \quad (4.24)$$

pri čemu su  $\Omega$  i  $\Phi$  opisane u potpoglavlju 4.4,  $\beta = 0.5$  i  $\gamma = 100$  su težinski faktori, a  $\Delta_{SP}(X)$  predstavlja udaljenost najniže točke u skupu točaka  $X$  od potporne ravnine na koju je smješten objekt. Detekcija potporne ravnine čest je korak u detekciji i prepoznavanju objekata, npr. (Aldoma et al., 2016), te za to postoji više metoda.

Kako bi se izračunali  $\Omega$  i  $\Phi$ , za svaku točku  $p \in O$  potrebno je identificirati njezinog najbližeg susjeda u  $V(R, q)$  i obratno. Kao mjera udaljenosti između točke  $p \in O$  i njoj najbliže točke u  $V(R, q)$  koristi se vrijednost ISF u točki  $p$ . Kako bi se izračunao drugi

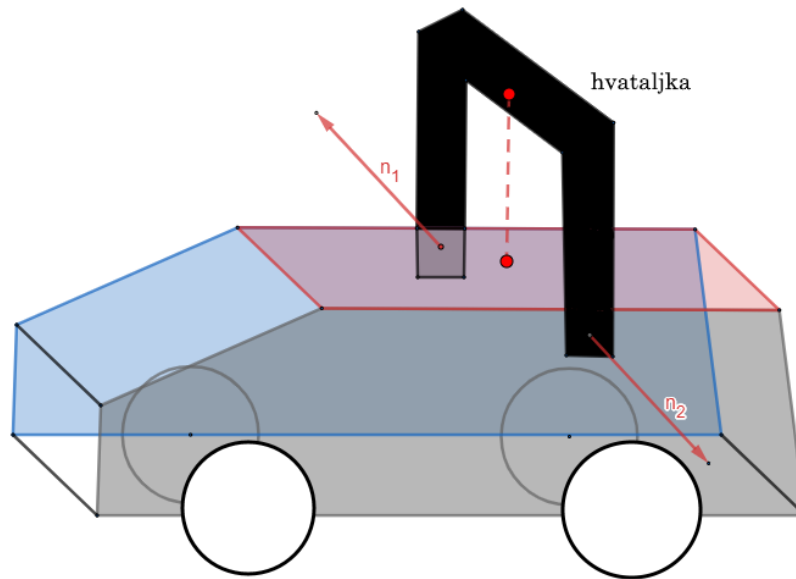
dio jednadžbe (4.18), VN model projiciran je na sliku te je vidljivi dio površine  $V(R, q)$  uzorkovan. Za zadani skup uzorkovanih točaka slike, računaju se najbliže i najudaljenije 3D točke VN modela  $V(R, q)$  uzduž optičkih zraka koje odgovaraju zadanim točkama slike. VN model je prethodno skaliran, zarotiran i translaticiran. Rezultat drugog koraka je skup  $n_{hyp}$  hipoteza  $h^{(2)} = (R^{(2)}, q^{(2)})$ .

Treći korak predstavlja izbor hipoteze s najvećom vrijednosti podudaranja sa scenom kao konačnog rješenja. Rezultat procedure opisane u ovom potpoglavlju je instanca razmatrane klase oblika koja odgovara lokalnom minimumu funkcije podudaranja sa scenom (4.24), parametrizirana svojom orijentacijom, pozicijom, skalom i vektorom oblika. Orijetacija izabrane instance oblika definirana je optimalnom rotacijskom matricom  $R^{(2)}$ , njezina je pozicija definirana s prva tri elementa  $q^{(2)}$ , dok preostali elementi  $q^{(2)}$  predstavljaju vektor oblika koji opisuje razliku između razmatranog i centralnog oblika klase. Oblik rezultatne instance klase potpuno je opisan funkcijom  $f(p; Mq^{(2)})$ .

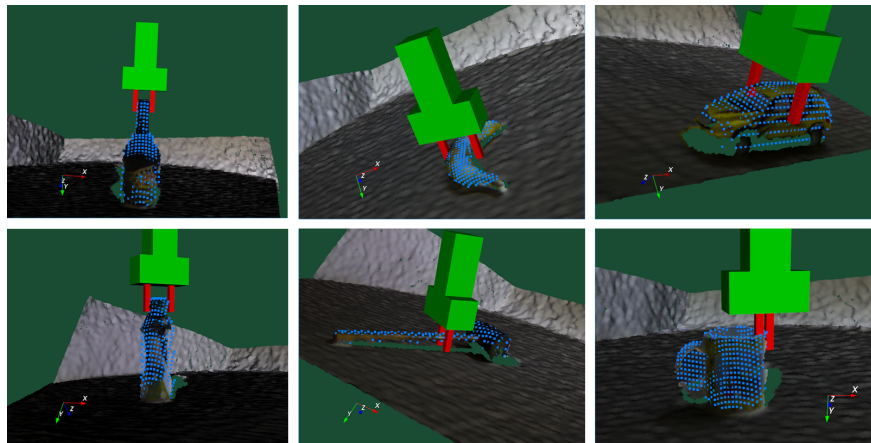
## 4.6 Planiranje hvatanja na temelju VN modela

Važnu prednost predložene metode u odnosu na većinu metoda klasifikacije predstavlja informacija o položaju instance oblika i njezini parametri oblika, koji mogu biti od koristi u robotskoj manipulaciji. Ovo je prikazano primjerima na slici 4.4, gdje je plavim točkama prikazan model, definiran s  $(R, q)$ , projiciran na sliku snimljenu kamerom. Stupci matrice  $RA^T$  predstavljaju normale projiciranog modela u koordinatnom sustavu kamere. Prikladno hvatanje za svaku klasu definira se specificiranjem dviju stranica VN modela koje se koriste kao površine za hvatanje. Primjer je prikazan na slici 4.3, gdje su ove stranice označene svijetlo plavom i sivom bojom.

Orijetacija hvataljke definirana je normalama tih dviju stranica, prikazanima na slici 4.3 crvenim vektorima i trećom izabranom stranicom, označenom svijetlo crvenom bojom na slici 4.3. Udaljenost između hvataljke i objekta određena je zadanom sigurnosnom udaljenošću između hvataljke i treće stranice, prikazanom isprekidanom crvenom linijom na slici 4.3. Razmak između prstiju hvataljke definiran je udaljenošću dviju specificiranih stranica. S obzirom da je svaka stranica VN modela definirana normalom  $a_i$  i udaljenošću  $d_i$ , koja predstavlja  $i$ -ti element VN deskriptora, parametri definiranog hvatanja ovise o estimiranoj orijentaciji oblika kao i o elementima VN deskriptora. Prilagodbom instance klase oblika razmatranom objektu na sceni postupkom opisanim u poglavlju 4.5, parametri definiranog



Slika 4.3: Planiranje hvatanja na temelju VN modela



Slika 4.4: Hvatanje definirano na temelju odabranih stranica VN modela prilagođeno objektima na sceni. Plave točkice predstavljaju instancu VN-a projiciranu na RGB-D sliku.

hvatanja se automatski prilagođavaju obliku toga objekta.

## 4.7 Određivanje VN strukture i deskriptora na osnovi skupa za učenje

U potpoglavlju 4.3, predložen je način definiranja modela klase oblika kao konveksne ljuške latentnih vektora uzoraka za učenje. Ovaj postupak zahtijeva da je svaki uzorak za učenje predstavljen vlastitim VN deskriptorom. U ovom je potpoglavlju opisana metoda koja određuje VN strukturu na osnovi skupa za učenje koji se sastoji od 3D modela objekata pripadnika razmatrane klase oblika predstavljenih mrežom trokuta te VN deskriptor

danog 3D modela.

### 4.7.1 Poravnanje uzoraka iz skupa za učenje s referentnim modelom

Predloženi postupak zahtijeva konzistentnu orijentaciju uzoraka za učenje. Zato je odabran jedan model iz skupa za učenje kao *referentni model* razmatrane klase i svi ostali modeli su poravnani s tim modelom. Ovakvo poravnanje podrazumijeva pronalazak optimalne rotacije, translacije i skale s obzirom na određenu mjeru sličnosti oblika. U ovom su radu prijedlozi orijentacija generirani sparivanjem parova dominantnih tangencijalnih ravnina dvaju poravnatih objekata, računanjem optimalne translacije i skale za svaki prijedlog orijentacije minimiziranjem izraza (4.10) i odabiranjem prijedloga za koji je (4.10) minimalan. Ovaj je postupak detaljno opisan u poglavlju 3. Drugi je mogući način da se poravnanje uzoraka za učenje pronađe odgovarajućom varijantom ICP algoritma, koji može minimizirati primjerice Chamferovu udaljenost između modela iz skupa za učenje i referentnog modela. U pokusima prikazanim u ovom radu koristi se prvi način.

### 4.7.2 Optimalna VolumeNet struktura

Struktura VN-a određena je matricama  $A$ ,  $o$  i  $W$ , kao što je opisano u potpoglavlju 4.3.1. Optimalne dimenzije i elementi ovih matrica mogli bi se odrediti optimizacijskom procedurom koja minimizira ukupnu udaljenost između slučajno odabranih točaka u okolini površine uzoraka za učenje i vrijednosti funkcije  $f$  u tim točkama, određene matricama  $A$ ,  $o$  i  $W$ . Pretpostavimo da se skup za učenje sastoji od  $n_{TS}$  poravnanih 3D modela objekata pripadnika iste klase oblika te da je  $P_i, i = 1, \dots, n_{TS}$  skup nasumično odabranih točaka na površini tih 3D modela ili točaka izvan tih modela, ali u blizini njihove površine. Nadalje, neka je svakoj točki  $p_{ij} \in P_i$  dodijeljena vrijednost  $y_{ij}$  koja predstavlja udaljenost do najbliže točke na površini  $i$ -tog modela. Dok je struktura  $(A, o, W)$  zajednička svim instancama određene klase oblika predstavljene VN-om, različite instance te klase određene su drugačijim vektorima  $d$ . Za danu VN strukturu  $(A, o, W)$ , optimalni vektor  $d$ , koji daje najbolju aproksimaciju  $i$ -tog uzorka za učenje, može se izračunati traženjem vektora  $d$  koji minimizira sljedeću funkciju troška

$$\mathfrak{S}_{d,i}(d; A, o, W) = \sum_{j=1}^{|P_i|} \min(|y_{ij} - f(p_{ij}; d)|, \delta_i), \quad (4.25)$$

gdje je  $\delta_i$  prag izračunat korisnički definiranim postotkom veličine graničnog okvira  $i$ -tog modela. Primjenom ove mjere sličnosti, optimalna VN struktura  $(A, o, W)$  cijelog skupa za učenje mogla bi se dobiti minimiziranjem ukupnog troška

$$\mathfrak{S}_{AoW}(A, o, W) = \sum_{i=1}^{n_{TS}} \min_d \mathfrak{S}_{d,i}(d; A, o, W). \quad (4.26)$$

Međutim, minimizacija izraza (4.26) vrlo je kompleksan optimizacijski problem, za koji je u ovom radu razvijena poluautomatska heuristička procedura koja se sastoji od sljedećih koraka:

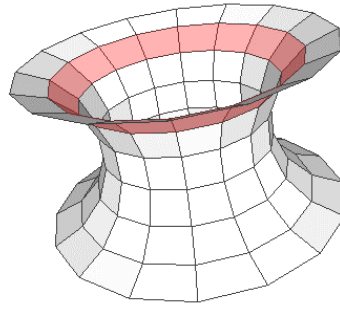
1. Korisnik definira metamodel razmatrane klase oblika, koji se sastoji od jednog ili više konveksnih, konkavnih ili toroidalnih komponenti. Ovaj metamodel definiran je vektorom  $o$  i matricom  $W$ .
2. Na temelju metamodela formira se početna matrica  $A$ .
3. Svaki se 3D model iz skupa za učenje segmentira na konveksne, konkavne ili toroidalne segmente.
4. Segmenti dobiveni u 3. koraku pridružuju se komponentama metamodela.
5. Odbacuju se irelevantne stranice komponenti metamodela, koje odgovaraju pojedinim retcima početne matrice  $A$ . Kao rezultat dobiva se konačna matrica  $A$ .

Ovi su koraci opisani u nastavku.

### 4.7.3 Metamodel

Razmatraju se oblici koji se mogu aproksimirati kombinacijom konveksnih i konkavnih skupova točaka. Na primjer, oblik čekića može se aproksimirati unijom dva konveksna skupa  $V_{ch} = V_{c,1} \cup V_{c,2}$ , pri čemu  $V_{c,1}$  predstavlja glavu, a  $V_{c,2}$  dršku. Drugi je primjer zdjela koja se može predstaviti presjekom jednog konveksnog skupa  $V_c$  koji predstavlja vanjsku stijenku zdjele i komplementa konveksnog skupa  $V_{\bar{c}}$  koji predstavlja njezinu unutrašnjost, tj.  $V_z = V_c \cap V_{\bar{c}}$ . Radi pojednostavljenja, izraz *konkavni* koristit će se kao sinonim za komplement konveksnog skupa.

Specifikacija osnovnih tipova oblika koji čine model klase oblika zajedno s operacijama između tih osnovnih oblika (unija ili presjek) u ovom se radu naziva *metamodel* klase oblika. Osnovni oblici koji čine metamodel u ovom se radu nazivaju *komponentama* metamodela.



Slika 4.5: Toroidalna površina definirana presjekom 7 komplementa 3D konusa. Jedan od tih konusa označen je crvenom bojom.

Osim konveksnih i konkavnih komponenti, metamodel također može sadržavati toroidalne komponente, koje predstavljaju presjek nekoliko komplementa 3D konusa, kao što je prikazano na slici 4.5. U trenutnoj implementaciji predložene metode, metamodele definira korisnik. Ovo je jednostavan korak koji zahtijeva samo nekoliko unosa. Primjerice, kako bi se definirao metamodel koji predstavlja klasu čekić, samo su tri unosa potrebna: tip prve komponente (koveksna), operacija (unija) i tip druge komponente (koveksna). Vektor  $o$  i matrica  $W$  potpuno su opisani definiranjem metamodela.

Matrica  $A$  formira se ulančavanjem matrica  $A_i$  definiranih za svaku  $i$ -tu komponentu metamodela. U slučaju koveksnih ili konkavnih komponenti, matrica  $A_i$  je definirana uniformnim uzorkovanjem jedinične sfere, dok je u slučaju toroidalnih komponenti,  $A_i$  sačinjena od normala stranica prikazanih na slici 4.5. Normale stranica toroidalnih komponenti definirane su kutovima azimuta,  $\alpha$  i elevacije,  $\beta$ , u odnosu na ravninu okomitu na centralnu os toroidalnog oblika, gdje su  $\alpha \in [0, 2\pi)$ , jednoliko uzorkovan rezolucijom od  $22.5^\circ$ ,  $\beta = k \cdot 22.5^\circ$ ,  $k \in [-3, 3]$ ,  $k \in \mathbb{Z}$ . Na ovaj je način definirano ukupno 112 površina, gdje normala svake površine odgovara jednom retku matrice  $A_i$ .

#### 4.7.4 Segmentacija 3D modela na konveksne, konkavne i toroidalne površine i njihovo pridruživanje

Trodimenzionalni modeli koji predstavljaju uzorke za učenje segmentirani su na konveksne, konkavne i toroidalne površine, ovisno o tome koja je vrsta površine uključena u metamodel. U tu svrhu, razvijena su dva algoritma, jedan koji segmentira 3D mrežu trokuta u konveksne i konkavne površine i drugi koji detektira toroidalne površine (Cupec; Cupec et al., 2020). Segmentacija se provodi segmentacijom mreže modela u ravninske dijelove primjenom metode predložene u (Cupec et al., 2017) i grupiranjem tih dijelova na temelju kriterija

konveksnosti.

Segmenti dobiveni segmentacijom pridruženi su komponentama metamodela optimizacijskim procesom. Kako bi se smanjila računaska kompleksnost ovog procesa, samo je predefinirani broj najvećih segmenata razmatran. Svaki je segment svakog uzroka za učenje pridružena jednoj komponenti metamodela istoga tipa (koveksni, konkavi ili toroidalni) ili je odbačena kao odudarajuća vrijednost. Općenito, svaki segment može biti pridružen bilo kojoj od nekoliko komponenti metamodela istoga tipa. Zato je moguć velik broj pridruživanja. Optimalno pridruživanje određeno je iscrpnim pretraživanjem, koje računa trošak (4.25) svakog mogućeg pridruživanja i odabire pridruživanje s najmanjim troškom. Vektori  $d$  korišteni za izračun (4.25), predstavljaju deskriptore segmenata. U slučaju konveksnih segmenata to su njihovi CTI deskriptori. Deskriptori konkavnih segmenata računaju se analogno, tj. na način objašnjen u poglavlju 3.3.2, samo što se umjesto maksimuma redaka matrice  $X$  u jednadžbi (3.4) računaju minimumi. Ako je više segmenata scene pridruženo istoj komponenti metamodela, tada se deskriptor računa za uniju tih segmenata. Toroidalne površine predstavljaju presjek više komplementa 3D konusa, kao na slici 4.5. Deskriptor  $d$  toroidalne površine sastoji se od 112 elemenata, pri čemu svaki element predstavlja udaljenost jedne površine od ishodišta. Računa se analogno deskriptoru  $d$  konkavne površine.

Traženje optimalnog pridruživanja segmenata 3D modela iz skupa za učenje komponentama metamodela iscrpnim pretraživanjem implementirano je na sljedeći način. Svakom se paru  $(i, j)$ , pri čemu je  $i$  indeks komponente metamodela, a  $j$  indeks segmenta modela skupa za učenje, pridružuje odgovarajući deskriptor. Iscrpno pridruživanje, primijenjeno u opisanoj implementaciji, provodi se na sljedeći način. Stvara se stablo čiji su čvorovi  $c_{ij}$  korespondencije segment-komponenta, pri čemu je  $i$  indeks segmenta modela iz skupa za učenje, osim korijena stabla  $c_{00}$ , koji ne predstavlja korespondenciju. Svakom čvoru  $c_{ij}$  dodijeljen je indeks neke komponente metamodela ili indeks 0, koji označava da  $i$ -ti segment ne predstavlja niti jednu komponentu metamodela. Stablo se grana od nekog čvora  $c_{ij}$  tako da se stvaraju novi čvorovi  $c_{i+1,k}$  za sve komponente istog tipa kao segment  $i + 1$  te čvor koji označava da segment  $i + 1$  ne predstavlja niti jednu komponentu metamodela. Svaka grana stabla od lista do korijena predstavlja jednu moguću kombinaciju korespondencija između svih segmenata modela iz skupa za učenje i komponenti metamodela, pri čemu je moguće da je više segmenata pridruženo istoj komponenti. Za svaku se granu stabla, odnosno za svaku

kombinaciju korespondencija, računa trošak pomoću

$$E = \sum_{p \in P_S(M)} \min\{|f(p; d) - \delta(p, M)|, \rho\}, \quad (4.27)$$

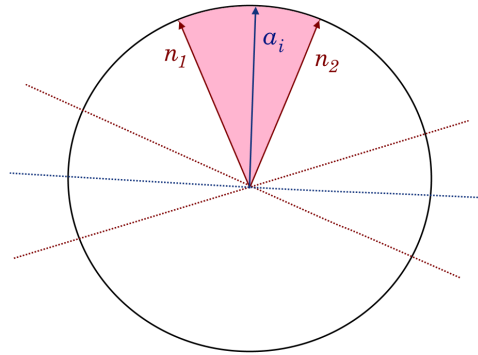
gdje je  $f$  ISF VN modela definirana u potpoglavlju 4.3.1,  $\rho$  je konstanta, a  $d$  je deskriptor izračunat za razmatranu granu stabla. Deskriptor  $d$  se sastoji od deskriptora komponenti metamodela. Ako je neka komponenta metamodela pridružena samo jednom segmentu modela iz skupa za učenje, onda je deskriptor te komponente jednak deskriptoru tog segmenta. Ako je neka komponenta metamodela pridružena više segmenata, onda je njezin deskriptor jednak deskriptoru unije tih segmenata. Ako neka komponenta nije pridružena niti jednom segmentu, onda se funkcija  $f$  izračunava kao da te komponente nema u VN modelu. Slika 4.1 predstavlja primjer VN modela od dvije komponente, predstavljene čvorovima  $f_8$  i  $f_9$ . Ako npr. komponenta predstavljena čvorom  $f_8$  nije pridružena niti jednom segmentu, onda se funkcija  $f$  izračunava kao da čvor  $f_8$  (i sve ispod njega) ne postoji, tj. kao da je  $w_{10,8} = 0$ .

Rezultat je deskriptor čiji je broj elemenata  $n = n_k \cdot n_{CTI} + n_t \cdot n_{dt}$ . Pri tome,  $n_k$  predstavlja ukupan broj konkavnih i konveksnih komponenti metamodela,  $n_t$  predstavlja broj toroidalnih komponenti metamodela,  $n_{CTI}$  je broj elemenata CTI deskriptora, a  $n_{dt}$  broj elemenata deskriptora toroidalnih površina.

#### 4.7.5 Odbacivanje irelevantnih stranica

Komponente metamodela su osmišljene tako da dozvoljavaju modeliranje proizvoljnih konveksnih, konkavnih i toroidalnih oblika. Stoga svaka matrica  $A_i$  određene komponente metamodela sadrži normale stranica orijentirane u svim smjerovima. Međutim, neke od tih normala nisu nužne za modeliranje određene klase oblika. Ovo je objašnjeno na temelju dva jednostavna 2D primjera prikazana na slici 4.7. Iako se metamodel koji predstavlja boce sastoji od dvije konveksne komponente, od kojih svaka ima 8 stranica, samo je 7 stranica dovoljno za modeliranje donjeg dijela boce, a samo 5 je dovoljno za modeliranje gornjeg dijela. Stranica komponente metamodela definirana normalom  $a_i$  i udaljenošću  $d_i$  smatra se relevantnom ako je ravnina definirana s  $a_i$  i  $d_i$  tangencijalna ravnina za većinu uzoraka za učenje koji pripadaju određenoj klasi. U primjerima na slici 4.7, stranice označene crnim vektorima  $a_i$  leže na tangencijalnim ravninama, dok ravnine označene crvenim isprekidanim linijama i crvenim vektorima  $a_i$  nisu tangencijalne jer sijeku površinu objekta u njihovim točkama dodira. Relevantne stranice komponenti metamodela određuju se na temelju 3D



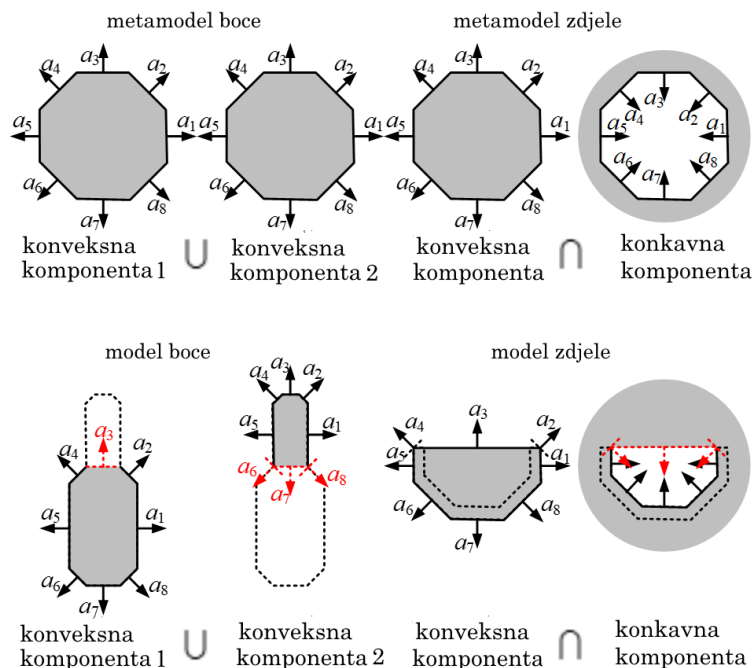


Slika 4.6: Dvodimenzionalni prikaz određivanja relevantnih stranica. Normale  $n_1$  i  $n_2$  stranica označenih isprekidanim tamno crvenim linijama definiraju kružni isječak. Stranica čija se normala  $a_i$  nalazi unutar kružnog isječka smatra se relevantnom.

modela iz skupa za učenje. Za svaki od tih 3D modela ispituje se koje stranice komponenti metamodela sijeku površinu 3D modela, a koje ne. Ako neka stranica neke komponente metamodela siječe površinu većine 3D modela iz skupa za učenje, ta se stranica smatra irelevantnom. Siječe li neka stranica komponente metamodela površinu danog 3D modela ispituje se na sljedeći način. S obzirom na način na koji se izračunavaju udaljenosti  $d_i$ , koje definiraju stranice komponenti metamodela, dodirne točke tih stranica s površinom 3D modela su vrhovi u kojima se sastaju tri ravninska segmenta na koje je segmentirana površina 3D modela, kako je objašnjeno u poglavlju 4.7.4, normale triju ravninskih segmenata koji se sijeku u nekom vrhu definiraju isječak jedinične kugle. Ukoliko se normala stranice komponente metamodela nalazi unutar isječka definiranog normalama ravninskih segmenata koje se sastaju u dodirnoj točki te stranice, ta se stranica smatra relevantnom. U suprotnom, stranica se smatra irelevantnom. Dvodimenzionalni primjer određivanja relevantne stranice prikazan je slikom 4.6. Nakon odbacivanja irelevantnih vektora  $a_i$  iz matrice  $A$  metamodela, struktura VN-a je potpuna.

#### 4.7.6 Računanje deskriptora

Nakon određivanja strukture VN-a pomoću procedure opisane u potpoglavljima 4.7.4 - 4.7.5, za svaki uzorak za učenje može se odrediti VN deskriptor. To se radi određivanjem optimalnog pridruživanja između prethodno detektiranih konveksnih, konkavnih i toroidalnih površina i komponenti modela te računanjem deskriptora za optimalno pridruživanje. Ovaj je korak ponavljanje procedure opisane u potpoglavljju 4.7.4, ali s novom matricom  $A$  dobivenom nakon odbacivanja irelevantnih stranica, kako je opisano u 4.7.5. Nakon što se



Slika 4.7: Metamodeli (gornji redak) i modeli (donji redak) dviju klasa oblika: boce (lijevo) i zdjele (desno). Boce su predstavljene unijom dviju konveksnih komponenti, dok su zdjele predstavljene presjekom jedne konveksne i jedne konkavne komponente. Cjelokupni objekti predstavljeni su isprekidanom crtom, relevantne stranice predstavljene su punom crnom linijom, a irelevantne stranice crvenom isprekidanom linijom.

promijeni matrica  $A$ , optimalno pridruživanje se također može promijeniti u odnosu na ono dobiveno prvim korakom. Dobiveni deskriptori uzoraka za učenje koriste se pri kreiranju modela klase oblika, kako je opisano u potpoglavlju 4.3.

## 4.8 Eksperimentalna evaluacija

U ovom su poglavlju opisani pokusi kojima je ispitana učinkovitost predloženog modela klase oblika. Prvo, VN-i su upotrijebljeni za modeliranje 10 modela klasa oblika na temelju izazovnog 3DNet skupa podataka (Wohlkinger et al., 2012). 3DNet skup podataka detaljno je opisan u poglavlju 3.5 kao i postupak generiranja gustih mreža trokuta. Dobiveni modeli koriste se u klasifikaciji objekata na RGB-D slikama. Svrha ovih pokusa je istražiti je li VN deskriptor dovoljno deskriptivan za razlikovanje razmatranih klasa.

Program klasifikacije objekata razvijen je u C++ programskom jeziku pomoću PCL (Rusu i Cousins, 2011) programske biblioteke i biblioteke OpenCV. Dio računskih operacija tijekom procesa učenja proveden je u Matlabu. Postupak opisan u potpoglavlju 4.7 koristi se za određivanje strukture  $(A, o, W)$  VN modela za svaku od 10 klasa i računanje VN deskriptora  $d$  za 351 uzorak za učenje. Ovi deskriptori koriste se za određivanje latentnog potprostora, definiranog ortonormiranom bazom  $M$  i konveksnom ljuskom  $C$  u ovom

Tablica 4.1: Parametri modela klasa oblika

| Klasa          | Metamodel                       | Veličina deskriptora | Dimenzija latentnog prostora |
|----------------|---------------------------------|----------------------|------------------------------|
| jabuka         | konveksni                       | 66                   | 2                            |
| banana         | konveksni + torusni             | 79                   | 4                            |
| boca           | $2 \times$ konveksni            | 102                  | 6                            |
| zdjela         | konveksni + konkavni            | 104                  | 6                            |
| auto           | konveksni                       | 66                   | 2                            |
| krofna         | konveksni + torusni             | 173                  | 2                            |
| čekić          | $2 \times$ konveksni            | 107                  | 7                            |
| šalica         | $2 \times$ konveksni + konkavni | 149                  | 6                            |
| tetrapak       | konveksni                       | 66                   | 2                            |
| toaletni papir | konveksni + konkavni            | 82                   | 2                            |

potprostoru. Latentni se potprostor određuje za svaku klasu, primjenom metode opisane u potpoglavlju 4.3. Na taj su način kreirani modeli razmatranih klasa oblika. Parametri dobivenih modela klasa oblika prikazani su u tablici 4.1.

Ovi su modeli tada primijenjeni u klasifikaciji ispitnih RGB-D slika iz 3DNet skupa. Segmentacija objekata od interesa od pozadine nije doprinos ovog rada, a provedena je heurističkom metodom zasnovanoj na određivanju potporne ravnine, izbacivanjem te ravnine sa slike te povezivanjem preostalih povezanih skupova točaka detektiranih unutar unaprijed određene udaljenosti od kamere. Kao rezultat, skup točaka  $O$  koji predstavlja razmatrani objekt dobiven je za svaku ispitnu sliku. Za zadani skup točaka  $O$ , izabrana je najsljednija instanca svake klase oblika zajedno s optimalnom pozicijom, orijentacijom i skalom primjenom postupka opisanog u potpoglavlju 4.5. Sličnost svake instance klase s razmatranim objektom mjeri se vrijednošću podudaranja sa scenom (4.24). Konačno, razmatranom se objektu dodjeljuje klasa s najvećom vrijednošću podudaranja sa scenom.

Postignuti rezultati prikazani su u tablici 4.2 te su uspoređeni s rezultatima drugih pet metoda na istom skupu podataka (Wohlkinger et al., 2012). Te su metode bazirane na deskriptirima: Viewpoint Feature Histogram (VFH) (Rusu et al., 2010), Clustered Viewpoint Feature Histogram (CVFH) (Aldoma et al., 2011), the Ensemble of Shape Functions (ESF) (Wohlkinger i Vincze, 2011a; Patten et al., 2016) i SHOT deskriptor (Salti et al., 2014; Wohlkinger et al., 2012). Osim toga, prikazani su i rezultati dobiveni metodom klasifikacije opisanom u poglavlju 3 (CHASF). Može se zaključiti da je klasifikacija objekata postignuta VN modelom usporediva s ostalim metodama te da VN opisuje razmatranu klasu dovoljno

Tablica 4.2: Rezultati klasifikacije objekata [%]

|                | VFH | CVFH       | ESF        | SHOT       | CHASF     | VN         |
|----------------|-----|------------|------------|------------|-----------|------------|
| jabuka         | 64  | 95         | <b>99</b>  | 93         | 91        | 80         |
| banana         | 97  | 99         | 89         | 73         | 95        | <b>100</b> |
| boca           | 84  | 87         | 94         | <b>99</b>  | 93        | <b>99</b>  |
| zdjela         | 97  | 97         | 88         | <b>100</b> | 99        | <b>100</b> |
| auto           | 87  | 91         | <b>98</b>  | 95         | 95        | 72         |
| krofna         | 78  | 96         | 96         | 90         | 86        | <b>100</b> |
| čekić          | 93  | 98         | <b>100</b> | 97         | 97        | 99         |
| šalica         | 98  | <b>100</b> | <b>100</b> | 95         | 94        | 94         |
| tetrapak       | 5   | 5          | 91         | 74         | <b>96</b> | 91         |
| toaletni papir | 20  | 11         | 61         | 58         | 84        | <b>100</b> |
| UKUPNO         | 72  | 78         | 92         | 87         | 93        | <b>94</b>  |

precizno da bi se omogućilo razlikovanje instanci pripadnika različitih klasa. Nadalje, s obzirom da vrijednost podudaranja sa scenom primijenjena kao kriterij klasifikacije zahtijeva točno estimiran položaj objekta, uspješna klasifikacija podrazumijeva da predloženi postupak ispravno poravnava velik postotak razmatranih objekata s modelima klasa oblika.

Klasifikacijski algoritam izvođen je na standardnom PC-u sa Intel Core i7-4790 CPU, 3.60GHz sa 16GB RAM. Prosječno vrijeme obrade jednog razmatranog objekta, što uključuje segmentaciju, podudaranje VN modela 10 klasa i klasifikaciju, bilo je 3 sekunde.

## 4.9 Zaključak

U ovom poglavlju opisan je VolumeNet model klasa 3D oblika te je pokazano kako se taj model može primijeniti u klasifikaciji objekata kao i za estimaciju položaja objekta i njegovih parametara oblika. Opisan je algoritam klasifikacije objekata, koji koristi VN model za modeliranje više klasa oblika te određuje sličnost ispitivanog objekta prikazanog na RGB-D slici s navedenim modelima klasa oblika. Djelotvornost predloženog algoritma pokazana je pokusima klasifikacije objekata gdje su objekti prikazani RGB-D slikama klasificirani podudaranjem VN modela 10 klasa sa skupovima točaka tih objekata te dodjeljivanjem klase objektu na temelju najveće vrijednosti podudaranja sa scenom. Postignuti rezultati klasifikacije usporedivi su s rezultatima postignutima na temelju četiri deskriptora prethodno ispitanim na istom skupu podataka. Nadalje, predložena je primjena VN modela u robotici i adaptivnom planiranju hvatanja.

Najveći je nedostatak trenutne implementacije predložene metode ručno definiranje me-

tamodela. Iako je taj postupak vrlo jednostavan, koristan bi bio automatizirani postupak koji bi pronašao optimalni metamodel pojedine klase oblika. Osim toga, moguće je primijeniti generalni optimizacijski pristup za minimizaciju troška (4.26) umjesto trenutno primijenjene heurističke metode opisane u potpoglavlju 4.7. Konačno, mogući smjer istraživanja u budućnosti je primjena predložene metode za klasifikaciju objekata na scenama s većim brojem objekata.

# 5

## Određivanje korespondencija dijelova unutar klasa objekata zasnovano na poliedarskim modelima klasa oblika

Većina složenih objekata sastoji se od semantički različitih dijelova koji su uglavnom i geometrijski različiti. Točke objekta relevantne za određene robotske akcije uglavnom su određene različitim fizikalnim svojstvima objekta, poput dimenzija, distribucije mase i samom svrhom objekta. Robotska operacija definirana za određeni dio reprezentativnog objekta, može se prilagoditi na druge instance iste klase detekcijom odgovarajućih, korespondentnih komponenti. U ovom je poglavlju predstavljen četvrti doprinos ovog doktorskog rada, metoda određivanja korespondencija dijelova unutar klasa objekata zasnovana na poliedarskim modelima klasa oblika. Ova je metoda opisana u (Đurović et al., 2020). Predložena metoda za semantičko pridruživanje komponenti objekata pripadnika iste klase prikladna je za robotske zadatke i izvođenje u stvarnom vremenu te zahtijeva samo nekolicinu unaprijed označenih reprezentativnih modela. Predloženi se postupak zasniva na grafu pridruživanja komponenti i novom deskriptoru koji opisuje geometrijski razmještaj komponenti. Metoda je eksperimentalno ispitana na zahtjevnom skupu podataka.

## 5.1 Uvod

Jedan od trendova u robotici je učenje robota na temelju radnji koje obavlja ljudski instruktor, poput kinestetičkog treniranja, pri čemu čovjek ručno navodi robota da učini određenu akciju te robot naučeni pokret primjenjuje pri rješavanju drugih sličnih praktičnih zadataka (Abdo et al., 2013; Pastor et al., 2011). Unaprijeđena verzija takvog treninga predstavlja opisivanje određene akcije robota za određenu instancu klase objekata koju se u ovom radu naziva *reprezentativni objekt* te primjena algoritma koji bi prilagodio tu radnju na druge instance iste klase. Kako bi se postigla ta mogućnost, razmatrani algoritam mora povezati komponente reprezentativnog objekta relevantne za određenu radnju s odgovarajućim segmentima drugih instanci iste klase objekata.

Problem koji se rješava u ovom poglavlju je pridruživanje komponenti različitih objekata koje imaju istu svrhu. S obzirom da je ciljna primjena razmatrana u ovom radu u robotici, komponente su definirane kao područja površine objekta koje bi potencijalno mogle predstavljati kontaktne površine između objekta i robotskog alata pri nekoj robotskoj radnji. Na primjer, ako je zadatak donijeti šalicu, robot bi trebao prepoznati i uhvatiti ručku. Ako je zadatak promjena žarulje neke svjetiljke, robot treba prepoznati žarulju koju će uhvatiti i zamijeniti. Uobičajen postupak segmentacije objekata je učenje algoritma na ručno označenim skupovima za učenje i validaciju te ispitivanje na ispitnom dijelu istog skupa podataka. S obzirom da ručno označavanje oduzima puno vremena i energije, motivacija za ovo istraživanje je upravo smanjenje takvog posla. Bez ikakvog prethodnog znanja o svrsi određene klase objekata, rješavanju ovog problema jedino mogu pomoći sličnost oblika komponenti i njihov prostorni razmještaj. Pri tome je izvođenje ovog algoritma u stvarnom vremenu vrlo korisno svojstvo pri praktičnoj primjeni u robotici.

Problem se, dakle, može izraziti na sljedeći način. Razmatra se skup 3D modela objekata pripadnika iste klase, predstavljenih mrežom trokuta, koji se u ovom poglavlju naziva baza modela. Postoje dostupni skupovi podataka takvog tipa. Algoritam odabire manji podskup reprezentativnih objekata iz toga skupa. Na svakom reprezentativnom objektu, stručnjak označava ciljnu komponentu za određenu radnju. Prema tim oznakama, algoritam identificira semantički korespondentne komponente na svakom od preostalih modela u bazi, koji se u ovom poglavlju nazivaju razmatrani ili ispitivani objekti (engl. *query*). Predloženi postupak omogućava jednostavno i brzo proširenje postojeće baze modela novim razmatranim objektima. U idealnom slučaju, dovoljno je označavanje komponenti jednog reprezentativ-

nog objekta kako bi se identificirale sve korespondentne komponente u danoj klasi objekata. Međutim, neke klase sadrže objekte koji se dosta razlikuju po obliku. Stoga je ponekad potrebno označavanje više od jednog objekta. Fokus ovog istraživanja je računalna efikasnost potrebna za praktičnu primjenu u robotici. Cilj je razviti metodu koja prepoznaje ciljnu komponentu na prvi puta viđenom objektu i dodaje taj objekt u postojeću bazu modela u nekoliko sekundi.

Predloženi se postupak temelji na detekciji konveksnih i konkavnih površina, koje se u ovom radu nazivaju *segmenti*. Komponente su ili predstavljene jednim ili s više segmenata. Odabrana komponenta reprezentativnog objekta pridružena je korespondentnim segmentima drugih instanci iste klase objekata kreiranjem *grafa pridruživanja komponenti* (engl. *component association graph*) (CAG). Čvorovi tog grafa predstavljaju sve segmente svih modela iz baze modela. Čvorovi su međusobno povezani vezama (engl. *edge*). Težina svake veze predstavlja mjeru vjerojatnosti da ta dva segmenta pripadaju komponentama objekta koji imaju istu svrhu. Ova mjera obuhvaća veličinu, oblik i poziciju segmenta te sličnost susjedstva. Predložena mjera sličnosti temelji se na CTI deskriptoru koji je opisan u poglavlju 3 te opisuje segmente objekata aproksimacijom njihova oblika konveksnim poliedrima. Sličnost susjedstva definirana je novim deskriptorom, koji se naziva *deskriptor topoloških odnosa* (engl. *Topological Relation Descriptor*) (TRED) te opisuje topološke odnose između dva segmenta modela. TRED je baziran na CTI deskriptoru. Tri metode za pridruživanje izabranih komponenti reprezentativnog modela segmentima drugih instanci objekata iste klase korištenjem CAG-a su predložene u ovom radu. Metoda *direktnog pridruživanja segmenata* povezuje svaki segment s njemu najbližim susjedom u CAG-u. Metoda *pridruživanja ograničenog na objekte* pridružuje segmente razmatranog objekta s komponentama svih reprezentativnih objekata koristeći pohlepno pretraživanje (engl. *greedy search*) te računa vrijednost podudaranja između razmatranog objekta i svih reprezentativnih objekata. Pridruživanja između razmatranog i reprezentativnih objekata s najvećom vrijednošću podudaranja se uzimaju kao konačni rezultat. Metoda *pridruživanja na temelju minimalno razapinjajućeg stabla* (engl. *minimum spanning tree*) (MST) pridružuje segmente razmatranog objekta sa segmentima reprezentativnih objekata na temelju njihove povezanosti u MST-u. Rezultat bilo koje od ovih triju metoda je pridruživanje segmenata svih modela iz baze modela ciljnoj komponenti reprezentativnog modela.

Poglavlje je strukturirano na sljedeći način. Potpoglavlje 5.2 donosi pregled istraživanja



u ovom području. Potpoglavlja 5.3-5.6 opisuju predloženu metodologiju. U potpoglavlju 5.3 dan je službeni opis problema, pregled predloženog pristupa i objašnjenje CAG-a. U potpoglavlju 5.4 opisana je mjera vjerojatnosti semantičke korespondencije, dodijeljena veza CAG-a. Tri metode za konačno pridruživanje segmenata i komponenti ciljnog oblika, zasnovane na CAG-u, predložene su u 5.5. Potpoglavlje 5.6 opisuje metodu za odabir reprezentativnih objekata. Eksperimentalna evaluacija i analiza rezultata opisane su u 5.7. Konačno, poglavlje je zaključeno potpoglavljem 5.8.

## 5.2 Istraživanja u ovom području

Semantička segmentacija u 3D primjenjuje se u autonomnoj vožnji, interakciji između čovjeka i stroja, robotskoj manipulaciji objektima, tvorničkoj proizvodnji, geometrijskom modeliranju i rekonstrukciji 3D scena, itd. Kako bi se potaknuo razvoj razumijevanja 3D oblika, organizirano je natjecanje, *the ShapeNet Challenge* (Yi et al., 2017b), za semantičku segmentaciju 3D oblika na velikim bazama 3D modela. Za eksperimentalnu evaluaciju metoda na ovom natjecanju, upotrijebljen je ShapeNet Parts, podskup od 16 klasa iz Shapenet skupa podataka (Chang et al., 2015), koji se također koristi i u eksperimentalnoj analizi metode predložene u ovom poglavlju ovog doktorskog rada. Metoda PointCNN koja je predstavljena na ShapeNet Challenge, kasnije je unaprijeđena (Li et al., 2018b) te predstavlja generalizaciju uobičajene konvolucijske neuronske mreže (engl. *convolutional neural network*) (CNN) za učenje značajki na temelju oblaka točaka. Točnost segmentacije PointCNN, eksperimentalno ispitana na skupu ShapeNet Parts, pokazala se najvećom za 7 klasa u usporedbi s drugih 14 prijavljenih metoda, dok je podjednaku točnost postigla na preostalih 9 klasa. Arhitektura CNN pokazala se najtočnijim i najefikasnijim pristupom u nedavnim usporedbama metoda dubokog učenja u svrhu semantičke segmentacije (Garcia-Garcia et al., 2017). Najveća točnost segmentacije objekata postignuta sa sedam metoda učenja, predstavljenih u (Yi et al., 2017b), bila je između 63 i 96 %, ovisno o klasi objekata.

Međutim, manualno označavanje velikih skupova podataka kakve zahtijeva učenje neuronskih mreža, vremenski je zahtjevan problem. Stoga se u radu (Yi et al., 2016) predlaže nova metoda aktivnog učenja koja je sposobna segmentirati velike geometrijske skupove podataka u točna semantička područja, uz dobar omjer točnosti i vremenske efikasnosti. Na taj je način označen i sam ShapeNet Parts skup. Cilj ovog pristupa - smanjenje ljudskog napora potrebnog pri označavanju velikih skupova podataka, također je osnovna motivacija

metode za pridruživanje komponenti predložene u ovom radu. Metoda (Yi et al., 2016) postiže označavanje velikih skupova podataka izmjenom ručnog označavanja dijelova, automatske propagacije ovih oznaka na ostale oblike, ručnog potvrđivanja ljudskog i automatskog označavanja te učenja na temelju rezultata verifikacije kako bi se poboljšavao algoritam automatske propagacije. Iako predloženi postupak (Yi et al., 2016) uključuje propagaciju oznaka komponenti s manjih skupova na velike, fokus tog istraživanja je na cjelokupni iterativni postupak označavanja. Nastavak istraživanja (Yi et al., 2016) predstavljen je u (Yi et al., 2017a). Ova metoda također propagira oznake s malog skupa na veliki primjenom globalne optimizacije, analogno postupku (Kalogerakis et al., 2010), koji se temelji na uvjetnom slučajnom području (engl. *conditional random field*). Istraživanje predstavljeno ovim doktorskim radom, s druge strane, fokusira se na učinkovitu detekciju ciljne komponente na razmatranom objektu na temelju malog označenog skupa reprezentativnih objekata, te mogućnost proširenja baze u stvarnom vremenu. Metode (Kalogerakis et al., 2010; Yi et al., 2017a) zahtijevaju dosta računskog vremena pa je motivacija za ovaj rad skraćenje tog vremena.

Automatizirani pristup određivanja semantičke segmentacije 3D modela predložen je u (Sun, 2018). Ovaj postupak koristi konkavne i konveksne značajke kao pomoć pri dekompoziciji objekata u strukturalne dijelove. Svakom od njih dodijeljena je semantička kategorija analizom njihove pozicije, oblika, veličine i konfiguracije. Predložena metodologija podsjeća na pristup predložen u ovom doktorskom radu. Dodjeljivanje semantičkih oznaka dijelovima radu (Sun, 2018) provedeno je stablom odlučivanja (engl. *decision tree*) i prilagođenom metodom potpornih vektora (engl. *support vector machine*), dok je u ovom doktorskom radu dodjeljivanje oznaka bazirano na CAG-u. Nadalje, metodologija predložena u (Sun, 2018) koristi se za semantičko označavanje zgrada pa je i deskriptor prilagođen toj primjeni, tj. opisuje relativnu visinu, volumen, omjer dimenzija itd.

Analogno metodama opisanima u ovom potpoglavlju, metoda predložena ovim doktorskim radom provodi segmentaciju cjelokupnih 3D modela. Međutim, kako bi se algoritam primijenio u praktičnim zadacima robotske manipulacije, potreban je cjelokupni 3D model danog razmatranog oblika, dobivenog na temelju podataka iz senzora. Neuronske mreže Pixel2Mesh (Wang et al., 2018a), AtlasNet (Groueix et al., 2018) i Occupancy network (ONet) (Mescheder et al., 2019) rekonstruiraju cjelokupni 3D model iz RGB slika. AtlasNet i ONet također prihvaćaju i 3D oblake točaka kao ulaz. Ove metode zahtijevaju da ispitivani

objekt bude izdvojen od pozadine. Dakle, da bi se ove metode mogle praktično primijeniti, potrebno ih je kombinirati s nekom metodom koja izdvaja objekte iz složene scene. Generative Shape Proposal Network, GSPN (Yi et al., 2018), obrađuje oblake točaka u svrhu rješavanja semantičke segmentacije. Ovaj postupak se bavi složenim scenama unutarnjeg prostora, djelomično vidljivim oblacima točaka i segmentacijom 3D modela na dijelove, što svakako predstavlja stvarne situacije u robotici. I ova metoda za učenje zahtijeva ručno označeni skup podataka. Problem smanjenja potrebe za ručnim označavanjem velikog skupa podataka se upravo razmatra u ovom poglavlju.

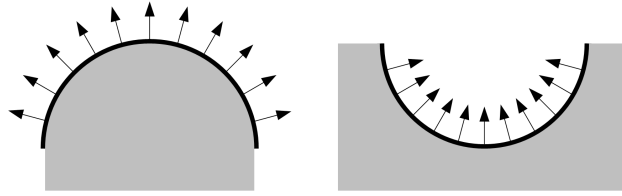
## 5.3 Pregled predložene metode

Neka je  $M$  skup 3D modela objekata pripadnika iste semantičke klase koji predstavlja *bazu modela*. Pretpostavljamo da su ti modeli predstavljeni mrežama trokuta  $P_k, k = 1, \dots, n_M$ . Razmatrani algoritam treba izabrati mali skup  $R$  reprezentativnih objekata i predstaviti ih stručnjaku koji označava komponente relevantne za određenu radnju na svakom objektu ovog skupa. Označavanje podrazumijeva odabir podskupa vrhova svake mreže  $P_r \in R$ , koji se u ovom radu nazivaju točke. Algoritam automatski dodjeljuje binarne oznake (engl. *label*) svakoj mreži  $P_k \in M \setminus R$ , dodjeljivanjem oznake 1 točkama koje predstavljaju ciljnu komponentu i oznake 0 ostalim točkama.

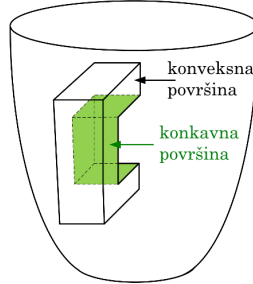
### 5.3.1 Detekcija komponenti

U predloženom postupku komponente objekta detektirane su segmentacijom površine objekta na konveksne i konkavne segmente na isti način kao što je to izvedeno u metodi opisanoj u potpoglavlju 4.7.4. Ovi segmenti predstavljaju prijedloge komponenti. U ovom radu se naziv konkavne površine koristi za označavanje inverznih konveksnih površina, npr. konveksnih površina sa suprotnim normalama lokalnih površina kao što je prikazano na slici 5.1. Svakom segmentu svakog modela iz baze modela dodijeljen je jedinstveni ID koji predstavlja par indeksa  $(i, k)$  pri čemu  $i$  označava indeks segmenta,  $k$  indeks modela. Jedna semantička komponenta može se predstaviti s više konveksnih ili konkavnih segmenata. Na primjer, drška šalice prikazana na slici 5.2 predstavljena je jednom konveksnom i jednom konkavnom površinom.

S obzirom da je segmentacija određenog oblika u nekim slučajevima višeznačna, generirani su i dodatni segmenti dobiveni spajanjem izvornih segmenata, kako bi se obuhvatilo što



Slika 5.1: Primjer konveksne (lijevo) i konkavne (desno) površine sa strelicama koje predstavljaju normale



Slika 5.2: Prikaz ručke šalice jednom konveksnom i jednom konkavnom plohom

više varijanti segmentacije. Algoritam za spajanje segmenata je objašnjen u nastavku.

Kriterij za odabir kandidata za spajanje zasniva se na blizini dvaju segmenata i konveksnosti njihove unije. Dva segmenta  $C_i$  i  $C_j$  zadovoljavaju kriterij blizine ako

$$\min(\mu_{ij}, \mu_{ji}) \geq -\tau_{prox,1}, \quad (5.1)$$

pri čemu je  $\tau_{prox,1} = 0.2$  eksperimentalno određen prag. Koeficijent vrste odnosa,  $\mu_{ij}$ , definiran je jednadžbom (5.9). Konveksnost unije dvaju segmenata je ispitana računanjem njihove konveksne ljuske i brojenjem odudarajućih točaka, tj. točki koje ne leže na konveksnoj ljusci unutar zadanog praga. Postotak odudarajućih točaka definiran je s

$$\eta(C_i, C_j) = \frac{|C_i \cup C_j| - |inliers(C_i \cup C_j, CH(C_i \cup C_j))|}{\min(|C_i|, |C_j|)}, \quad (5.2)$$

pri čemu  $|X|$  označava kardinalnost skupa  $X$ ,  $CH(X)$  označava konveksnu ljusku skupa  $X$ , a  $inliers(X, Y)$  označava skup podudarajućih točaka. Skup podudarajućih točaka sadrži sve točke  $p \in X$  koje su blizu površine konveksnog poliedra  $Y$ . Nadalje,  $p \in inliers(X, Y)$  samo ako je lokalna površina u okolini točke  $p$  slično orijentirana kao i površina  $Y$  u okolini točke  $p$ . Ovo je ispitano sljedećim kriterijem koji se zasniva na udaljenosti točke  $p$  od tangencijalne ravnine poliedra  $Y$  s normalom paralelnom normali lokalne površine u točki  $p$ .

$$\max_{p' \in V_Y} (n_p^T (p' - p)) \leq \tau_{prox,2},$$

Pri čemu je  $n_p$  normala lokalne površine u točki  $p$ ,  $V_Y$  je skup vrhova poliedra  $Y$ , a  $\tau_{prox,2}$  je eksperimentalno određen prag. U pokusima predstavljenima u potpoglavlju 5.7,  $\tau_{prox,2}$  je

postavljen na 5% radijusa granične sfere modela, pri čemu je približna minimalna granična sfera izračunata algoritmom predloženim u (Ritter, 1990). Ako  $\eta(C_i, C_j) < \tau_{new}$ , tada je novi segment kreiran iz unije  $C_i$  i  $C_j$ . Ako  $\eta(C_i, C_j) < \tau_{merge} < \tau_{new}$ , tada novi segment zamjenjuje  $C_i$  i  $C_j$ , tj. dva izvorna segmenta su odbačena. Vrijednosti pragova  $\tau_{new}$  i  $\tau_{merge}$ , koje se koriste u pokusima predstavljenim u potpoglavlju 5.7, su 0.2 i 0.05. Ovaj proces spajanja primijenjen je ne samo na izvorne segmente, već i na nove segmente nastale spajanjem izvornih segmenata. U tom slučaju, kandidati za spajanje su uvijek jedan izvorni i jedan novi segment. Ako je jedan segment izvorni,  $C_i$ , i drugi segment sastavljen od izvornih segmenata,  $C_j$ , tada su blizina i kriterij konveksnosti ispitani za svaki par segmenata  $(C_i, C_j)$ .

### 5.3.2 Graf pridruživanja komponenti (CAG)

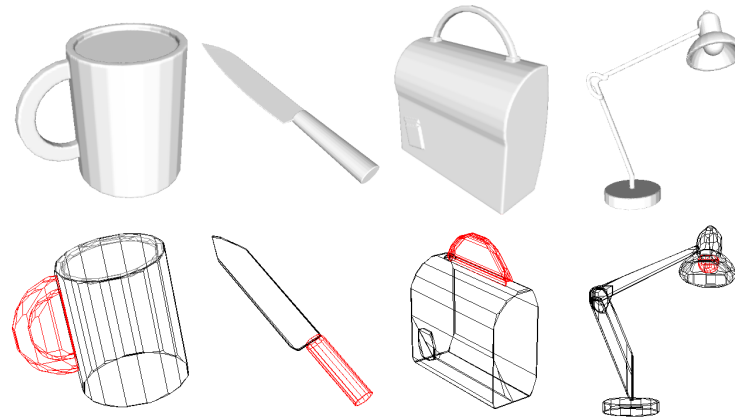
Metoda za određivanje veza između segmenata objekata iste klase, predložena ovim radom zasniva se na *grafu pridruživanja komponenti* (CAG). Čvorovi CAG-a predstavljaju segmente svih modela iz baze modela. Čvorovi su povezani vezama s dodijeljenim mjerama vjerojatnosti semantičke korespondencije (engl. *semantic correspondence likelihood measure*) (SCLM). Svaki segment razmatranog objekta je na taj način povezan s  $n$  čvorova s najvećom SCLM vrijednošću, koji se u ovom radu nazivaju najbližim susjedima. Broj čvorova ograničen je kako bi se smanjila računaska složenost, što je osobito važno u slučaju velikih baza modela. SCLM se sastoji od mjera sličnosti oblika, veličine i pozicije segmenta te mjere sličnosti susjedstva. U potpoglavlju 5.5 predložene su tri metode za utvrđivanje konačnih veza između segmenata i ciljnih komponenti zasnovane na CAG-u. Kao rezultat ovog procesa pridruživanja, svakom je segmentu u bazi dodijeljena oznaka vrijednosti 1, ako je taj segment pridružen ciljnoj komponenti, ili 0 ako nije.

## 5.4 Vjerojatnost semantičke korespondencije

Vjerojatnost da  $j$ -i segment razmatranog objekta  $C_j^Q$  i  $i$ -ti segment  $k$ -tog modela  $C_{ik}^M$  predstavljaju semantički korespondentne komponente ocijenjena je SCLM koja se računa na sljedeći način

$$y_{ijk} = y_{ijk}^C + w_N y_{ijk}^N, \quad (5.3)$$

gdje  $y_{ijk}^C$  predstavlja sličnost oblika, veličine i pozicije segmenta,  $y_{ijk}^N$  predstavlja sličnost susjedstva segmenta, a  $w_N$  je težinski faktor. Izračun sličnosti oblika, veličine, pozicije i



Slika 5.3: Prikaz objekata pomoću nekoliko CTI-ja

susjedstva opisan je u sljedećim potpoglavljima.

#### 5.4.1 Sličnosti oblika, veličine i pozicije segmenta

Sličnost oblika, veličine i pozicije dvaju segmenata mjeri se uspoređivanjem njihovih CTI deskriptora, opisanih u poglavlju 3. Četiri primjera objekata predstavljenih CTI-jevima prikazani su na slici 5.3. Kako bi se smanjila računaska složenost, CTI deskriptor,  $d$ , projicira se na niže-dimenzionalni latentni prostor, kao što je predloženo u poglavlju 4. Deskriptori  $q$  sa  $n_q < n_d$  elemenata dobiveni su na sljedeći način

$$q = M^T d, \quad (5.4)$$

pri čemu  $M$  predstavlja ortonormiranu bazu koja definira latentni prostor dobiven pomoću metode glavnih komponentata (PCA) na CTI deskriptorima  $d$  dobivenim na temelju skupa za učenje. Druga svrha računanja latentnog vektora je odvajanje informacija o obliku i poziciji.

Prva tri elementa vektora  $q$ , u ovom radu označeni sa  $q^t$ , predstavljaju poziciju segmenta u koordinatnom sustavu objekta, dok vektor ostalih 21 elemenata, u ovom radu označen s  $q^s$ , opisuje oblik i veličinu. Ako se razmatra baza modela  $B_M$  s  $n_M$  modela, tada su segmenti  $k$ -tog modela predstavljeni latentnim vektorima  $q_{ik}^M, i = 1, \dots, n_{M,k}$ .

Sličnost između dva segmenta koja uzima u obzir njihovu poziciju ( $t$ ), oblik ( $s$ ) i veličinu ( $a$ ) mjeri se Gausovom funkcijom

$$y_{ijk}^C = \exp \left( -\frac{1}{2} \left( \frac{e_{ijk}^t}{\sigma_t^2} + \frac{e_{ijk}^s}{\sigma_s^2} + \frac{e_{ijk}^a}{\sigma_a^2} \right) \right), \quad (5.5)$$

gdje su  $\sigma_t$ ,  $\sigma_s$  i  $\sigma_a$  parametri koji opisuju doprinos razlike između pozicije, oblika i veličine segmenta ukupnoj mjeri sličnosti. Vrijednosti parametara algoritma koji se koriste u ekspe-

rimentima kao i sami eksperimenti opisani su u potpoglavlju 5.7

Razlike u translaciji, obliku i skali računaju se sljedećim izrazima.

$$e_{ijk}^t = \frac{\|q_j^{Q,t} - q_{ik}^{M,t}\|^2}{\|q_j^{Q,s}\| \|q_{ik}^{M,s}\|}, \quad (5.6)$$

$$e_{ijk}^s = 1 - \left( \left( \frac{q_j^{Q,s}}{\|q_j^{Q,s}\|} \right)^T \left( \frac{q_{ik}^{M,s}}{\|q_{ik}^{M,s}\|} \right) \right)^2, \quad (5.7)$$

$$e_{ijk}^a = \frac{\left( \|q_j^{Q,s}\| - \|q_{ik}^{M,s}\| \right)^2}{\|q_j^{Q,s}\| \|q_{ik}^{M,s}\|}. \quad (5.8)$$

U jednadžbi (5.6), razlika u poziciji podijeljena je s veličinom segmenta te se na taj način omogućava da veći segmenti imaju veću udaljenost, a postižu jednaku vrijednost mjere sličnosti. Jednadžba (5.7) predstavlja skalarni produkt između dva jedinična vektora. Što je veća vrijednost skalarnog produkta, to su oblici sličniji. Norma vektora  $q^s$  predstavlja mjeru veličine segmenta. Jednadžba (5.8) predstavlja mjeru razlike u veličini između segmenata.

### 5.4.2 Sličnost susjedstva

Pretpostavimo dva segmenta,  $C_i$  i  $C_j$ , opisana CTI deskriptorima  $d_i$  i  $d_j$ , koji pripadaju istom modelu. Kako bi se opisao geometrijski razmještaj segmenata u modelu, uvodi se novi deskriptor topoloških odnosa (TRED) koji opisuje toploške odnose između segmenata. TRED predstavlja n-torku  $T(C_i, C_j) = (\mu_{ij}, v_{ij}, \sigma_{ij}^v)$ . Koeficijent vrste odnosa,  $\mu_{ij}$ , se računa na sljedeći način

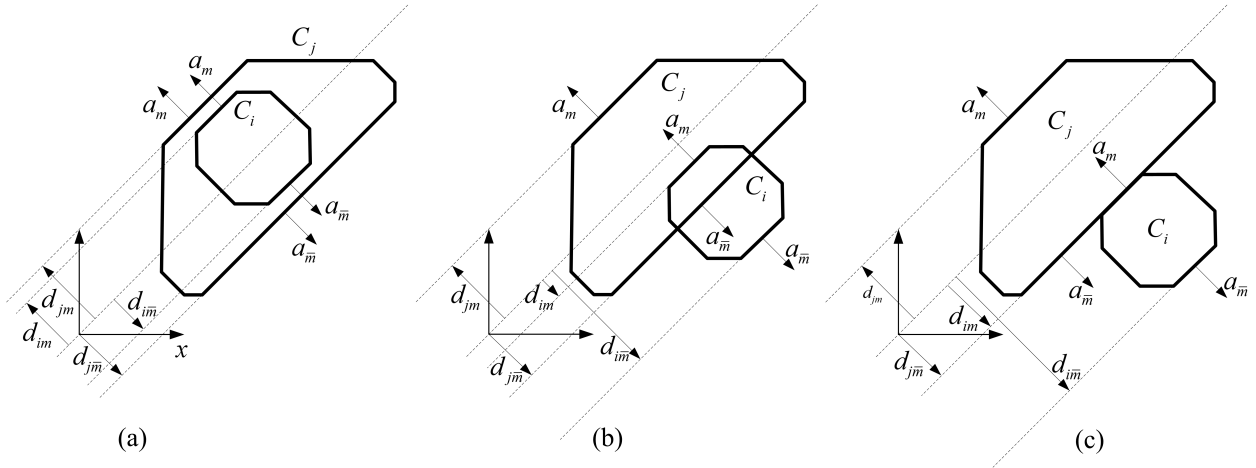
$$\mu_{ij} = \min_{m=1, \dots, n_d} \rho_{ijm}, \quad (5.9)$$

pri čemu

$$\rho_{ijm} = \frac{d_{im} + d_{j\bar{m}}}{d_{im} + d_{i\bar{m}}}. \quad (5.10)$$

U jednadžbi (5.10)  $m$  i  $\bar{m}$  predstavljaju indekse dva suprotna jedinična vektora  $a_m, a_{\bar{m}} \in A$ . Razmatra se pet vrsta toploških odnosa između segmenata  $C_i$  i  $C_j$ , određenih s  $\mu_{ij}$ :

1.  $C_j$  sadrži  $C_i$



Slika 5.4: Deskriptor topoloških odnosa (TRED). Elementi CTI deskriptora  $d_{xm}$  koji su označeni suprotno usmjerenim strelicama od njima odgovarajućih vektora  $a_m$  imaju negativnu vrijednost.

$$\mu_{ij} \geq 1 \wedge \mu_{ij} > \mu_{ji}$$

2.  $C_i$  i  $C_j$  su identični

$$\mu_{ij} = \mu_{ji} = 1$$

3.  $C_i$  i  $C_j$  se sijeku

$$0 < \mu_{ij} < 1 \wedge 0 < \mu_{ji} < 1$$

4.  $C_i$  dodiruje  $C_j$

$$\mu_{ij} = 0$$

5.  $C_i$  i  $C_j$  su disjunktne

$$\mu_{ij} < 0.$$

Vrsta odnosa između dva segmenta u ovom se radu označava s  $type(T(C_i, C_j))$ . Tri vrste odnosa prikazane su slikom 5.4: segment  $C_i$  je unutar segmenta  $C_j$  (lijevo), dva segmenta  $C_i$  i  $C_j$  se sijeku (sredina) i segment  $C_i$  dodiruje segment  $C_j$  (desno).

Ako se dva segmenta dodiruju  $v_{ij}$  i  $\sigma_{ij}^v$  se računaju na sljedeći način

$$v_{ij} = \frac{\sum_{m=1}^{n_d} w_{ijm} a_m}{\left\| \sum_{m=1}^{n_d} w_{ijm} a_m \right\|}, \quad (5.11)$$



$$\sigma_{ij}^v = \sqrt{\frac{\sum_{m=1}^{n_d} w_{ijk} \arccos^2 v_{ij}^T a_m}{\sum_{m=1}^{n_d} w_{ijk}}} + \sigma_{v0}^2, \quad (5.12)$$

pri čemu

$$w_{ijm} = \exp\left(-\frac{1}{2} \left(\frac{\mu_{ij} - \rho_{ijm}}{\sigma_\rho}\right)^2\right). \quad (5.13)$$

Vektor  $v_{ij}$  je približno okomit na ravninu koja razdvaja  $C_i$  i  $C_j$  te je usmjeren od  $C_i$  prema  $C_j$ . Njegova je svrha opisati smjer dodira. Ovaj se vektor računa kao otežani prosjek svih vektora  $a_m$ , pri čemu vektori pridruženi manjim vrijednostima  $\rho_{ijm}$  imaju veće težine. Vrijednost  $\sigma_{ij}^v$  opisuje nesigurnost vektora  $v_{ij}$ , pri čemu je  $\sigma_{v0}$  konstanta koja opisuje nesigurnost uzrokovanu mjernim šumom. TRED-ovi se računaju za svaki par segmenata modela. Segment  $C_j$  je susjed segmenta  $C_i$  ako je su  $C_i$  i  $C_j$  u odnosu 1 ili 4. Kako bi metoda bila robusna na mjerni šum, definicija vrsta topoloških odnosa relaksirana je uvođenjem odgovarajuće tolerancije. Kao posljedica ove relaksacije, topološki odnos vrste 2 obuhvaća i segmente koji zauzimaju približno jednak prostor, a ne samo identične segmente.

Nadalje, ako su dva segmenta različite konveksnosti (konveksni i konkavni), tada se smatraju susjedima samo ako su u topološkom odnosu vrste 1 ili 2. Susjedi svakog segmenta grupirani su u klaster na temelju pridruženih TRED deskriptora. Ovi se klasteri u ovom radu označavaju s  $\Gamma$ , a skup klastera dodijeljen segmentu  $C_p^X$  označava se s  $N_p^X$ , pri čemu  $X$  predstavlja razmatrani ( $Q$ ) ili reprezentativni ( $M$ ) model. Dva susjeda od  $C_i$  grupirana su u klaster samo ako je njihov topološki odnos sa  $C_i$  jednake vrste. U slučaju odnosa vrste 4, dva susjeda  $C_j$  i  $C_k$  mogu biti grupirana u isti klaster ako su njihovi vektori  $v_{ij}$  i  $v_{ik}$  slični. Vektori  $v_{ij}$  i  $v_{ik}$  smatraju se sličnima ako je kut između njih  $\leq 60^\circ$ . U slučaju odnosa vrste 1, svi susjedi  $C_j$  takvi da  $\mu_{ij} > \mu_{ji}$  grupirani su u jedan klaster i svi susjedi  $C_j$  takvi da  $\mu_{ij} < \mu_{ji}$  grupirani su u drugi. Svi susjedi različite konveksnosti od  $C_i$  čiji je topološki odnos vrste 2 grupirani su u jedan klaster.

Sličnost između susjedstva segmenta  $C_j^Q$  razmatranog objekta i segmenta  $C_{ik}^M$   $k$ -tog modela mjeri se sparivanjem njihovih susjednih klastera. Neka su  $C_m^Q$  i  $C_{lk}^M$  susjedi od  $C_j^Q$  i  $C_{ik}^M$ . Sličnost između ta dva segmenta mjeri se sljedećim izrazom

$$y_{lmk}^n = \varepsilon_{lmk} \exp\left(-\frac{1}{2} \left(\frac{e_{ijk}^s}{\sigma_s^2} + \frac{e_{ijk}^a}{\sigma_a^2}\right)\right), \quad (5.14)$$

pri čemu je  $\varepsilon_{lmk}$  binarna varijabla, koja ima vrijednost 1 samo ako

$$\text{type}(T(C_j^Q, C_m^Q)) = \text{type}(T(C_{ik}^M, C_{lk}^M)),$$

dok u suprotnom ima vrijednost 0. Nadalje, dva su susjeda  $C_m^Q$  i  $C_{lk}^M$  vrste 4 sparena samo ako vektori  $v_{jm}$  i  $v_{ikl}$  pridruženih TRED-ova imaju dovoljno sličan smjer, uzimajući u obzir njihove nesigurnosti opisane s  $\sigma_{jm}^v$  i  $\sigma_{ikl}^v$ . Inače,  $\varepsilon_{lmk} = 0$ .

Sličnost dva susjedna klastera  $\Gamma_{rj}^Q \in N_j^Q$  i  $\Gamma_{pik}^M \in N_{ik}^M$ , gdje  $p$  i  $r$  predstavljaju indekse klastera, mjeri se sličnošću oblika i veličine dviju najbližijih komponenti koje pripadaju tim dvama klasterima. Mjera sličnosti ta dva klastera računa se na sljedeći način

$$y^\Gamma(\Gamma_{rj}^Q, \Gamma_{pik}^M) = \max_{\substack{C_m^Q \in \Gamma_{rj}^Q \\ C_{lk}^M \in \Gamma_{pik}^M}} y_{lmk}^n.$$

Mjera sličnosti susjedstva računa se primjenom sljedećeg postupka. Prvo se stvori prazni skup  $B_{ijk}^N$ . Nakon toga se odredi par klastera s najvećim vrijednostima  $y^\Gamma$  i spremi u  $B_{ijk}^N$ . Ova se korak ponavlja za sve preostale klastera sve dok se više ne može formirati ni jedan par klastera. Na kraju se konačni skup  $B_{ijk}^N$  koristi pri računanju mjere sličnosti susjedstva prema sljedećoj formuli

$$y_{ijk}^N = \sum_{(\Gamma, \Gamma') \in B_{ijk}^N} y^\Gamma(\Gamma, \Gamma').$$

## 5.5 Pridruživanje segmenta i ciljne komponente

Predložene su tri varijante pridruživanja ciljne komponente segmentu razmatranog objekta zasnovane na CAG-u. Samo stvaranje CAG-a neovisno je o primijenjenoj varijanti.

**Direktno pridruživanje segmenata** (engl. *Direct Segment Association*) Segmentu razmatranog objekta dodjeljuje se ID najbližeg susjeda u CAG-u koji predstavlja segment reprezentativnog objekta.

**Pridruživanje ograničeno na objekte** (engl. *Object-Constrained Association*)

Ova metoda uspoređuje razmatrani model sa svakim reprezentativnim modelom. Svaki segment razmatranog modela pridružuje se svakoj komponenti reprezentativnog modela po hlepnom pretragom (engl. *greedy search*). Razmatranom segmentu dodjeljuje se ID komponente reprezentativnog modela s najvećom SCLM. Razmatrani segment je isključen iz

daljnje pretrage te se pretraga nastavlja dok svi razmatrani segmenti nisu pridruženi s korespondentnim segmentima reprezentativnog objekta. Vrijednost podudaranja objekta između razmatranog i reprezentativnog modela računa se kao suma SCLM-a između pridruženih segmenata razmatranog i segmenata reprezentativnog modela koji predstavljaju ciljnu komponentu. Segmentima razmatranog modela se konačno dodjeljuje ID pridruženih segmenata reprezentativnog modela s najvećom vrijednosti podudaranja objekata.

**Pridruživanje zasnovano na MST (engl. *MST-Based Association*)** Korijenski čvor koji ne predstavlja niti jedan segment dodan je u CAG. Nakon toga, kreira se minimalno razapinjajuće stablo (MST) uz uvjet da su svi segmenti reprezentativnog objekta direktno povezani s korijenskim čvorom. Stoga, svaki segment svakog reprezentativnog modela razapinje jedno podstablo. Segmentima razmatranih objekata dodijeljuje se ID segmenta reprezentativnog modela u istom podstablu.

Na kraju bilo koje predstavljene procedure, svaki segment razmatranog objekta ima dodijeljen ID segmenta reprezentativnog objekta. Konačno, segment razmatranog objekta nasljeđuje oznake segmenta pridruženog reprezentativnog objekta. Prva i treća varijanta pridružuju svaki segment razmatranog objekta s ID-em korespondentnog segmenta reprezentativnog objekta, neovisno o oznakama segmenata reprezentativnog objekta. Tako da se ove metode mogu primijeniti prije označavanja ciljne komponente reprezentativnog objekta.

## 5.6 Izbor reprezentativnih objekata

Razmatra se baza modela  $M$  koja predstavlja skup modela  $M_k$  objekata koji pripadaju istoj klasi objekata. Svaki model  $M_k$  predstavljen je 3D mrežom trokuta. Kao što je objašnjeno, svaka se mreža segmentira u konveksne i konkavne segmente te se svaki od segmenata opisuje CTI deskriptorom. Kako bi se olakšalo objašnjavanje predloženog postupka, model  $M_k$  predstavljen je sekvencom skupova točaka  $M_k = (C_{1k}, \dots, C_{n_{M,k}k}, P_k)$ , pri čemu je  $C_{ik}$   $i$ -ti segment modela, a  $P_k$  je skup 3D točaka koje leže na površini  $k$ -tog modela. U svrhu računske učinkovitosti, ove se točke mogu dobiti poduzorkovanjem vrhova mreže. Sličnost  $k$ -tog i  $l$ -tog objekta procjenjuje se mjerenjem udaljenosti između točaka  $k$ -tog modela i CTI-a  $l$ -tog modela i obratno. Mjera sličnosti objekata računa se na sljedeći način

$$y_{kl}^O = y_{lk}^O = z_{kl}^O \cdot z_{lk}^O, \quad (5.15)$$

pri čemu

$$z_{kl}^O = \frac{1}{|P_k|} \sum_{p \in P_k} \exp \left( -\frac{1}{2} \frac{\delta_{pM}^2(p, M_l)}{\sigma_p^2} \right) \quad (5.16)$$

$$\delta_{pM}(p, M_l) = \min_{C_{ji} \in M_l} (\delta_{pC}(p, C_{jl})) \quad (5.17)$$

$$\delta_{pC}(p, C_{jl}) = \max_{m=1, \dots, n_d} (a_m^T p - d_{jlm}) \quad (5.18)$$

Parametar  $\sigma_p$  u (5.16) predstavlja eksperimentalno određenu konstantu. Ako je  $C_{jl}$  konkavni segment, tada se u (5.18) umjesto max primjenjuje operacija min. Treba napomenuti da  $y_{kl}^O \in [0, 1]$ .

Skup  $R$  reprezentativnih objekata odabran je pohlepnim algoritmom koji maksimizira sličnost između  $R$  i svih objekata  $M_k \in M$ . Sličnost između modela objekta  $M_k \in M$  i skupa  $R$  definira se kao sličnost između ovog modela i najbližijeg modela  $M_r \in R$ . Ova se sličnost mjeri vrijednošću  $y_k^R$  koja se računa kao

$$y_k^R = \max_{M_r \in R} y_{kr}^O.$$

Sličnost između  $M$  i  $R$  mjeri se vrijednošću  $y^{MR}$  koja predstavlja sumu ukupnih vrijednosti  $y_k^R$

$$y^{MR} = \sum_{k=1}^{|M|} y_k^R.$$

U postupku predloženom u ovom radu, skup reprezentativnih objekata  $R$  odabire se iterativnim postupkom, pri čemu se u svakoj iteraciji odabire model koji maksimizira  $y_{MR}$  sve dok predefimirani broj reprezentativnih objekata  $n_R$  nije izabran.

## 5.7 Eksperimentalna evaluacija

Predloženi je postupak eksperimentalno provjeren koristeći skup 3D modela ShapeNet (Chang et al., 2015). Dio ovog skupa namijenjen je ispitivanju metode koja segmentira 3D modele u semantičke dijelove. Ovaj skup podataka sastoji se od 16 podskupova klasa objekata. Svaki podskup podijeljen je u skup za učenje te skupove za validaciju i ispitivanje. Pri tome je skup za učenje znatno veći od skupova za validaciju i ispitivanje. Zamišljeno je da se ovaj skup koristi na sljedeći način: metoda se uči na ručno označenim skupovima za učenje i validaciju te se ispituje na skupu za ispitivanje. U ovom se radu koristi drugačiji

pristup koji segmentira neoznačene 3D modele određene klase objekata na segmente te povezuje te segmente na temelju sličnosti njihova oblika, veličine i geometrijskog razmještaja. Dobivene veze služe pri identifikaciji korisnički zadane komponente svih modela na temelju malog ručno označenog skupa reprezentativnih objekata. Kako bi se automatizirao postupak, umjesto ručnog označavanja reprezentativnog skupa, koristile su se oznake dostupne za svaki model u skupu.

S obzirom da je ciljna primjena predloženih metoda omogućavanje robotskih radnji, izabrano je 6 klasa objekata iz razmatranog skupa podataka, koji se mogu povezati s točno definiranom robotskom radnjom: šalice, noževi, torbe, svjetiljke, kape i prijenosna računala (laptopi). Za šalice, noževe i torbe, ciljna komponenta je drška koju robot treba uhvatiti. Kod svjetiljki, ciljna komponenta je žarulja koju robot treba promijeniti. Ciljna komponenta kape je obod za radnju oblačenja, a u slučaju prijenosnih računala, to je ekran za radnju otvaranja i zatvaranja.

U slučaju svih klasa osim svjetiljki, upotrijebljene su izvorne oznake ShapeNet skupa, gdje su ručka, obod i ekran označeni kao posebna semantička kategorija. U slučaju svjetiljki, izabran je podskup od 210 objekata s jasno odvojenom žaruljom te su žarulje ručno označene.

Sljedeća je procedura primijenjena na svaku od šest razmatranih klasa objekata, s vrijednostima parametara danima u tablici 5.1.

Tablica 5.1: Vrijednosti parametara

| $n_d$ | $n_q$ | $\sigma_t$ | $\sigma_s$ | $\sigma_a$ | $\sigma_{v0}$ | $\sigma_p$ | $\tau_{tr}$ | $\tau_v$ |
|-------|-------|------------|------------|------------|---------------|------------|-------------|----------|
| 66    | 24    | 2.4        | 0.132      | 0.707      | 0.1           | 0.025      | 0.333       | 2        |

1. Svaki je model segmentiran u konveksne i konkavne segmente primjenom postupka predloženog u (Cupec et al., 2020).
2. Provedeno je spajanje segmenata opisano u 5.3.1.
3. Svaki je segment predstavljen CTI deskriptorom. Analogno poglavljima 3 i 4, upotrijebljen je konveksni predložak koji se sastoji od  $n_d = 66$  jediničnih vektora jednoliko raspoređenih po jediničnoj kugli.
4. TRED deskriptor, predstavljen u 5.4, računa se za svaki par segmenata svakog modela.
5. Identificirani su susjedni klasteri te su dodijeljeni svakom segmentu.

6. CTI deskriptor svakog segmenta projiciran je na korespondentni latentni prostor pomoću (5.4).
7. Za svaka dva segmenta različitih modela u bazi modela računa se SCLM, predložena u 5.4.
8. Stvara se CAG. Čvorovi ovog grafa predstavljaju sve segmente svih modela u bazi modela. Svaki je čvor povezan s najviše 100 najbližijih segmenata, na temelju SCLM izračunate u koraku 7.
9. Metodom predloženom u potpoglavlju 5.6 izabran je mali broj reprezentativnih objekata. Ovaj je broj  $\leq 10\%$  svih modela u razmatranoj bazi modela.
10. Na reprezentativnim objektima označena je ciljna komponenta U praktičnoj primjeni, ovaj korak obavlja čovjek. U predstavljenim eksperimentima, postojeće oznake, koje su dostupne za sve modele, dodijeljene su reprezentativnim modelima objekata.
11. Svi segmenti svih preostalih modela automatski su označeni primjenom CAG-a kreiranog u koraku 8 i triju metoda predloženih u 5.5, DSA, OCA i MST.
12. Rezultat ostvaren za svaki model uspoređen je s referentnim (engl. *ground truth*) podacima primjenom *Intersection over Union* (IoU) indeksa.

Koraci 1-8, računanje sličnosti objekata u koraku 9 i stvaranje MST-a u koraku 11 implementirani su u C++ programskom jeziku. Ostatak koraka 9, 11 i korak 12 implementirani su u MATLAB-u.

Izračun latentnog vektora, proveden u koraku 6 zahtijeva ortonormiranu bazu koja definira latentni prostor. Ta je baza predstavljena  $n_d \times n_q$  matricom  $M$ . Matrica  $M$  se računa procedurom za učenje predloženoj u poglavlju 3, gdje je jednostavniji CTI deskriptor korišten umjesto složenog VolumeNet deskriptora, predloženog u poglavlju 4. Učenje se provodi koristeći 3DNet skup podataka (Wohlkinger et al., 2012). Ukupno 351 mreža objekata pripadnika 10 klasa segmentirana je u konveksne i konkavne površine te je CTI deskriptor određen za svaku površinu. Svaki CTI deskriptor predstavlja točku u  $n_d$ -dimenzionalnom prostoru. Dobiveni deskriptori podijeljeni su u dva podskupa, jedan predstavlja konveksne, a drugi konkavne segmente. Matrica  $M$  računa se za svaki skup deskriptora na temelju analize glavnih komponenti (PCA) predložene u poglavlju 4. I za konveksne i za konkavne segmente,

generira se latentni prostor od  $n_q = 24$  dimenzije. Ovaj broj dimenzija izabran je kao najmanji broj prvih glavnih komponenti, tako da su varijance preostalih glavnih komponenti manje ili jednake  $10^{-4}$  za oba skupa, skup konveksnih i skup konkavnih segmenata.

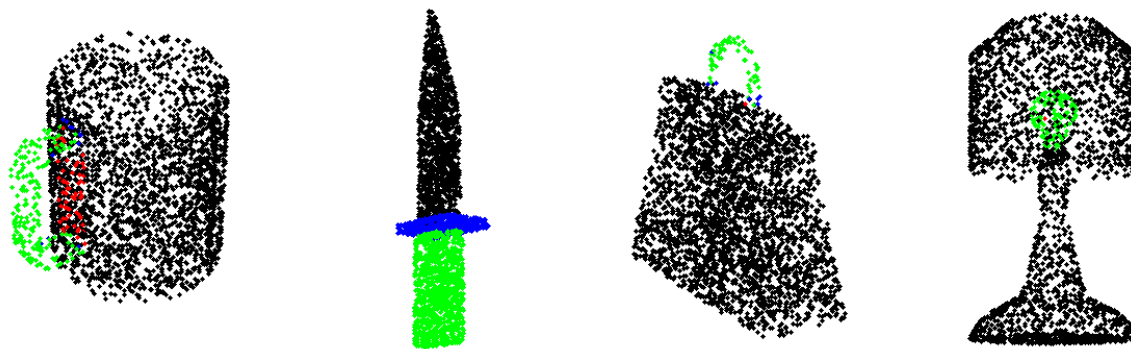
Oznake komponenti dostupne su u skupu ShapeNet za svaki model u obliku skupa točaka sa dodijeljenim oznakama. S obzirom da predloženi postupak pridružuje segmente, segmenti moraju biti pridruženi označenim točkama. Svaka je točka pridružena najbližem segmentu. Nadalje, računa se udaljenost između svake točke i CTI-a svakog segmenta te je svaka točka pridružena segmentu ako je udaljenost do CTI-a tog segmenta manja ili jednaka 0.001. Ova udaljenost računa se pomoću (5.18). Treba napomenuti da ovo nije euklidska udaljenost, ali je njezina dobra aproksimacija, koja se može vrlo efikasno izračunati. Isti prag udaljenosti može se koristiti za sve objekte s obzirom da su u ShapeNet skupu svi objekti skalirani na približno jednaku veličinu. Ovo pridruživanje točka-segment omogućava da se svaka točka pridruži više segmenata.

U koraku 10 ove eksperimentalne procedure, segment reprezentativnog objekta označen je kao dio ciljne komponente ako je više od pola točaka kojima je pridružen označeno. Korak 11 označava segmente svih modela iz razmatrane baze modela, osim reprezentativnih objekata, koji su već označeni u koraku 10. Kako bi se izračunala IoU u koraku 12, oznaka se mora prenijeti sa segmenata na točke. Točka je označena kao da pripada ciljnoj komponenti ako je pridružena najmanje jednom označenom segmentu.

### 5.7.1 Rezultati

Nekoliko je rezultata prikazano slikom 5.5. Točno pozitivno pridružene točke (engl. *True positive*), tj. točke koje su ispravno pridružene ciljnoj komponenti, označene su zelenom bojom, dok su netočno pozitivno pridružene točke (engl. *false positive*), tj. točke koje su neispravno pridružene ciljnoj komponenti, označene crvenom bojom. Netočno negativno pridružene točke (engl. *False negative*), tj. točke koje predstavljaju ciljne točke komponenti, ali nisu pridružene ciljnoj komponenti predloženim algoritmom, označene su plavom bojom.

Kvantitativni rezultati dobiveni opisanom eksperimentalnom procedurom prezentirani su u tablici 5.2. Svaka od tri metode pridruživanja komponenti ispitana je s i bez sličnosti susjedstva, tj. uz parametar  $w_N$  jednak 1 i 0. Broj reprezentativnih objekata bio je najveći cijeli broj  $\leq 10\%$  svih modela u bazi modela svake klase. Predstavljene IoU vrijednosti označavaju prosjek za sve modele određene klase, bez reprezentativnih.



Slika 5.5: Primjeri rezultata

Tablica 5.2: Točnost automatskog označavanja komponenti [%]

| Klasa                |           | šalica       | nož          | torba        | svjetiljka   | kapa         | laptop       |
|----------------------|-----------|--------------|--------------|--------------|--------------|--------------|--------------|
| ukupni broj objekata |           | 184          | 391          | 68           | 210          | 53           | 451          |
| broj repr. objekata  |           | 18           | 39           | 6            | 21           | 5            | 45           |
| DSA                  | $w_N = 0$ | <b>78.50</b> | 75.40        | 44.10        | 43.10        | 82.50        | 75.30        |
|                      | $w_N = 1$ | 77.40        | 73.30        | 50.30        | 57.20        | <b>84.80</b> | <b>77.20</b> |
| IoU OCA              | $w_N = 0$ | 73.90        | 58.00        | 48.00        | 53.60        | 82.40        | 58.90        |
|                      | $w_N = 1$ | 75.10        | 62.40        | 43.60        | <b>65.40</b> | 82.40        | 59.30        |
| MST                  | $w_N = 0$ | 77.90        | <b>75.90</b> | 42.60        | 36.80        | 82.70        | 76.70        |
|                      | $w_N = 1$ | 76.30        | 73.20        | <b>60.90</b> | 53.00        | 75.90        | 76.10        |

Kako bi se rezultati predložene metode usporedili s referentnim metodama, proveden je sljedeći pokus. Za svaku od pet klasa: šalice, noževi, torbe, kape i prijenosna računala, provedeni su koraci 1-10, koji obuhvaćaju odabir i označavanje reprezentativnih objekata, na skupu za učenje. Treba napomenuti da je samo mali podskup reprezentativnih objekata na skupu za učenje označen. Broj označenih reprezentativnih objekata dan je u tablici 5.2. Nakon toga, baza je proširena (koraci 1-8) ispitnim dijelom skupa podataka te su dodijeljene oznake modelima iz ovog skupa. Analiza točnosti provedena je koracima 11 i 12. Rezultati analize točnosti na ispitnom skupu mjerene pomoću IoU nad svakom klasom, dani su tablici 5.3. Na dnu tablice nalaze se točnosti postignute s trima metodama (DSA, OCA i MST) predloženima u ovom radu. S obzirom da ovaj rad istražuje pridruživanje komponenti s malim brojem reprezentativnih objekata, nije bilo za očekivati da će predložena metoda postići bolje rezultate od metoda koje su iscrpno učene na velikim označenim skupovima.



Tablica 5.3: Usporedba točnosti s relevantnim metodama [%]

| Metoda / Klasa                        |           | šalica       | nož          | torba        | kapa         | laptop       |
|---------------------------------------|-----------|--------------|--------------|--------------|--------------|--------------|
| SyncSpecCNN (Yi et al., 2017c)        |           | 92.73        | 86.10        | 81.72        | 81.94        | 95.61        |
| Pd-Network (Klokov i Lempitsky, 2017) |           | 94.00        | 87.25        | 82.42        | 87.04        | 95.44        |
| SSCN (Graham et al., 2018)            |           | 95.23        | <b>89.10</b> | 82.99        | 83.97        | 95.78        |
| SpiderCNN (Xu et al., 2018)           |           | 93.50        | 87.30        | 81.00        | 87.20        | 95.80        |
| SO-Net (Li et al., 2018a)             |           | 94.20        | 83.90        | 77.80        | <b>88.00</b> | 94.80        |
| PCNN (Atzmon et al., 2018)            |           | 94.80        | 86.00        | 80.10        | 85.50        | 95.70        |
| KCNet (Shen et al., 2018)             |           | 94.40        | 87.20        | 81.50        | 86.40        | 95.50        |
| Kd-Net (Klokov i Lempitsky, 2017)     |           | 86.70        | 87.20        | 74.60        | 74.30        | 94.90        |
| 3DmFV-Net (Ben-Shabat et al., 2017)   |           | 94.00        | 85.70        | 84.30        | 86.00        | 95.20        |
| RSNet (Huang et al., 2018)            |           | 92.60        | 87.00        | 86.40        | 84.10        | 95.40        |
| DGCNN (Wang et al., 2018b)            |           | 93.30        | 87.30        | 83.70        | 84.40        | 96.00        |
| PointNet (Qi et al., 2017a)           |           | 93.00        | 85.90        | 78.70        | 82.50        | 95.30        |
| PointNet++ (Qi et al., 2017b)         |           | 94.10        | 85.90        | 79.00        | 87.70        | 95.30        |
| SGPN (Wang et al., 2019)              |           | 93.80        | 83.00        | 78.60        | 78.80        | 95.80        |
| PointCNN (Li et al., 2018b)           |           | <b>95.28</b> | 88.44        | <b>86.47</b> | 86.04        | <b>96.11</b> |
| DSA                                   | $w_n = 0$ | 82.14        | 75.53        | 39.67        | 69.59        | 75.54        |
|                                       | $w_n = 1$ | 76.65        | 73.42        | 41.66        | 74.76        | 74.53        |
| OCA                                   | $w_n = 0$ | 75.74        | 56.35        | 36.24        | 81.45        | 59.67        |
|                                       | $w_n = 1$ | 76.47        | 59.18        | 40.34        | 80.28        | 59.95        |
| MST                                   | $w_n = 0$ | 77.68        | 75.59        | 34.92        | 75.57        | 76.74        |
|                                       | $w_n = 1$ | 75.80        | 64.78        | 44.01        | 72.32        | 65.91        |

Kako bi se ispitalo je li metoda prikladna za izvođenje u stvarnom vremenu, eksperimentalno je ispitano vrijeme izvođenja automatskog označavanja komponenti. U ovim je pokusima u SCLM uključena mjera sličnosti susjedstva ( $w_N = 1$ ). Rezultati su prikazani u tablici 5.4. Prikazano je i prosječno vrijeme izvođenja svakog koraka po objektu. Nadalje, prikazano je i ukupno vrijeme izvođenja svih koraka potrebno za proširenje baze modela s novim razmatranim objektom. Pretpostavlja se da je, prije proširenja, baza sadržavala ukupan broj objekata prikazan u tablici 5.4 minus jedan. Prosječno vrijeme izvođenja po modelu prikazano je za korake 1-8, implementirane u C++ te za korake 11 i 12, implementirane u MATLAB-u. U slučaju MST metode, vrijeme izvođenja koraka 11 uključuje i izračun MST-a implementiranog u C++. Vrijeme izvođenja izbora reprezentativnih objekata u koraku 9, potrebno je samo kada se izabiru reprezentativni objekti. Kada se baza proširuje bez označavanja novog reprezentativnog objekta, ovaj korak se ne izvršava. Stoga je to vrijeme prikazano odvojeno od drugih koraka. Nadalje, korak 10 kojeg obavlja stručnjak, ne razma-

tra se u tablici 5.4. Eksperimenti su provedeni na računalu Intel Core i7-4790 3.60 GHz, 16 GB RAM, OS Windows 10 64-bit OS.

Tablica 5.4: Vrijeme izvođenja automatskog označavanja komponenti

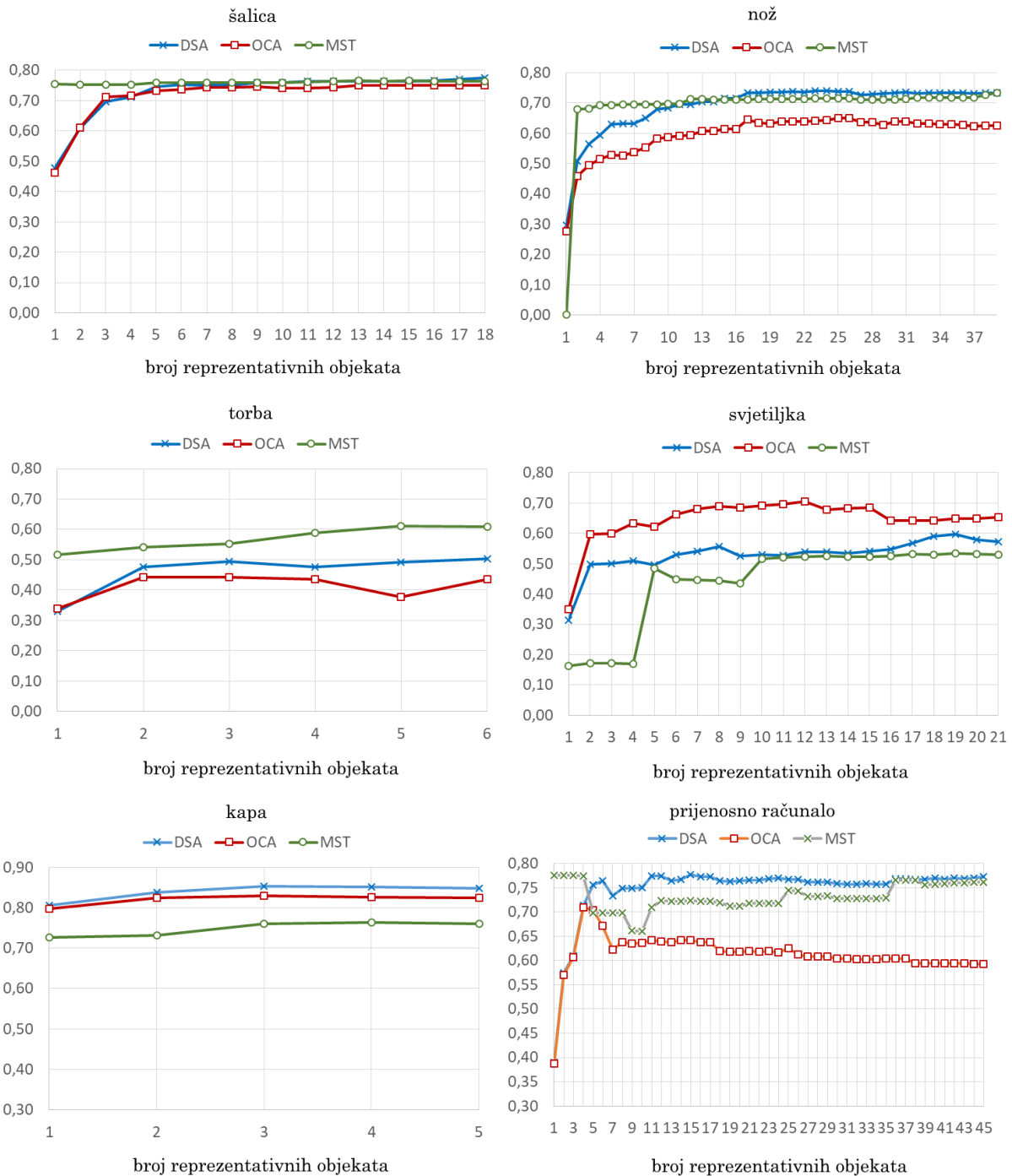
| Klasa  |        |                  | šalica | nož    | torba | svjetiljka | kapa | laptop |     |
|--|--------|------------------|--------|--------|-------|------------|------|--------|-----|
| ukupni broj objekata                               |        |                  | 184    | 391    | 68    | 210        | 53   | 451    |     |
| broj reprezentativnih objekata                     |        |                  | 18     | 39     | 6     | 21         | 5    | 45     |     |
| prosječno vrijeme izvođenja koraka po objektu [ms] | C++    | koraci 1-6       | 825    | 1112   | 1484  | 917        | 702  | 288    |     |
|  |        | koraci 7-8       | 19     | 70.6   | 36.2  | 62.4       | 3.8  | 14.6   |     |
| vrijeme izbora repr. objekata [ms]                 | MATLAB | DSA koraci 11-12 | 4.7    | 10.6   | 5.6   | 8.2        | 2.1  | 3.5    |     |
|  |        | OCA              | 109.7  | 667.5  | 11.9  | 199.1      | 4.6  | 383.4  |     |
|  |        | MST              | 52.2   | 106.7  | 14.9  | 45.4       | 5.3  | 72.5   |     |
| ukupno vrijeme po objektu [ms]                     |        |                  | min    | 849    | 1193  | 1526       | 988  | 708    | 306 |
|  |        |                  | max    | 954    | 1850  | 1535       | 1179 | 711    | 686 |
| vrijeme izbora repr. objekata [ms]                 | C++    | korak 9          | 59847  | 198611 | 12249 | 82930      | 4873 | 314175 |     |
|  | MATLAB | korak 9          | 10     | 26     | 2     | 4          | 1    | 35     |     |

Kako bi se istražilo kako pridruživanje komponenti ovisi o broju reprezentativnih objekata, provedena je dodatna serija eksperimenata, pri kojoj se broj reprezentativnih objekata mijenja od 1 do vrijednosti prikazane u tablici 5.2. Rezultati ove analize prikazani su na slici 5.6. Eksperimenti su provedeni za sve tri metode pridruživanja komponenti uz  $w_N = 1$ .

## 5.7.2 Diskusija rezultata

Uobičajeni postupak semantičke segmentacije sa standardnim strojnim učenjem ili dubokim neuronskim mrežama sastoji se od učenja, validacije i ispitivanja. Učenje se uglavnom odvija na velikom podskupu označenih objekata. Vrijeme koje se koristi pri označavanju i učenju rijetko je naznačeno u istraživanjima, ali zasigurno zahtijeva puno vremena i energije. Ova je eksperimentalna analiza istraživala točnost semantičke segmentacije bez iscrpnog učenja i označavanja. Prikazani rezultati dokazuju da je dovoljno samo nekoliko reprezentativnih modela da bi se ostvarila očekivana točnost. U slučaju jednostavnih objekata poput noževa, kapa i šalica, dovoljan je jedan ili dva označena modela kako bi se postigla IoU otprilike 0.7.

Na temelju predstavljene eksperimentalne analize može se zaključiti da uključivanje mjere sličnosti susjedstva u SCLM značajno poboljšava rezultate u slučaju torbi i svjetiljki, koje imaju složeniji oblik od šalica, noževa, kapa i prijenosnih računala. U slučaju šalica, noževa, kapa i prijenosnih računala, koji imaju jednostavniji oblik, informacija o topološkim odno-



Slika 5.6: Točnost pridruživanja komponenti u ovisnosti o broju reprezentativnih objekata i primijenjenoj metodi mjerena pomoći IoU

sima između komponenti ne poboljšava identifikaciju ciljne komponentu.

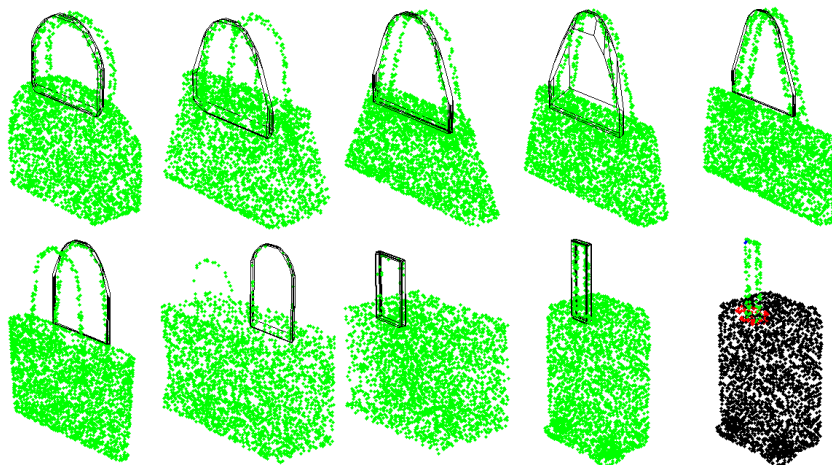
Za sve klase osim svjetiljki, preporučena metoda bila bi MST, zato što pruža relativno visok IoU za mali broj reprezentativnih objekata, a IoU značajno ne raste povećanjem njihova broja. U slučaju šalica, kapa i prijenosnih računala, dovoljan je jedan reprezentativni objekt kako bi se identificirala ciljna komponenta među preostalim objektima s  $\text{IoU} > 0.70$ . U slučaju noževa, pogrešan izbor prvog reprezentativnog objekta predloženom metodom rezultira niskim IoU. To se, doduše, značajno mijenja s drugim reprezentativnim objektom, kada se psotiže IoU približno 0.7.

Za očekivati je da MST metoda daje bolje rezultate od DSA u slučaju kada baza modela sadrži dovoljno guste uzorke razmatrane klase objekata. Smatra se da je baza modela gusta ako se za bilo koja dva objekta može pronaći sekvenca objekata pri kojoj dva susjedna objekta imaju dovoljno slične oblike da se omogući jednoznačno pridruživanje segmenata. Primjer takve sekvence dan je na slici 5.7. Ručka torbe, prikazane gore lijevo na slici 5.7, koja predstavlja reprezentativni objekt, ispravno je pridružena segmentu prikazanom zelenim točkama u donjem desnom kutu slike. U idealnom slučaju, procesiranje takve baze modela MST metodom rezultiralo bi ispravnim pridruživanjem komponenti na temelju samo jednog označenog reprezentativnog objekta. Međutim, kako bi baza bila gusta i pokrivala velik broj varijacija objekata, ona mora biti velika. Kako bi se ispitalo kako veličina baze objekata utječe na točnost pridruživanja metodom MST, postupak opisan koracima 1-12 proveden je za šest klasa s različitim brojem modela u bazi, koji varira od 25% do 100% ukupnog broja objekata prikazanog u tablici 5.2. Tijekom postupka, uključena je mjera sličnosti susjedstva,  $w_N = 1$ . Rezultati su prikazani u tablici 5.5.

Tablica 5.5: Točnost označavanja komponenti primjenom MST metode s različitim brojem objekata u bazi

| Postotak baze/ Klasa | šalica       | nož          | torba        | svjetiljka   | kapa         | laptop       |
|----------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 25%                  | 0.721        | <b>0.755</b> | 0.400        | 0.080        | 0.706        | <b>0.762</b> |
| 50%                  | 0.747        | 0.726        | 0.419        | 0.177        | 0.732        | 0.694        |
| 75%                  | 0.752        | 0.749        | 0.604        | 0.400        | <b>0.793</b> | 0.739        |
| 100%                 | <b>0.763</b> | 0.732        | <b>0.609</b> | <b>0.530</b> | 0.759        | 0.761        |

Primijenjena metoda segmentacije često stvara segmente koji predstavljaju mali dio ciljne komponente koji ne može biti povezan s odgovarajućim segmentom na temelju oblika. Nadalje, ako se baza modela sastoji od složenih objekata s velikom raznolikošću oblika, mnogi stvoreni segmenti nemaju odgovarajući par u bazi. U tim se slučajevima može dogoditi da je



Slika 5.7: Primjer guste baze sa sličnim uzastopnim objektima

najbliži susjed takvog segmenta u CAG-u predstavlja ciljnu komponentu, iako taj segment u stvarnosti ne predstavlja ciljnu komponentu. Tada DSA pogrešno pridružuje razmatrani segment sa ciljnom komponentom. Metoda OCA smanjuje ovaj efekt u slučajevima kada uz pogrešnog najbližeg susjeda nekom segmentu ispitivanog modela, reprezentativni model sadrži i ispravne najbliže susjede drugih segmenata ispitivanog modela koji predstavljaju ciljnu komponentu, a imaju veći SCLM od pogrešnog pridruživanja. Pretpostavimo da je ciljna komponenta predstavljena jednim segmentom  $C_{ik}^M$  reprezentativnog objekta te je ovaj segment ispravno pridružen segmentu  $C_j^Q$  ispitnog objekta. Nadalje, pretpostavimo da ovo pridruživanje ima najveću SCLM od svih pridruživanja segmenata ova dva objekta. U tom će slučaju OCA ispravno pridružiti razmatrani ispitni segment objekta sa ciljnom komponentom. Ako je  $C_{ik}^M$  također najbliži susjed drugog segmenta  $C_l^Q$  razmatranog ispitnog objekta koji ne predstavlja ciljnu komponentu, tada  $C_l^Q$  ne bi bio pogrešno pridružen komponenti  $C_{ik}^M$  jer OCA dopušta samo pridruživanje segmenata jedan-na-jedan. S druge strane, ako je ciljna komponenta predstavljena u ispitnom objektu s dva segmenta i u najsličnijem reprezentativnom objektu s jednim segmentom, tada jedan od ova dva ispitna segmenta ne bi bio pridružen ciljnoj komponenti metodom OCA. U slučaju svjetiljki, gdje je ciljna komponenta, žarulja, često predstavljena jednim malim segmentom te postoje velike varijacije oblika između instanci ove klase objekata, OCA daje najbolje rezultate. Za preostale klase, opisani nedostatak ove metode je dominantan.

Iako predložena metoda pruža zanimljive rezultate za mali broj reprezentativnih objekata, ne ostvaruje veliku točnost niti uz visok postotak označenih reprezentativnih objekata. Moguć razlog leži u tome što metoda koristi segmentaciju koja stvara segmente koji često

prelaze granicu ciljne komponente. To je primjetno u slučaju ručki šalica i torbi, gdje se konkavni segment, koji predstavlja unutrašnjost ručke, proteže na dio tijela objekta, što rezultira brojem *false positiva*, kao što je prikazano lijevim primjerom na slici 5.5, gdje su crvene točke pogrešno pridružene ručki.

Cilj predloženog postupka bio je smanjenje napora uloženog u označavanje i omogućavanje jednostavnog proširenja baze modela novim modelom u stvarnom vremenu. Vremena prikazana u tablici 5.4 upućuju na to da proširenje baze novim modelom, novi izračun CAG-a i utvrđivanje korespondencija između komponenti i segmenata traje između 0.3 i 1.9 sekunde, ovisno o složenosti strukture objekta, izabranoj metodi i trenutnom broju objekata u bazi. Stoga se može zaključiti da je predložena metoda prikladna za robotske radnje u stvarnom vremenu. Metoda za aktivno učenje označavanja velikog geometrijskog skupa podataka, (Yi et al., 2016) prikazuje prosječno vrijeme potrebno za predobradu i označavanje po obliku od približno jedne minute za bazu koja sadrži 400 objekata. Ovo vrijeme linearno raste s veličinom baze. Nadalje, analiza novog oblika, u metodi za konverziju geometrijskih oblika u hijerarhijski segmentirane označene dijelove, predloženoj u (Yi et al., 2017a), prosječno traje oko 25 sekundi. Usporedbom sa spomenutim metodama, koje također propagiraju labela s malog na veliki skup objekata, predložena metoda je značajno brža.

## 5.8 Zaključak

Cilj istraživanja opisanog u ovom poglavlju bio je razviti metodu semantičkog pridruživanja bez iscrpnog učenja i označavanja velike količine podataka, te procijeniti je li takva metoda prikladna za izvođenje u stvarnom vremenu.

Predložena metoda zasniva se na konstrukciji grafa pridruživanja komponenti, čiji čvorovi predstavljaju segmente objekata. Segmenti su aproksimirani konveksnim poliedrima. Veze CAG-a predstavljaju mjeru vjerojatnosti semantičkog pridruživanja između segmenata, koja razmatra poziciju, oblik i veličinu segmenata te sličnost njihova susjedstva. Konačno pridruživanje odvija se jednom od tri predložene metode bazirane na CAG-u. Evaluacija predloženog pristupa provedena je na skupu ShapeNet Part te polučuje zanimljive rezultate koji dokazuju da se s relativno malim brojem reprezentativnih označenih objekata, identifikacija ciljne komponente s točnošću mjerenoj u IoU većoj od 0.6 može postići u samo nekoliko sekundi.

S obzirom da je primjena neuronskih mreža u semantičkoj segmentaciji i klasifikaciji

objekata opravdana točnošću prezentiranom u relevantnim istraživanjima, učenje neuronskih mreža za općenitu svrhu segmentacije na komponente i njihovo pridruživanje zanimljiv je nastavak ovog istraživanja. Jedan od mogućih načina primjene neuronske mreže za problem predstavljen u ovom radu je njezino učenje na različitim skupovima podataka kako bi se naučio generički kriterij za moguće semantičko pridruživanje između komponenti modela. Nakon učenja, takva bi se mreža primijenila za detekciju i pridruživanje komponenti drugog neoznačenog skupa.

Ciljna primjena predložene metode je u robotici, gdje se može primjenjivati u kombinaciji s metodama za detekciju objekata i 3D rekonstrukciju. Zadatak tih metoda bio bi detektirati objekte određene klase na RGB-D slikama ili snimkama LiDAR-a te rekonstruirati cijeli 3D model. Predloženi postupak tada bi služio za pridruživanje komponenti rekonstruiranog objekta odgovarajućim komponentama modela objekta. Kako bi bila primjenjiva u stvarnim scenarijima, percepcija robota treba biti sposobna analizirati složene scene, na kojima je ciljni objekt djelomično vidljiv te se može nalaziti u različitim položajima. Proširenje predložene metode sa metodom koja detektira objekte zadane klase na složenim scenama i provodi njihovu 3D rekonstrukciju čini se kao logičan izbor za nastavak ovog istraživanja.

# 6

## Zaključak

Cilj ovog doktorskog rada bio je razvoj metoda računalnog vida s primjenom u robotskoj manipulaciji koje bi činile dijelove autonomnog robotskog sustava koji bi bio sposoban prvi puta viđenom objektu, na dubinskoj slici snimljenoj RGB-D kamerom, dodijeliti jednu od prethodno naučenih klasa te planirati i izvesti neku zadanu operaciju tim objektom. Da bi mogao isplanirati radnu operaciju nad ciljnim objektom, sustav bi klasificirani objekt trebao podijeliti na segmente te segmentima pridružiti semantičke oznake dijelova, tj. pronaći komponentu objekta predviđenu za neku robotsku radnju. Osim prepoznavanja same klase i dijela objekta, ovakav sustav bio bi sposoban odrediti i njihove položaje u koordinatnom sustavu kamere. Položaj dijela objekta namijenjen nekoj robotskoj radnji predstavljao bi određeni položaj vrha robotskog alata koji bi se u taj položaj doveo na temelju razvijenog algoritma za vizualno vođenje. Ovakav se sustav sastoji od robotske ruke niske cijene vodene RGB-D kamerom kao jedinim sensorom.

U ovom su doktorskome radu predstavljene razvijene metode koje su eksperimentalno ispitane pokusima hvatanja ili usporedbom točnosti s drugim metodama na relevantnim skupovima podataka.

Metoda vizualnog vođenja za SCARA robote koji koriste dubinsku kameru u boji kao jedini senzor ispitana je nizom pokusa. Pokusi pozicioniranja na temelju vizualnog vođenja pokazuju da je metoda prikladna za manipulaciju objektima za koje je dovoljna točnost od 7 mm. Ako se npr. razmotri rukovanje malim kućanskim predmetima, za kakve je metoda



prvenstveno i namijenjena, može se zaključiti da je točnost od 7 mm zadovoljavajuća, pod uvjetom da robot posjeduje hvataljku čiji je raspon za oko 2 cm veći od minimalno potrebnog za hvatanje predmeta. To je i dokazano pokusima hvatanja, gdje je 95 % pokusa hvatanja predmeta čije su dužine stranica manje od 10 cm bilo uspješno. Kako bi se, ipak, povećala preciznost pozicioniranja i uspješnost hvatanja, potrebno je razmotriti drugačiji dizajn robotske ruke, koji bi smanjivao udaljenost između dominantne ravnine i vrha alata. Naime, pogreška u pozicioniranju linearno se povećava s ovom udaljenošću. Osim toga, uporabom senzora koji pruža točniju registraciju RGB i dubinske slike, otklonila bi se pogreška uzrokovana pomakom ovih dviju slika. S obzirom da predložena metoda koristi obje informacije, o boji i o dubini, važno je da su slike točno registrirane.

Metoda klasifikacije objekata na dubinskim slikama zasnovana na poravnanju konveksne ljuske ispitana je na referentnom skupu podataka i uspoređena s relevantnim metodama. Metoda je uspješna preko 90 % za osam od deset ispitanih klasa, te je ukupna točnost ove metode na svim klasama bolja od metoda s kojima je uspoređena. Osim bolje točnosti, prednost ove metode je što osim klase, određuje i položaj predmeta u koordinatnom sustavu kamere. To je čini prikladnom za primjenu u robotskoj manipulaciji, gdje je važna pozicija i orijentacija objekta kako bi se ispravno pozicionirala i orijentirala hvataljka robota. Na ovu bi se metodu uspješno nadovezala metoda planiranja hvatanja koja bi transformirala željeno hvatanje definirano za referentni objekt iz baze na ostale objekte pripadnike iste klase, klasificirane ovom metodom na sceni. Ovakva bi metoda mogla dovesti hvataljku robota u blizinu prikladnog mjesta za hvatanje gdje bi se optimizacijom na malom lokalnom susjedstvu hvatanje prilagodilo dimenzijama i drugim svojstvima objekta na sceni. Problem ove metode je što zahtijeva da skup za učenje sadrži dovoljno sličan model ispitivanom objektu. Osim toga, klasifikacija je često pogrešna kada je vidljivi dio objekta na sceni sličan dijelu objekta pripadnika neke druge klase. To bi se moglo riješiti klasifikacijom na temelju više pogleda. Daljni nedostatak je linearno povećanje računске složenosti povećanjem broja objekata u bazi što bi se moglo riješiti primjenom algoritma indeksiranja. Konačno, ova metoda ne rješava problem okluzije i izdvajanja objekta iz pozadine.

Druga predložena metoda klasifikacije objekata na dubinskim slikama zasnovana na poli-edarskim modelima klasa oblika također je ispitana na istom referentnom skupu podataka i uspoređena s relevantnim metodama, kao i prethodno opisana metoda. Točnost ove metode je još veća, te postiže 100 % točnost na četiri klase objekata, a i ukupno daje najbolji rezultat

po svim klasama. Osim klasifikacije, ova je metoda ispitana u simulaciji pokusa hvatanja pri čemu se utvrdila mogućnost njezine primjene u robotici i planiranju adaptivnog hvatanja. Prednost ove metode je tolerancija relativno velikih razlika u obliku ispitivanog objekta i naj-sličnijeg objekta u skupu za učenje. Osim toga, parametri operacije hvatanja izravno se mogu izračunati iz estimiranih parametara ispitivanog objekta određenih u postupku klasifikacije, što joj također daje prednost pred drugim citiranim metodama. Nedostatak ove metode je što nije u potpunosti automatizirana, već se jedan njezin dio, definiranje metamodela, odvija ručno. To bi se moglo unaprijediti postupkom koji bi samostalno pronalazio optimalni metamodel pojedine klase. Nadalje, poželjno je zamijeniti heurističku metodu minimizacije troška generalnim optimizacijskim pristupom. Zanimljivo bi bilo vidjeti učinkovitost ove metode na složenim scenama, što je plan za daljnje istraživanje.

Konačno, predstavljena metoda određivanja korespondencija dijelova unutar klasa objekata zasnovana na poliedarskim modelima klasa oblika razvijena je u svrhu eliminacije ljudskog napora pri anotiranju velikih skupova podataka za potrebe semantičke segmentacije. Razvijena metoda pridružuje semantičke oznake dijelovima objekata na temelju malog broja anotiranih reprezentativnih objekata te omogućava brzo dodavanje novog objekta u bazu i njegovu semantičku segmentaciju. Metoda je ispitana na velikom skupu podataka 3D modela objekata pripadnika nekoliko klasa koje imaju jasno odvojivu komponentu prikladnu za robotsku radnju. Točnost automatske anotacije komponenti uspoređena je i s petnaest drugih referentnih metoda za semantičku segmentaciju koje su učene na velikom anotiranom skupu za učenje. Uspoređena je i brzina dodavanja novog modela u bazu i njegove anotacije, pri čemu je predložena metoda puno brža i prikladna za rad u stvarnom vremenu. Potencijalni nedostatak ove metode je potreba za gustom bazom modela pri kojoj dva susjedna objekta imaju dovoljno slične oblike da se omogući jednoznačno pridruživanje segmenata. Kako bi se ova metoda upotpunila i omogućila njezina primjena u robotskim radnim operacijama, treba je prilagoditi radu na RGB-D slikama, kakve snima RGB-D kamera. Stoga se kao nastavak ovog istraživanja, predlaže se primjena ove metode u robotici u kombinaciji s drugim postupcima detekcije objekata i 3D rekonstrukcije, gdje bi ova metoda pridruživala komponente rekonstruiranog objekta na sceni korespondentnim komponentama modela objekta.

# Literatura

- ArUco: a minimal library for augmented reality applications based on OpenCV*, Pristup: 2020-06-01. URL <https://www.uco.es/investiga/grupos/ava/node/26>.
- Orbbec Astra*, Pristup: 2020-06-01. URL <https://orbbec3d.com/>.
- International Federation of Robotics*, Pristup: 2020-06-01. URL <https://ifr.org/>.
- N. Abdo, H. Kretschmar, L. Spinello i C. Stachniss. Learning manipulation actions from a few demonstrations. *stranica* 1268–1275, May 2013.
- A. Aldoma, N. Blodow, D. Gossow, S. Gedildia, R. B. Rusu, M. Vincze i G. Bradski. Cad-model recognition and 6dof pose estimation using 3d cues. *3rd IEEE Workshop on 3D Representation and Recognition*, 2011.
- A. Aldoma, F. Tombari, L. Di Stefano i M. Vincze. A global hypotheses verification method for 3d object recognition. *European Conference on Computer Vision*, stranica 511–524. 2012.
- A. Aldoma, F. Tombari, L. D. Stefano i M. Vincze. A global hypothesis verification framework for 3d object recognition in clutter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1383–1396, July 2016.
- U. Asif, M. Bennamoun i F. Sohel. Efficient rgb-d object categorization using cascaded ensembles of randomized decision trees. *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, stranice 1295–1302. IEEE, 2015.
- M. Atzmon, H. Maron i Y. Lipman. Point convolutional neural networks by extension operators. *ACM Transactions on Graphics*, 37(4):1–12, Jul 2018.
- H. Barrow, J. Tenenbaum, R. Bolles i H. Wolf. Parametric correspondence and chamfer matching: two new techniques for image matching. *5th International Joint Conference on Artificial Intelligence*, stranice 659–663, 1977.
- Y. Ben-Shabat, M. Lindenbaum i A. Fischer. 3d point cloud classification and segmentation using 3d modified fisher vector representation for convolutional neural networks. *arXiv:1711.08241 [cs]*, Nov 2017. arXiv: 1711.08241.

- M. Blum, J. T. Springenberg, J. Wülfing i M. Riedmiller. A learned feature descriptor for object recognition in rgb-d data. *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, stranice 1298–1303. IEEE, 2012.
- L. Bo, X. Ren i D. Fox. Hierarchical matching pursuit for image classification: Architecture and fast algorithms. J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira i K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, stranice 2115–2123. Curran Associates, Inc., 2011.
- L. Bo, X. Ren i D. Fox. Unsupervised feature learning for rgb-d based object recognition. *International Symposium on Experimental Robotics*, 2012.
- R. A. Bobby i S. K. Saha. Single image based camera calibration and pose estimation of the end-effector of a robot. *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, stranica 2435–2440. 2016.
- C. Borst, M. Fischer i G. Hirzinger. Calculating hand configurations for precision and pinch grasps. *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, svezak 2, stranica 1553–1559. 2002.
- A. Brock, T. Lim, J. M. Ritchie i N. Weston. Generative and discriminative voxel modeling with convolutional neural networks. *CoRR*, abs/1608.04236, 2016.
- A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi i F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.
- S. Choi, Q. Zhou, S. Miller i V. Koltun. A large dataset of object scans. *CoRR*, abs/1602.02481, 2016.
- P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli i G. Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. V. Scarano, R. D. Chiara i U. Erra, editors, *Eurographics Italian Chapter Conference*. 2008.
- A. Coates i A. Y. Ng. The importance of encoding versus training with sparse coding and vector quantization. *Proceedings of the 28th international conference on machine learning (ICML-11)*, stranice 921–928, 2011.
- J. J. Craig. *Introduction to Robotics - Mechanics and Control*. Pearson Prentice Hall, 2005.
- R. Cupec. Detection of local reference frames and geometric primitives based on planar patches, Technical Report ARP3D.TR4.11-B. Technical report.
- R. Cupec i P. Đurović. Volume net: Flexible model for shape classes. *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, stranica 248–255. Dec 2018.

- R. Cupec, E. K. Nyarko i D. Filko. Global localization based on 3d planar surface segments. *stranice 13–36*, 2013.
- R. Cupec, D. Filko i E. K. Nyarko. Segmentation of depth images into objects based on local and global convexity. *2017 European Conference on Mobile Robots (ECMR)*. Sep 2017.
- R. Cupec, I. Vidović, D. Filko i P. Đurović. Object Recognition Based on Convex Hull Alignment. *Pattern Recognition*, 2020.
- A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser i M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *CoRR*, abs/1702.04405, 2017.
- R. Detry, C. H. Ek, M. Madry i D. Kragic. Learning a dictionary of prototypical grasp-predicting parts from grasping experience. *2013 IEEE International Conference on Robotics and Automation*, stranica 601–608. May 2013.
- M. Filipović, P. Đurović i R. Cupec. Experimental evaluation of point cloud classification using the pointnet neural network. *Proceedings of the 10th International Joint Conference on Computational Intelligence (IJCCI 2018)*, stranice 47–54, Sep 2018.
- M. Firman. RGBD Datasets: Past, Present and Future. *CVPR Workshop on Large Scale 3D Data: Acquisition, Modelling and Analysis*, 2016.
- M. A. Fischler i R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24/6, 1981.
- R. M. Freeman, S. J. Julier i A. J. Steed. A method for predicting marker tracking error. *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, stranica 157–160. 2007.
- A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez i J. Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv:1704.06857 [cs]*, Apr 2017. arXiv: 1704.06857.
- B. Graham, M. Engelcke i L. v. d. Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, stranica 9224–9232. Jun 2018.
- T. Groueix, M. Fisher, V. G. Kim, B. C. Russell i M. Aubry. Atlasnet: A papier-mache approach to learning 3d surface generation. *arXiv:1802.05384 [cs]*, Jul 2018. arXiv: 1802.05384.
- Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan i N. M. Kwok. A comprehensive performance evaluation of 3d local feature descriptors. *International Journal of Computer Vision*, 116(1):66–89, Jan 2016.

- S. Gupta, J. Hoffman i J. Malik. Cross modal distillation for supervision transfer. *CVPR*, 2016.
- J. Heller, D. Henrion i T. Pajdla. Hand-eye and robot-world calibration by global polynomial optimization. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, stranica 3157–3164, May 2014.
- A. C. Hernández, C. Gómez, J. Crespo i R. Barber. Object classification in natural environments for mobile robot navigation. *Autonomous Robot Systems and Competitions (ICARSC), 2016 International Conference on*, stranice 217–222. IEEE, 2016.
- S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua i V. Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):876–888, May 2012.
- D. Holz i S. Behnke. Fast range image segmentation and smoothing using approximate surface reconstruction and region growing. *Proc. International Conference on Intelligent Autonomous Systems (IAS)*, 2012.
- Q. Huang, H. Wang i V. Koltun. Single-view reconstruction via joint analysis of image and shape collections. *ACM Transaction on Graphics*, 34/4, 2015.
- Q. Huang, W. Wang i U. Neumann. Recurrent slice networks for 3d segmentation of point clouds. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, stranica 2626–2635. Jun 2018.
- M. Jaderberg, K. Simonyan, A. Zisserman et al. Spatial transformer networks. *Advances in Neural Information Processing Systems*, stranice 2017–2025, 2015.
- A. Kaehler i B. G. *Learning OpenCV 3*. O'Reilly Media, 2016.
- E. Kalogerakis, A. Hertzmann i K. Singh. Learning 3d mesh segmentation and labeling. *ACM Transactions on Graphics*, 29(4):1, Jul 2010.
- A. Khan, L. Sun, G. Aragon-Camarasa i J. P. Siebert. Interactive perception based on gaussian process classification for house-hold objects recognition & sorting. *Robotics and Biomimetics (ROBIO), 2016 IEEE International Conference on*, stranice 1087–1092. IEEE, 2016.
- H.-S. Kim, J.-K. Min i J.-B. Song. Multi-dof counterbalance mechanism for low-cost, safe and easy-usable robot arm. *Ubiquitous Robots and Ambient Intelligence (URAI), 2014 11th International Conference on*, stranica 185–188. 2014.
- R. Klovov i V. Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. *2017 IEEE International Conference on Computer Vision (ICCV)*, stranica 863–872. Oct 2017.

- D. Kragic, A. T. Miller i P. K. Allen. Real-time tracking meets online grasp planning. *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, svezak 3, stranica 2460–2465. 2001.
- S. Kucuk i Z. Bingul. Robot Kinematics: Forward and Inverse Kinematics. Technical report, 2006.
- K. Lai, L. Bo, X. Ren i D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, stranice 1817–1824. IEEE, 2011.
- B. P. Larouche i Z. H. Zhu. Autonomous robotic capture of non-cooperative target using visual servoing and motion predictive control. *Autonomous Robots*, 37(2):157–167, Aug 2014.
- J. Li, B. M. Chen i G. H. Lee. So-net: Self-organizing network for point cloud analysis. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, stranica 9397–9406. Jun 2018a.
- Y. Li, R. Bu, M. Sun, W. Wu, X. Di i B. Chen. Pointcnn: Convolution on  $\mathcal{X}$ -transformed points. *arXiv:1801.07791 [cs]*, Jan 2018b. arXiv: 1801.07791.
- V. Lippiello, B. Siciliano i L. Villani. Eye-in-hand/eye-to-hand multi-camera visual servoing. *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, stranica 5354–5359. 2005.
- E. Lopez-Damian, D. Sidobre i R. Alami. Grasp planning for non-convex objects. *International symposium on robotics*, svezak 36, stranica 167. 2005.
- Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort i M. Beetz. General 3d modelling of novel objects from a single view. *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, stranica 3700–3705. 2010.
- L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin i A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- A. Miller i P. Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, 11(4):110–122, 2004.
- J. Miseikis, K. Glette, O. J. Elle i J. Torresen. Automatic calibration of a robot manipulator and multi 3d camera system. *arXiv preprint arXiv:1601.01566*, 2016.
- C. A. Mueller, K. Pathak i A. Birk. Object shape categorization in rgb-d images using hierarchical graph constellation models based on unsupervisedly learned shape parts described

- by a set of shape specificity levels. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, stranica 3053–3060. Sep 2014.
- L. Nan, K. Xie i S. A. Approach for cluttered indoor scene understanding. *ACM SIGGRAPH*, 2012.
- E. Nyarko, I. Vidović, K. Radočaj i R. Cupec. A nearest neighbor approach for fruit recognition in RGB-D images based on detection of convex surfaces. *Expert Syst. Appl.*, 114: 454–466, 2018.
- R. Osada, T. Funkhouser, B. Chazelle i D. Dobkin. Shape distributions. *ACM Transactions on Graphics (TOG)*, 21(4):807–832, 2002.
- C. Papazov i D. Burschka. An efficient ransac for 3d object recognition in noisy and occluded scenes. *Asian Conference on Computer Vision*, stranica 135–148. 2010.
- P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou i S. Schaal. Skill learning and task outcome prediction for manipulation. stranica 3828–3834, May 2011.
- T. Patten, M. Zillich, R. Fitch, M. Vincze i S. Sukkarieh. Viewpoint evaluation for online 3-d active object classification. *IEEE Robotics and Automation Letters*, 1(1):73–81, 2016.
- C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan i L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, stranice 5648–5656, 2016.
- C. R. Qi, H. Su, K. Mo i L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017a.
- C. R. Qi, L. Yi, H. Su i L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv:1706.02413 [cs]*, Jun 2017b. arXiv: 1706.02413.
- Y. Qiao, Q.-s. Liu, L. Peng i G.-p. Liu. A new method of self-calibration of hand-eye systems based on active vision. *IFAC Proceedings Volumes*, 47(3):9347–9352, 2014.
- A. Ricci. A constructive geometry for computer graphics. *The Computer Journal*, 16/2: 157–160, 1973.
- J. Ritter. An efficient bounding sphere. *A.S. Glassner (Ed.), Graphics Gems, Academic Press Professional, Inc.*, stranice 301–303, 1990.
- R. B. Rusu i S. Cousins. 3D is here: Point Cloud Library (PCL). *IEEE International Conference on Robotics and Automation (ICRA)*, May 9-13 2011.



- R. B. Rusu, N. Blodow i M. Beetz. Fast point feature histograms (fpfh) for 3d registration. *2009 IEEE International Conference on Robotics and Automation*, stranice 3212–3217, May 2009.
- R. B. Rusu, G. Bradski, R. Thibaux i J. Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. *International Conference on Intelligent Robots and Systems (IROS)*, stranice 2155–21628, 2010.
- S. Salti, F. Tombari i L. D. Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125(Supplement C):251–264, 2014.
- Y. Shen, C. Feng, Y. Yang i D. Tian. Mining point cloud local structures by kernel correlation and graph pooling. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, stranica 4548–4557. Jun 2018.
- T. Simeon, J.-P. Laumond i F. Lamiroux. Move3d: A generic platform for motion planning. *In Proc. 4th International Symposium on Assembly and Task Planning (ISATP'2001)*. 2001.
- S. Song, S. P. Lichtenberg i J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. *Proceedings of the IEEE conference on computer vision and pattern recognition*, stranice 567–576, 2015.
- J. Speth, A. Morales i P. J. Sanz. Vision-based grasp planning of 3d objects by extending 2d contour based algorithms. *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, stranica 2240–2245. 2008.
- K. H. Strobl i G. Hirzinger. Optimal hand-eye calibration. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, stranica 4647–4653. 2006.
- X. Sun. Semantic annotation of 3d architecture models based on the geometric structure characteristics. *2018 26th International Conference on Geoinformatics*, stranica 1–6. Jun 2018.
- L. Twardon i H. Ritter. Interaction skills for a coat-check robot: Identifying and handling the boundary components of clothes. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, stranica 3682–3688. May 2015.
- P. Đurović, R. Grbić i R. Cupec. Visual servoing for low-cost scara robots using an rgb-d camera as the only sensor. *Automatika*, 58(4):495–505, Oct 2017a.
- P. Đurović, R. Grbić, R. Cupec i D. Filko. Low cost robot arm with visual guided positioning. *Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, *2017 40th International Convention on*, stranica 1120–1125. 2017b.

- P. Đurović, M. Filipović i R. Cupec. Alignment of similar shapes based on their convex hulls for 3d object classification. *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, stranica 1586–1593. Dec 2018.
- P. Đurović, I. Vidović i R. Cupec. Semantic component association within object classes based on convex polyhedrons. *Applied Sciences*, 10(8):2641, Apr 2020.
- S. van Delden i F. Hardy. Robotic eye-in-hand calibration in an uncalibrated environment.
- N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu i Y.-G. Jiang. *Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images*, svezak 11215 of *Lecture Notes in Computer Science*, stranica 55–71. Springer International Publishing, 2018a.
- W. Wang, R. Yu, Q. Huang i U. Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. *arXiv:1711.08588 [cs]*, May 2019. arXiv: 1711.08588.
- Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein i J. M. Solomon. Dynamic graph cnn for learning on point clouds. *arXiv:1801.07829 [cs]*, Jan 2018b. arXiv: 1801.07829.
- W. Wohlkinger i M. Vincze. Ensemble of shape functions for 3d object classification. *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, stranice 2987–2992. IEEE, 2011a.
- W. Wohlkinger i M. Vincze. Shape distributions on voxel surfaces for 3d object classification from depth images. *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*, stranice 115–120. IEEE, 2011b.
- W. Wohlkinger, A. Aldoma Buchaca, R. Rusu i M. Vincze. 3DNet: Large-Scale Object Class Recognition from CAD Models. *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.
- H. Xu, Y. Wang, W. Chen i J. Lu. A self-calibration approach to hand-eye relation using a single point. *Information and Automation, 2008. ICIA 2008. International Conference on*, stranica 413–418. 2008.
- Y. Xu, T. Fan, M. Xu, L. Zeng i Y. Qiao. *SpiderCNN: Deep Learning on Point Sets with Parameterized Convolutional Filters*, svezak 11212, stranica 90–105. Springer International Publishing, 2018. Series Title: Lecture Notes in Computer Science.
- Z. Xue i R. Dillmann. Efficient grasp planning with reachability analysis. *International Journal of Humanoid Robotics*, 08(04):761–775, Dec 2011.
- L. Yi, L. Guibas, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang i A. Sheffer. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics*, 35(6):1–12, Nov 2016.

- 
- L. Yi, L. Guibas, A. Hertzmann, V. G. Kim, H. Su i E. Yumer. Learning hierarchical shape segmentation and labeling from online repositories. *ACM Transactions on Graphics*, 36 (4):1–12, Jul 2017a. arXiv: 1705.01661.
- L. Yi, L. Shao, M. Savva, H. Huang, Y. Zhou, Q. Wang, B. Graham, M. Engelcke, R. Klokov, V. Lempitsky i et al. Large-scale 3d shape reconstruction and segmentation from shapenet core55. *arXiv:1710.06104 [cs]*, Oct 2017b. arXiv: 1710.06104.
- L. Yi, H. Su, X. Guo i L. Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, stranica 6584–6592. Jul 2017c.
- L. Yi, W. Zhao, H. Wang, M. Sung i L. Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. *arXiv:1812.03320 [cs]*, Dec 2018. arXiv: 1812.03320.
- K. Yu, Y. Lin i J. Lafferty. Learning image representations from the pixel level via hierarchical sparse coding. *CVPR 2011*, stranice 1713–1720, June 2011.

## Sažetak

Motivacija za istraživanje predstavljeno u ovoj disertaciji jest razvoj inteligentnih robota koji djeluju u nestrukturiranim unutarnjim prostorima, poput kućanstava, bolnica ili škola. Pri tome, takvi roboti rukuju predmetima koje prethodno nisu vidjeli. Za takvo autonomno kretanje i manipulaciju, robotskom je sustavu potreban odgovarajući percepcijski senzor. Uporaba RGB-D kamere u ulozi senzora pruža trodimenzionalnu informaciju o okolini robotskog manipulatora. Iz slike snimljene RGB-D kamerom mogu se dobiti informacije o obliku i veličini predmeta na sceni, korisne u algoritmima za prepoznavanje i klasifikaciju prvi puta viđenih predmeta, kao i u algoritmima semantičke segmentacije. Segmentacija objekta na semantičke dijelove korisna je za određivanje dijela objekta koji je zbog svoje svrhe ili geometrijskih svojstava, prikladan za različite radne operacije. RGB-D kamera daje i informacije o položaju predmeta i njegovih dijelova na sceni, koje se, zajedno s vizualnim obilježjima robotske ruke, mogu upotrijebiti u vizualnom vođenju i planiranju radnih operacija.

Cilj istraživanja opisanog u ovoj disertaciji je razvoj metoda za klasifikaciju objekata na sceni snimljenoj RGB-D kamerom, metode za semantičku segmentaciju objekata i metode vizualnog vođenja. Namjena ovih metoda je integracija u autonomni robotski sustav s rukom i kamerom koji prepoznaje i manipulira prvi puta viđenim objektima.

Istraživanje opisano u ovoj disertaciji nastoji upotpuniti nedostatke postojećih metoda. Najčešći je nedostatak metoda klasifikacije objekata što ne estimiraju položaj objekata potreban za primjene u robotskoj manipulaciji. Nadalje, problem segmentacije na dijelove i određivanje korespondencija između dijelova objekata iste klase u svrhu robotske manipulacije je relativno neistražen, a izvođenje automatiziranih postupaka semantičkog označavanja velikih skupova objekata zahtijeva ljudske i vremenske resurse. Naposljetku, nedostatak su i dugotrajne metode umjeravanja robotske ruke i kamere koje je teško prilagoditi za automatizirani rad u stvarnom vremenu kao i nedovoljan broj prikladnih metoda za vizualno

vođenje cjenovno dostupnih robota. Nabrojani nedostaci dovode do potrebe za razvojem novih metoda.

Tijekom istraživanja razvijene su četiri nove metode:

- 1. Metoda vizualnog vođenja za SCARA robote koji koriste dubinsku kameru u boji kao jedini senzor.** Metoda koristi marker pričvršćen blizu alata robota, koji omogućuje određivanje trenutne pozicije alata pomoću RGB-D kamere te informaciju o dominantnoj ravnini u radnom prostoru robota dobivenu iz RGB-D slike. Umjeravanje sustava robot-kamera se izvodi jednim pokretom robotske ruke.
- 2. Metoda klasifikacije objekata na dubinskim slikama zasnovana na poravnanju konveksne ljuske.** Za dani objekt prikazan na RGB-D slici, metoda određuje najbližijeg pripadnika određene klase objekata. Metoda omogućuje određivanje položaja objekta na sceni u odnosu na referentni objekt dane klase. Učenje se provodi na bazi 3D modela objekata razvrstanih u nekoliko klasa.
- 3. Metoda klasifikacije objekata na dubinskim slikama zasnovana na poliedarskim modelima klasa oblika.** Za dani objekt, metoda određuje parametre najbližijeg pripadnika određene klase oblika. Učenje se provodi na bazi 3D modela objekata razvrstanih u nekoliko klasa. Metoda omogućuje određivanje položaja objekta te korespondencija između dijelova površine objekata iste klase.
- 4. Metoda određivanja korespondencija dijelova unutar klasa objekata zasnovana na poliedarskim modelima klasa oblika.** Metoda pridružuje segmentima razmatranih objekata semantičku oznaku na temelju malog broja anotiranih reprezentativnih objekata i omogućuje proširenje baze objekata u stvarnom vremenu.

Razvijena metoda vizualnog vođenja eksperimentalno je ispitana nizom pokusa navođenja robota iznad željene pozicije i hvatanjem jednostavnih predmeta. Uspješnost metoda klasifikacije objekata na dubinskim slikama i određivanja dijelova na referentnim skupovima 3D objekata uspoređene su s drugim referentnim metodama te je u disertaciji opisana analiza rezultata usporedbe.

**Ključne riječi:** vizualno vođenje, umjeravanje robotske ruke i kamere, robotska manipulacija, klasifikacija objekata, korespondencije dijelova objekata, 3D senzori

# Abstract

## **Object classification and intra-class part correspondence determination in depth images for robot manipulation**

The motivation for the research presented in this dissertation is development of intelligent robots capable of operating in unstructured environment such as houses, hospitals and schools. Such robots are expected to manipulate with previously unseen objects. In order to facilitate autonomous locomotion and manipulation, a robot system requires an appropriate perception sensor. RGB-D camera as a sensor provides 3D information about the robot's environment. The information about the shape and size of objects on a scene, obtained by the RGB-D camera, is useful in object recognition and classification algorithms, as well as in algorithms for semantic segmentation. Segmentation of objects into semantic parts is required in order to identify the object parts, relevant for different robot operations. Furthermore, RGB-D cameras provide information about the object pose and the pose of its parts, which is, along with the robot arm visual features, utilized in visual servoing and robot operation planning.

The goal of the research presented in this dissertation, was to develop methods for object classification in RGB-D scenes, semantic object segmentation and visual servoing. The purpose of these methods is integration in an autonomous robot system with an arm and a camera, capable of recognizing and manipulating with previously unseen objects.

The research presented in this thesis attempts to overcome the existing methods limitations. A common drawback of the existing object classification methods is not estimating the object's pose required for application in robot manipulation. Furthermore, there is a relatively small number of research regarding semantic part segmentation and part correspondence determination for robot manipulation tasks, while the part annotation of large object datasets is labour-intensive. Finally, the lack of hand-eye calibration methods suitable for real time recalibration as well as visual servoing methods suitable for low cost robots

are common flaws of the state-of-the-art approaches. The mentioned drawbacks encourage the development of novel approaches.

During the research, four novel methods are developed:

- 1. A visual servoing method for SCARA robots using an RGB-D camera as the only sensor.** This method utilizes a marker, placed at the robot end effector, which determines the current robot tool position, and dominant plane information obtained by the RGB-D camera. Hand-eye calibration is performed by a single robot arm movement.
- 2. A method for object classification in depth images based on convex hull alignment.** For a given object in an RGB-D image, this method determines the most similar object belonging to a particular class. The method estimates the object position with respect to the referent object from the same class. The learning is based on a 3D model database, sorted into classes.
- 3. A method for object classification in depth images based on polyhedral shape class models.** For a given object, this method determines the parameters of the most similar object belonging to the particular class. Analogously to the previous proposed object classification method, the learning is based on a 3D model database, sorted into classes. This method estimates the pose of the object and the correspondences between intra-class object surfaces.
- 4. A method for determination of part correspondences within object classes based on polyhedral shape class models.** This method assigns the semantic labels to the considered objects' segments, based on a small number of annotated representative objects and enables real time database expansion.

The developed visual servoing approach is experimentally tested by performing positioning and simple objects' grasping experiments. The accuracy of the proposed classification methods are experimentally evaluated and compared to several state-of-the-art approaches using a publicly available benchmark dataset. The accuracy of the part association method is evaluated using a set of 3D object models of several classes.

**Keywords:** visual servoing, hand-eye calibration, robotic manipulation, object classification, object part correspondences, 3D sensors

# Životopis

Petra Đurović rođena je Zagrebu, 24.12.1991. godine. Diplomirala je na Elektrotehničkom fakultetu u Osijeku 2015. godine i stekla akademski naziv magistre inženjerke računarstva. Iduće, 2016., godine upisala je poslijediplomski doktorski studij Elektrotehnike, smjer Komunikacije i informatika na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek. Zaposlena je na istom fakultetu kao asistentica i doktorandica na projektu Hrvatske zaklade za znanost, *Napredna 3D percepcija za mobilne robotske manipulatore*, pod mentorstvom prof.dr.sc. Roberta Cupeca.

Područja istraživanja su robotika i robotski vid. Do obrane doktorskog rada objavila je kao autor i koautor jedanaest radova, od toga dva CC rada u časopisima A kategorije u prvom i drugom kvartilu, jedan rad u časopisu A kategorije u četvrtom kvartilu te osam radova prezentiranih i objavljenih u zbornicima međunarodnih znanstvenih konferencija, od kojih ističe *ROBIO 2018* u Kuala Lumpuru, Malezija i *IJCCI 2018* u Sevilji, Španjolska.

Sudjeluje u raznim manifestacijama popularizacije znanosti i mentorira studente na projektu u suradnji s tvrtkom Systec, *Humans Detected by Robots*. Osim na Fakultetu, zaposlena je i u EDukOS - Centar znanja, gdje podučava fiziku.



# Popis slika

|      |  |    |
|------|--|----|
| 2.1  | Robotska ruka u trenutnom (crveno) i odredišnom (zeleno) položaju . . . . .  | 13 |
| 2.2  | Detekcija potporne ravnine označene zelenom bojom . . . . .  | 16 |
| 2.3  | Koordinatni sustav markera i orijentacija alata . . . . .  | 18 |
| 2.4  | Primjeri graničnih okvira detektiranih objekata, označeni zelenim linijama . . . . .   | 19 |
| 2.5  | Koordinatni sustav robota i alata u poziciji ${}^C p_{M'}$ . . . . .   | 20 |
| 2.6  | VICRA - robotska ruka u SCARA konfiguraciji . . . . .  | 21 |
| 2.7  | Vizualno navođen robotski sustav u SCARA konfiguraciji . . . . .   | 22 |
| 2.8  | Udaljenost $d$ između centra markera (crvena točka) i točke lasera (zeleno točka) . . . . .  | 23 |
| 2.9  | Normalizirani kumulativni histogram pogreške pozicioniranja alata . . . . .  | 24 |
| 2.10 | Inicijalni položaj hvataljke . . . . .   | 24 |
| 3.1  | Dva slična oblika (a) i njihovo poravnanje (b). . . . .  | 27 |
| 3.2  | 3D modeli (gornji redak) i odgovarajući CTI-evi (donji redak). Prva tri CTI-a predstavljaju cijeli oblik prva tri objekta, dok je četvrti objekt, čaša, predstavljena s tri CTI-a za svaki konveksni segment. . . . .  | 32 |
| 3.3  | Poliedar (sivo) i CTI (zeleno) generiran na osnovi tog poliedra. Stranica $i$ vidljiva je kameri, dok stranica $j$ nije. . . . .   | 34 |
| 3.4  | 2D prikaz površine poliedra. Susjedstvo $\mathcal{N}_{\pi/4}(F)$ stranice $F$ označeno je zelenim linijama. Tangencijalna ravnina između stranica $F' \in \mathcal{N}_{\pi/4}(F)$ i $F'' \notin \mathcal{N}_{\pi/4}(F)$ predstavljena je crvenom isprekidanom linijom. . . . . | 35 |
| 3.5  | Slika u boji (lijevo) i pripadajuća dubinska slika (desno). Svijetlo plavom bojom označeni su dijelovi koji nisu vidljivi iz trenutnog pogleda kamere. . . . .   | 38 |
| 3.6  | Primjeri ispravno poravnanih modela s razmatranim objektima . . . . .  | 42 |
| 3.7  | Matrica konfuzije nakon poravnanja objekata (a) i nakon cjelokupne klasifikacije objekata (b) . . . . .  | 43 |

|     |   |    |
|-----|---|----|
| 4.1 | Primjer VolumeNet strukture i nekoliko instanci 2D oblika klase predstavljene tom strukturom . . . . .  | 50 |
| 4.2 | Skup 3D modela, od kojih svaki pripada klasi oblika (gornji redak) i njihovi pripadajući 3D volumeni modelirani VN-om (donji redak). . . . .  | 50 |
| 4.3 | Planiranje hvatanja na temelju VN modela . . . . .  | 58 |
| 4.4 | Hvatanje definirano na temelju odabranih stranica VN modela prilagođeno objektima na sceni. Plave točkice predstavljaju instancu VN-a projiciranu na RGB-D sliku. . . . .   | 58 |
| 4.5 | Toroidalna površina definirana presjekom 7 komplementa 3D konusa. Jedan od tih konusa označen je crvenom bojom. . . . .   | 61 |
| 4.6 | Dvodimenzionalni prikaz određivanja relevantnih stranica. Normale $n_1$ i $n_2$ stranica označenih isprekidanim tamno crvenim linijama definiraju kružni isječak. Stranica čija se normala $a_i$ nalazi unutar kružnog isječka smatra se relevantnom. . . . .   | 64 |
| 4.7 | Metamodeli (gornji redak) i modeli (donji redak) dviju klasa oblika: boce (lijevo) i zdjele (desno). Boce su predstavljene unijom dviju koveksnih komponenti, dok su zdjele predstavljene presjekom jedne konveksne i jedne konkavne komponente. Cjelokupni objekti predstavljeni su isprekidanom crtom, relevantne stranice predstavljene su punom crnom linijom, a irelevantne stranice crvenom isprekidanom linijom. . . . . | 65 |
| 5.1 | Primjer konveksne (lijevo) i konkavne (desno) površine sa strelicama koje predstavljaju normale . . . . .   | 75 |
| 5.2 | Prikaz ručke šalice jednom konveksnom i jednom konkavnom plohom . . . . .   | 75 |
| 5.3 | Prikaz objekata pomoću nekoliko CTI-ja . . . . .  | 77 |
| 5.4 | Deskriptor topoloških odnosa (TRED). Elementi CTI deskriptora $d_{xm}$ koji su označeni suprotno usmjerenim strelicama od njima odgovarajućih vektora $a_m$ imaju negativnu vrijednost. . . . .   | 79 |
| 5.5 | Primjeri rezultata . . . . .  | 87 |
| 5.6 | Točnost pridruživanja komponenti u ovisnosti o broju reprezentativnih objekata i primijenjenoj metodi mjerena pomoći IoU . . . . .  | 90 |
| 5.7 | Primjer guste baze sa sličnim uzastopnim objektima . . . . .  | 92 |

# Popis tablica

|     |   |    |
|-----|---|----|
| 2.1 | Eksperimentalni rezultati vizualnog vođenja . . . . .   | 23 |
| 2.2 | Rezultati pokusa hvatanja . . . . .   | 25 |
| 3.1 | Parametri klasifikacije objekata . . . . .  | 42 |
| 3.2 | Rezultati klasifikacije objekata [%] . . . . .  | 43 |
| 4.1 | Parametri modela klasa oblika . . . . .   | 66 |
| 4.2 | Rezultati klasifikacije objekata [%] . . . . .  | 67 |
| 5.1 | Vrijednosti parametara . . . . .  | 84 |
| 5.2 | Točnost automatskog označavanja komponenti [%] . . . . .  | 87 |
| 5.3 | Usporedba točnosti s relevantnim metodama [%] . . . . .   | 88 |
| 5.4 | Vrijeme izvođenja automatskog označavanja komponenti . . . . .                                    | 89 |
| 5.5 | Točnost označavanja komponenti primjenom MST metode s različitim brojem objekata u bazi . . . . . | 91 |