Video SLAM tehnologija u podmorju

Vičević, Dominik

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Preddiplomski sveučilišni studij računarstva

Završni rad

Video SLAM tehnologija u podmorju

Rijeka, rujan 2022.

Dominik Vičević
0069088197

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Preddiplomski sveučilišni studij računarstva

Završni rad

Video SLAM tehnologija u podmorju

Mentor: Izv. prof. dr. sc. Jonatan Lerga

Rijeka, rujan 2022.

Dominik Vičević
0069088197

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 21. ožujka 2022.

Zavod: **Zavod za računarstvo**
Predmet: **Digitalna logika**
Grana: **2.09.03 obradba informacija**

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Dominik Vičević (0069088197)**
Studij: Preddiplomski sveučilišni studij računarstva

Zadatak: **Video SLAM tehnologija u podmorju / Video SLAM Technology in Underwater Environment**

Opis zadatka:

Potrebno je proučiti i opisati SLAM video tehnologiju i glavne značajke iste. Nadalje, potrebno je proučiti izradu podvodnih 3D modela razdvajanjem slika na R, G, B kanale s ostalim SLAM tehnikama. Konačno, potrebno je analizrati prednosti i nedostatke korištenja SLAM tehnologije za mapiranje podmorja.
Rad je izrađen u suradnji s tvrtkom Vectrino d.o.o.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Dominik Vičević

Zadatak uručen pristupniku: 21. ožujka 2022.

Mentor:

Izv. prof. dr. sc. Jonatan Lerga

Predsjednik povjerenstva za završni ispit:

Prof. dr. sc. Kristijan Lenac

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Dominik Vičević

Rijeka, rujan 2022.

Dominik Vičević

Sadržaj

1. Uvod

SLAM tehnologija danas postaje sve popularnija. To je zato što dopušta robotima, autonomnim vozilima, dronovima, itd. da mapiraju svoju okolinu, te da nađu svoj položaj u toj okolini. Potencijal ove tehnologije je gotovo neograničen. Moguća je autonomna vožnja, potraga i spašavanje u okruženjima visokog rizika ili okruženjima u kojima je teško upravljati, proširenoj stvarnosti, pa čak i u medicini. Minimalno invazivna kirurgija (engl. *minimally invasive surgery*) kombinira praćenje alata pomoću SLAM tehnologije s 3D modelima organa za prikaz detaljne 3D vizualizacije organa i same pozicije alata unutar organa. Jedan od ciljeva ovog rada je istraživanje i opis SLAM tehnologije s fokusom na video SLAM.

Još jedna od brojnih primjena SLAM tehnologije je mapiranje podmorja, što služi morskim biolozima, terenskim ekolozima, i slično u istraživanju i rješavanju problema. Temeljni problem primjene SLAM tehnologije u podmorju je njegova dinamičnost. Ribe se neprestano kreću, alge i meki koralji mijenjaju svoje oblike s morskom strujom, svijetlo se mijenja s obzirom na dubinu, itd. Drugi cilj ovog rada je istraživanje korištenja SLAM tehnologije u podmorju, razdvajanjem slika na crvene, zelene i plave (R, G, B) kanale.

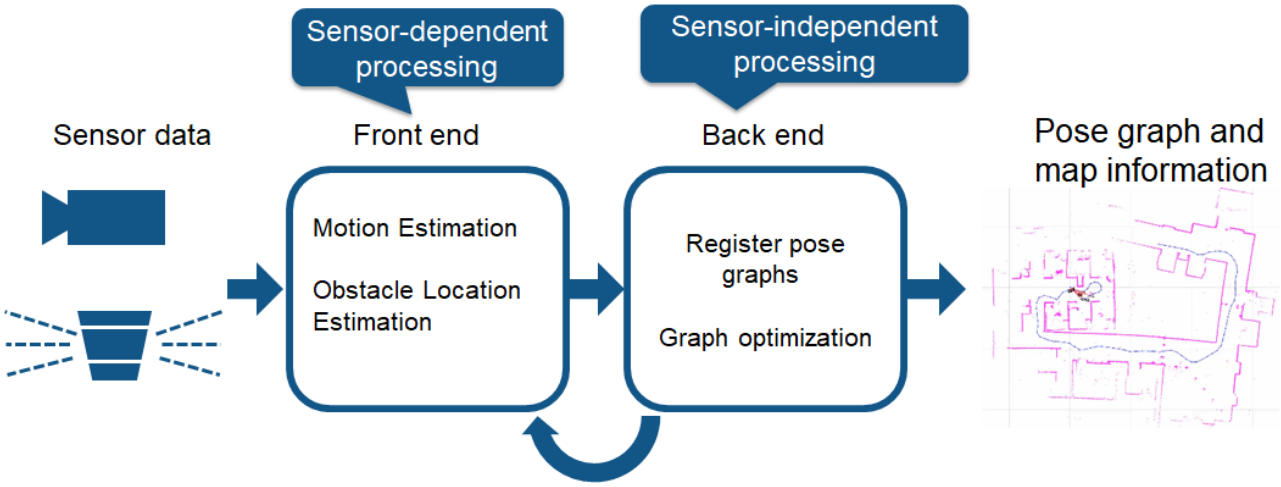
2. SLAM

SLAM (akronim od engl. *Simultaneous localization and mapping*: istodobna lokalizacija i mapiranje) je problem koji pita može li robot, koji stavljen na nepoznatu lokaciju u nepoznatoj okolini izgraditi dosljednu kartu te okoline, te odrediti i pratiti svoj položaj unutar te okoline [1]. Određivanje položaja u okolini pomoću kamera i drugih senzora nazivamo lokalizacijom, a korištenje kamere i ostalih senzora da bi se stvorila karta okoline nazivamo mapiranje. Proces mapiranja i lokalizacije se odvija istovremeno, zbog čega je tehnologija i dobila svoj naziv.

Iako postoji mnogo pojedinačnih rješenja za mapiranje i lokalizaciju, složenost SLAM-a dolazi od toga što se oba procesa (mapiranje i lokalizacija) odvijaju odjednom. Dugi niz godina se smatralo da je predmet koji konstruira kartu okoline dok prati vlastitu lokaciju klasični problem "kokoš ili jaje", bez jasnog rješenja. Danas, problem SLAM tehnologije teoretski je riješen u različitim oblicima i koristi se u robotskoj navigaciji, robotskom mapiranju i odometriji za virtualnu ili proširenu stvarnost [1]. Autonomna vožnja, koja je jedna od najpopularnijih tema danas, uz dodatne načine navigacije, isto koristi SLAM tehnologiju. SLAM je implementiran je u raznim domenama; od robota za zatvorene prostore, do robota za otvorene prostore, podvodnih robota i zračnih robota.

2.1. Općenita logika SLAM tehnologije

Općenito, postoje dvije vrste tehnoloških komponenti koje se koriste za postizanje SLAM-a. Prva vrsta je obrada signala senzora (engl. *sensor-dependent processing*), uključujući sučelja (engl. *front-end*) za obradu, koja uvelike ovisi o korištenim senzorima. Drugi tip je obrada signala neovisna o senzorima (engl. *sensor-independent processing*), koja uključuje back-end obradu [2] (slika 2.1).



Slika 2.1: Tehnološke komponente SLAM-a [2]

SLAM tehnologija funkcionira na sličan način kao i ljudi kada se nađu u nepoznatom okruženju.

Kao što mi pronalazimo orijentire u okolini i naš položaj u usporedbi s njima, tako i SLAM algoritam pronalazi orijentire u okruženju i pomoću određenih alata, specifičnih za tu SLAM metodu, pronalazi svoj položaj s obzirom na orijentir. Kretanjem pronalazi još orijentira sve dok ih ima dovoljno da stvori sveobuhvatnu kartu okoline.

Važno je napomenuti da SLAM nije jedan tehnološki proizvod ili jedinstveni sustav, nego koncept s gotovo beskonačnom količinom varijabilnosti. Kao što je već spomenuto, postoje brojna različita softverska rješenja i algoritmi, ali oni ovise o okruženju u kojem se mogu upotrijebiti.

2.1.1. Obrada signala senzora

Sva SLAM rješenja uključuju neku vrstu uređaja ili alata koji robotu ili drugom vozilu omogućuje promatranje i mjerenje okoline oko sebe. To mogu učiniti kamere, druge vrste senzora slike, lidar tehnologija laserskog skenera, pa čak i sonar. U osnovi, bilo koji uređaj koji se može koristiti za mjerenje fizičkih svojstava kao što su lokacija, udaljenost ili brzina može se uključiti kao dio SLAM sustava [2].

2.1.2. Obrada signala neovisna o senzorima

Nakon što se mjerenja dometa izračunaju, SLAM sustav mora imati neku vrstu softvera koji pomaže u interpretaciji tih podataka. Postoji širok raspon dostupnih opcija, od niza algoritama za isprepletanje do drugih vrsta složenog slaganja skeniranja (engl. *scan-matching*).

Sva ova rješenja služe istoj svrsi: izdvajanje senzorskih podataka koje prikuplja uređaj za mjerenje dometa i korištenje istih za prepoznavanje orijentira unutar nepoznatog okruženja [2]. Pravilno, funkcionirajuće SLAM rješenje podrazumijeva stalnu interakciju između uređaja za mjerenje dometa, softvera za ekstrakciju podataka, samog robota ili vozila, dodatnog hardvera, softvera ili drugih uključenih tehnologija obrade. Svi ovi elementi su varijabilni ovisno o slučaju upotrebe, ali kako bi bilo koji SLAM sustav točno istražio svoje okruženje, sve ove stavke moraju raditi zajedno besprijekorno [2].

2.2. Uobičajeni izazovi SLAM tehnologije

Iako se SLAM koristi u praktičnim primjenama, nekoliko tehničkih izazova nam stvaraju prepreku u posvojenju rješenja opće namjene.

Prvi od tih izazova je što se greške lokalizacije gomilaju, što, nakon nekog vremena, uzrokuje znatno odstupanje od stvarnih vrijednosti. Možemo uzeti za primjer robota koji se vozi unutar prolaza u obliku kocke. Kako se greške gomilaju, kada se robot vrati na početnu poziciju u stvarnome prostoru,

po njegovoj lokalizaciji to nije početna pozicija. Ovaj problem se naziva problem zatvaranja petlje (engl. *loop closure problem*). Ovakve pogreške u procjeni položaja su neizbježne. Važno je otkriti zatvaranja petlje i odrediti kako ispraviti ili poništiti akumulirane pogreške.

Slika 2.2: Primjer akumuliranja pogrešaka u pronalaženju položaja [2]

Jedno od rješenja ovog problema je optimiziranje grafa položaja (engl. *pose graph optimization*). Optimizacija položaja se postiže tako što uspoređujemo orijentire iz već posjećenog položaja s tim istim orijentirima ako ih vidimo iz drugog položaja. To nam omogućuje da procijenimo greške u lokalizaciji, što nam dozvoljava da ih probamo popraviti [2].

Slika 2.3: Primjena optimizacije grafa položaja na sliku 2.2 [2]

Korištenje optimizacije grafa položaja nam daje točnije karte okoline. Ovakva optimizacija se naziva podešavanje snopa (engl. *bundle adjustment*) u video SLAM-u.

Sljedeći izazov je situacija u kojoj lokalizacija ne uspije i položaj na karti je izgubljen. Mapiranje slike i oblaka točaka ne uzima u obzir karakteristike kretanja robota. U nekim slučajevima ovaj pristup može generirati diskontinuirane procjene položaja. Na primjer, rezultat izračuna koji pokazuje da je robot koji se kretao brzinom od 1 m/s iznenada skočio naprijed za 10 metara. Ova vrsta neuspješne lokalizacije može se spriječiti korištenjem algoritma za oporavak ili spajanjem modela kretanja s više senzora kako bi se izvršili izračuni na temelju podataka senzora.

Postoji nekoliko metoda za korištenje modela kretanja s fuzijom senzora. Uobičajena metoda je korištenje Kalman filtriranja za lokalizaciju. Kalman filtriranje je algoritam koji koristi niz mjerenja kako bi proizveo procjene nepoznatih varijabla [3]. Budući da većina robota s diferencijalnim pogonom (engl. *differential wheeled robot*) i vozila s četiri kotača koriste nelinearne modele gibanja često se koristi prošireni Kalmanov filtar (engl. *extended Kalman filter*), što je nelinearna verzija Kalman filtra namjenjena nelinearnim sustavima [4].

Kod neuspješne lokalizacije, jedno od rješenja je oporavak pomoću pamćenja orijentira kao ključnog kadra (engl. *key frame*) s prethodno posjećenog mjesta. Prilikom traženja orijentira primjenjuje se proces izdvajanja obilježja koji može skenirati velikom brzinom. Neke metode temeljene na značajkama slike uključuju BoF (akronim od engl. *Bag of Features*: vreća značajki) i BoVW (akronim od engl. *Bag of Visual Words*: vreća vizualnih riječi) [2]. Za usporedbu udaljenosti od značajki može se koristiti i duboko učenje.

Zadnji izazov je visok računalni trošak implementacije SLAM tehnologije na hardveru nekog vozila. Računanje se obično izvodi na kompaktnim ugradbenim sustavima niske potrošnje energije koji imaju ograničenu procesorsku snagu. Kako bi se postigla točna lokalizacija, bitno je izvršiti obradu slike i podudaranje oblaka točaka visokom frekvencijom. Osim toga, optimizacijski izračuni kao što je zatvaranje petlje su računski vrlo zahtjevni [2].

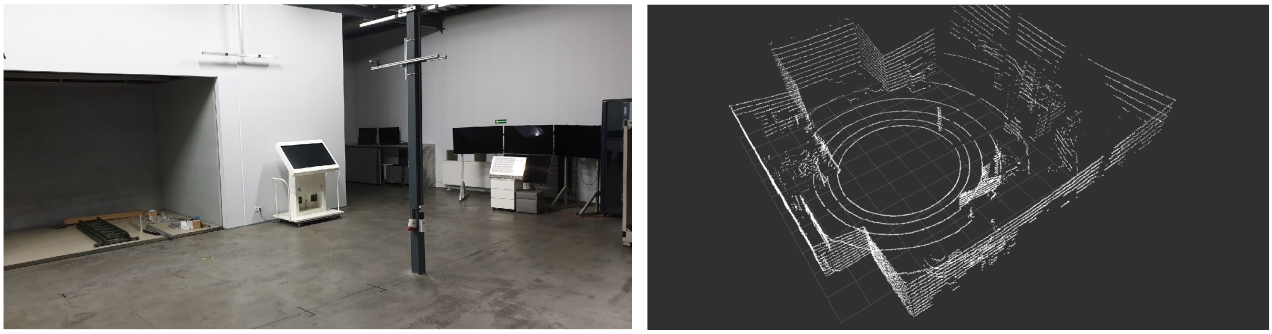
Izvođenje različitih procesa paralelno, koristeći više-jezgrene procesore i integrirane grafičke kartice može poboljšati performanse. Isto tako optimizacija grafa položaja se može izvoditi tijekom relativno dugog vremena, tako da bi pridodali veći prioritet drugim procesima i tako poboljšali performanse.

S obzirom zastupljenost i na vrstu informacija koje SLAM algoritam koristi, SLAM tehnologiju možemo podijeliti na dvije opće metode: Lidar SLAM i Video SLAM.

2.3. Lidar SLAM

Lidar SLAM je SLAM metoda koja koristi Lidar senzor. Lidar (akronim od engl. *Light Detection and Ranging*: svjetlosno otkrivanje i klasifikacija) je metoda određivanja udaljenosti ciljanjem nekog objekta s laserom i mjerenjem vremena potrebnog da se odbijeno svjetlo vrati do prijamnika.

U usporedbi s kamerama, ToF (akronim od engl. *Time of Flight*: vrijeme letenja) kamerama i drugim senzorima, laseri su znatno precizniji i koriste se za aplikacije s vozilima koja se kreću velikom brzinom kao što su samovozeći automobili i dronovi. Izlazne vrijednosti lidar senzora su općenito 2D (x, y) ili 3D (x, y, z) podaci oblaka točaka. Oblak točaka laserskog senzora pruža precizna mjerenja udaljenosti i vrlo je učinkovit za izradu karte okoline pomoću SLAM tehnologije. Općenito, kretanje kamere se procjenjuje sekvencijalnim usklađivanjem oblaka točaka. Najčešći algoritmi za usklađivanje oblaka točaka su ICP (akronim od engl. *Iterative closest point*: iterativna najbliža točka) i NDT (akronim od engl. *normal distributions transform*: transformacija normalne distribucije) algoritama [5].



Slika 2.4: Lidar oblak točaka, usporedba sa stvarnosti [5]

Kao što se vidi na slici, oblaci točaka koji proizlaze iz lidar senzora nemaju toliko finih detalja, po pitanju gustoće, kao što slike imaju. To znači da u situacijama gdje okolina ima malo objekata definiranih obruba, teško je uskladiti oblake točaka, što može uzrokovati gubitkom položaja vozila ili robota. Osim toga, usklađivanje oblaka točaka zahtjeva veliku procesorsku snagu, stoga je potrebno optimizirati procese kako bi poboljšali brzinu algoritma.

2.4. Video SLAM

Video SLAM ili vSLAM koristi slike dobivene s kamera i drugih senzora slike. Za video SLAM može se koristiti: jednu RGB kameru (što nazivamo monokularan SLAM), više RGB kamera (Stereo SLAM) ili jednu RGB-D kameru (RGB-D SLAM) [2]. RGB kamera nam dostavlja slike hvatanjem svjetlosti u valnim duljinama crvene, zelene i plave boje, te njihovim spajanjem prikazuje različite boje na slici. Razlika između RGB i RGB-D kamere je što nam RGB-D kamera dostavlja dodatan

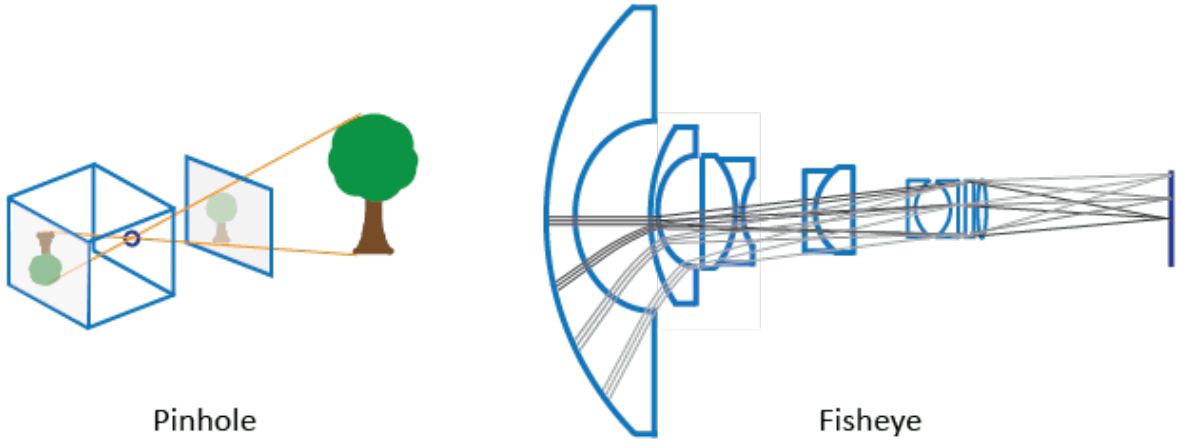
podatak dubine. Informacije o dubini moguće je dohvatiti putem karte/slike dubine koju stvara 3D senzor dubine kao što je senzor vremena leta (npr. već spomenuta ToF kamera) ili korištenjem dodatne kamere kao što se radi u stereo SLAM tehnologiji.

Video SLAM može se implementirati po niskoj cijeni s relativno jeftinim kamerama. Budući da kamere pružaju veliku količinu informacija, mogu se koristiti za otkrivanje orijentira (prethodno izmjerene pozicije). Detekcija orijentira također se može kombinirati s optimizacijom gafa položaja , čime se postiže fleksibilnost u implementaciji SLAM tehnologije [2].

Algoritmi video SLAM-a mogu se općenito klasificirati u dvije kategorije. Rijetke (engl. *sparse*) metode koje podudaraju značajke slika i koriste algoritme kao što su PTAM ili ORB-SLAM. Guste (engl. *dense*) metode koje koriste ukupnu svjetlinu slike i koriste algoritme kao što su DTAM, DSO, SVO [2].

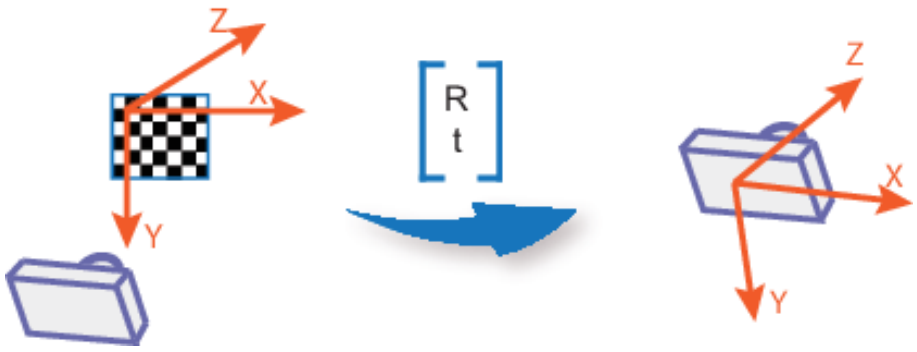
2.4.1. Kalibracija kamere

Svaka kamera iskrivljuje slike kada ih snimi. Video SLAM zahtjeva ne iskrivljene slike da bi mogao ispravno mapirati okolinu, zato nam treba kalibracija kamere. Geometrijska kalibracija kamere procjenjuje parametre leće i senzora slike slikovne ili video kamere. Pomoću ovih parametara može se ispraviti izobličenje leće, izmjeriti veličinu objekta u svjetskim jedinicama ili odrediti položaj kamere u sceni. Parametri kamere uključuju unutarnje (engl. *intrinsic*), vanjske (engl. *extrinsic*) i koeficijente izobličenja (engl. *distortion parameters*) [6]. Razlikujemo dvije vrste modela kamere: *pinhole* kamera i *fisheye* kamera. Da bi se procijenili parametri kamere, potrebne su nam 3D točke svijeta i njihove odgovarajuće 2D točke slike. Ove podudarnosti se dobivaju pomoću više slika kalibracijskog uzorka, kao što je šahovnica. Korištenjem podudaranja točaka između slika dobivamo parametre kamere.



Slika 2.5: Modeli kamera [6]

Kao što je već spomenuto parametri kamere uključuju unutarnje i vanjske parametre, te koeficijente izobličenja. Vanjski parametri se sastoje od rotacije R i translacije T (slika 2.6), koji predstavljaju transformaciju iz 3D koordinatnog sustava svijeta u 3D koordinate koordinatnog sustava kamere.



Slika 2.6: Vanjski parametri kamere [6]

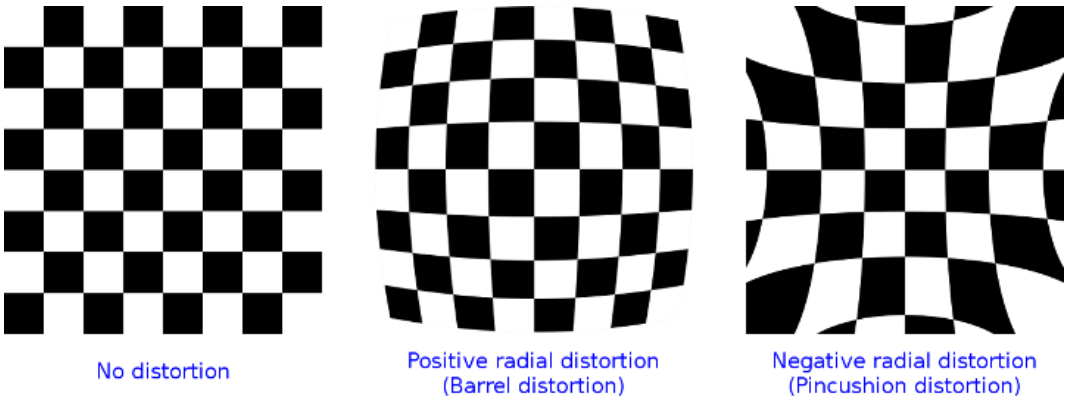
Unutarnji parametri kamere uključuju žarišnu duljinu (engl. *focal length*), optički centar (također poznat kao glavna točka engl. *optical center, principal point*) i koeficijent zakrivljenosti (engl. *skew coefficient*). Ti parametri predstavljaju projektivnu transformaciju iz 3D koordinatnog sustava kamere u 2D koordinate slike. Matrica unutarnjih parametara je oblika u izrazu (2.1).

$$\begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

gdje je:

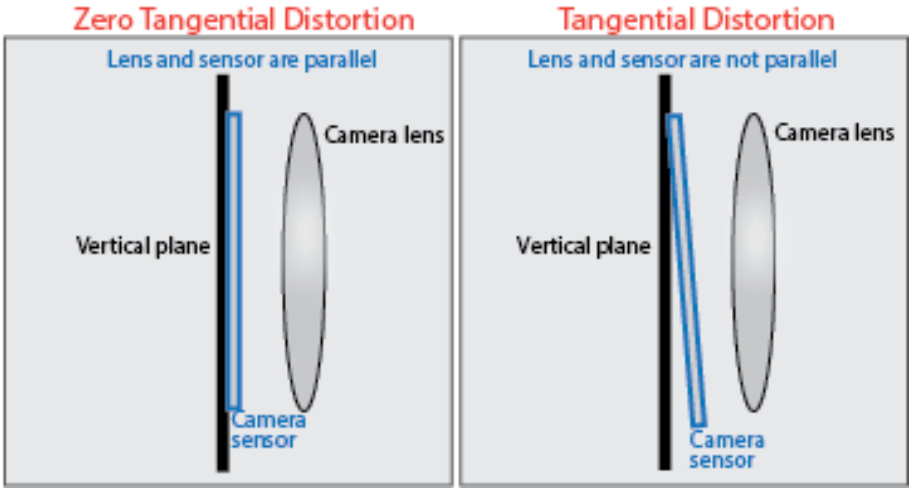
f_x = žarišna duljina (x os) u pikselima
f_y = žarišna duljina (y os) u pikselima
s = koeficijent zakrivljenosti
c_x = optički centar (x os) u pikselima
c_y = optički centar (y os) u pikselima

Koeficijentima izobličenja rješavamo problem radijalnog i tangencijalnog izobličenja leće. Radijalno izobličenje se događa kada se svjetlosne zrake savijaju više na rubovima leće nego u optičkom centru. Razlikujemo pozitivno i negativno radijalno izobličenje (slika 2.7). Koeficijenti radijalnog izobličenja su k_1, k_2, k_3 [6].



Slika 2.7: Vrste radijalnog izobličenja [7]

Tangencijalno izobličenje leće se događa kada leća i ravnina slike nisu paralelni (slika 2.8). Koeficijenti tangencijalnog izobličenja su p_1, p_2 [6].



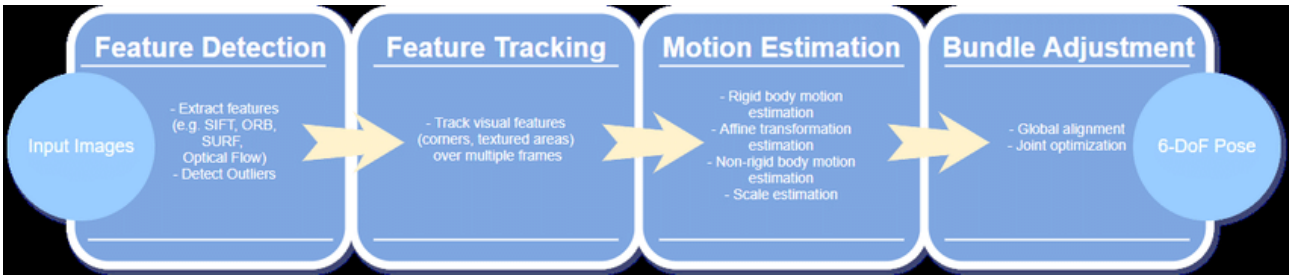
Slika 2.8: Tangecijalno izobličenje [6]

2.4.2. Vizualna odometrija

U robotici i računalnom vidu, vizualna odometrija je proces određivanja položaja i orijentacije robota analizom pridruženih slika kamere [8]. Za video SLAM postoji veliki istraživački problem, to jest, generirati petlju i učinkovito integrirati nova ograničenja na trenutnu kartu na temelju vizualne odometrije.

Postoje dvije glavne metode temeljenih na video SLAM-u. Jedan je opći pristup za primjenu klasičnog filtra spoju vizualnih informacija (engl. *visual information fusion*). Drugi je iskorištavanje odabranih ključnih kadrova (engl. *key frame*) za razvoj globalne optimizacije [9]. Ključni okvir je okvir dobiven od kamere koji sadrži ključne informacije o nekom orijentiru u okolini. Veza između video SLAM-a i vizualne odometrije je u tome što se potonji može smatrati modulom unutar prvog

i može postupno rekonstruirati putanju kamere. Neki znanstvenici smatraju video-SLAM daljnjim istraživanjem vizualne odometrije [9]. Razlika između video SLAM-a i vizualne odometrije je u tome što se vizualna odometrija usredotočuje samo na dosljednost lokalnih putanja, dok se video SLAM usredotočuje na dosljednost globalne putanje.



Slika 2.9: Općeniti "pipeline" vizualne odometrije [10]

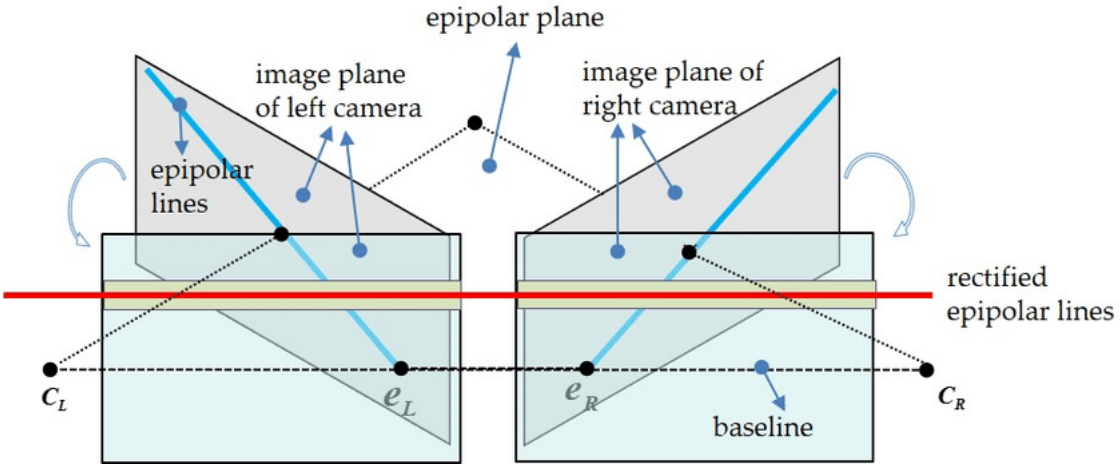
Vizualna odometrija se može obavljati na jednoj kameri (monokularna vizualna odometrija) ili na dvije kamere (stereo vizualna odometrija).

Kod monokularne vizualne odometrije koristi se set okvira koje kamera snima. Iz tog seta uzimaju se dva okvira snimljena jedan nakon drugog, te se računa rotacijska i translacijska matrica između ta dva okvira. Računanje tih matrica se provodi na svakome paru slijednih okvira i pomoću toga možemo izračunati putanju kamere. Translacijsku i rotacijsku matricu računamo tako što nađemo značajke (engl. *feature*) na oba okvira, te ih onda podudaramo (identificiramo iste značajke na oba okvira). Usporedbom položaja podudarajućih značajki u koordinatnom sustavu ekrana (engl. *screen space*) možemo izračunati koliko se kamera rotirala i translatirala između ta dva okvira, čime dobivamo translacijsku i rotacijsku matricu.

Stereo vizualna odometrija funkcionira na sličan način. Razlika je u tome što se umjesto dva slijedna okvira uzimaju četiri (dva za svaku kameru) i što se generiraju 3D točke podudarajućih značajki, da se može procijeniti kretanje kamera.

2.4.3. Stereo rektifikacija

Računalni stereo vid koristi dvije ili više kamera s poznatim relativnim položajem, koje prikazuju isti objekt iz različitih kutova, tako da bi mogao utvrditi odgovarajuću dubinu točaka na slici [11]. Za svaki se piksel može odrediti dubina tako što se prvo pronalaze podudarajući pikseli na obije slike (pikseli koji pokazuju na istu točku u svijetu), a zatim, primjenom triangulacije na pronađena podudaranja, možemo izračunati dubinu [11]. Pronalaženje podudarnosti u stereo vidu ograničeno je epipolarnom geometrijom: podudarnost svakog piksela u drugoj slici može se pronaći samo na liniji koja se naziva epipolarna linija (slika 2.10).



Slika 2.10: Diagram stereo rektifikacije [11]

Ako su dvije slike komplanarne, tj. snimljene su tako da je desna kamera samo vodoravno pomaknuta u usporedbi s lijevom kamerom, tada je epipolarna linija svakog piksela vodoravna i na istoj y poziciji kao taj piksel. Međutim, u općim situacijama (kamera se kretala prema objektu ili rotirala) epipolarne linije su nagnute. Stereo rektifikacija iskrivljuje obje slike tako da izgledaju kao da su snimljene samo s horizontalnim pomakom i kao posljedica toga sve su epipolarne linije vodoravne, što pojednostavljuje postupak stereo usklađivanja [11].

3. vSLAM razdvajanjem slike na R, G, B kanale

3.1. ROS

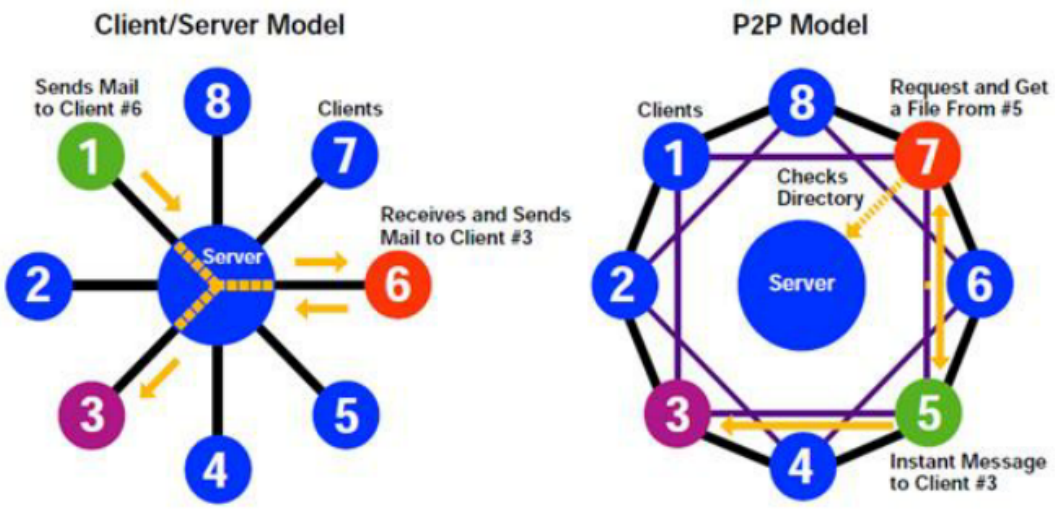
ROS (akronim od engl. *Robot Operating System*: robotski operativni sustav) je komplet razvojnih alata otvorenog koda (engl. *open source*) za primjene robotike [12]. ROS nudi standardnu softversku platformu programerima u svim industrijama koja će ih voditi od istraživanja i izrade prototipa sve do implementacije i proizvodnje. ROS je korišten za implementaciju vSLAM-a za svrhe ovoga rada.

Dizajniran je da bude otvorenog koda, s namjerom da korisnici mogu odabrati konfiguraciju alata i biblioteka koje su u interakciji s jezgrom ROS-a, tako da korisnici mogu premjestiti svoje softverske pakete kako bi odgovarali njihovom robotu i području primjene. Kao takvo, vrlo je malo toga je srž ROS-a, osim opće strukture unutar koje programi moraju postojati i komunicirati.
Filozofski ciljevi ROS-a se mogu sažeti kao:

- *Peer to peer*
- Temeljen na alatima (engl. *Tools-based*)
- Višejezični (engl. *Multi-lingual*)
- Tanak (engl. *Thin*)
- Besplatan i otvorenog koda (engl: *Free and Open-Source*) [13]

3.1.1. *Peer to peer*

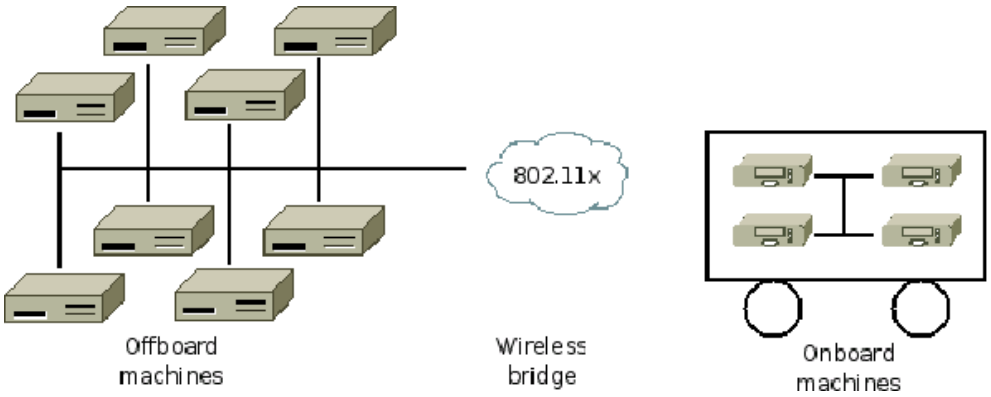
Sustav izgrađen korištenjem ROS-a sastoji se od niza povezanih procesa, potencijalno na više različitih računala, za vrijeme izvođenja koristeći *peer to peer* topologiju (slika 3.1) [13]. *Peer to peer* topologija znači izbjegavanje korištenja središneg poslužitelja (engl. *central server*), tako što se dijeljenje resursa i servisa između računala radi pomoću izravne razmjene. Iako se *peer* računalo ponaša kao klijent u klient/poslužitelj modelu, ono sadrži dodatni sloj softvera koji mu omogućava izvođenje funkcija poslužitelja [14]. ROS koristi *peer to peer* topologiju zbog heterogenih mreža, koja su česta pojava pri korištenju ROS-a.



Slika 3.1: Klijent/poslužitelj i peer to peer *modeli [14]*

Na primjer, na velikim servisnim robotima za koje ROS dizajniran, obično postoji nekoliko ugrađenih računala na brodu spojena putem etherneta. Taj segment mreže je premošten putem bežične LAN mreže do izvan brodskih strojeva velike snage koji izvršavaju računalno intenzivne zadatke poput računalnog vida ili prepoznavanja govora [13]. Korištenje središnjeg poslužitelja u ovome slučaju ne bi bilo adekvatno. To bi rezultiralo sporijom vezom za cijeli sustav, zato što bi promet mreže trebao prolaziti preko središnjeg poslužitelja. Takav promet je nepotreban jer se mnoge rute poruka nalaze u pod-mrežama na brodu ili izvan njega [13]. *Peer to peer* topologija nas služi bolje u ovome primjeru.

Na slici 3.2 vidimo primjer tipične konfiguracije ROS mreže.



Slika 3.2: Primjer tipične konfiguracije ROS mreže [13]

Peer to peer topologija zahtijeva neku vrstu mehanizma pretraživanja koji omogućuje procesima da pronađu jedni druge. Taj proces se naziva "ROS Master", i biti će detaljnije objašnjen kasnije u tekstu.

3.1.2. Temeljen na alatima

Da bi se lakše upravljalo kompleksnošću ROS-a, korišten je dizajn mikrojezgre (engl. *microkernel design*), gdje se veliki broj malih alata koristi za izgradnju i pokretanje raličitih ROS komponenti [13]. Ovi alati obavljaju razne zadatke, poput navigacije stabla izvornog koda, dobivanja i postavljanja (engl. *get and set*) konfiguracijskih parametara, vizualiziranja topologije *peer to peer* veze, mjerenja korištenja propusnosti (engl. *bandwidth*), i tako dalje [13].

3.1.3. Višejezičan

Različiti problemi zahtijevaju različite alate za njihovo rješavanje. To je razlog velikoj količini različitih programskih jezika. Programeri moraju odabrati programski jezik koji najbolje odgovara njihovim potrebama, ali to nije jedini faktor kod odabira jezika. Lakoća otklanjanja pogreški, razumljivost sintakse i čitljivost jedni su od brojnih razloga zbog kojih netko preferira jedan programski jezik od drugog.

Zbog tih razloga je ROS dizajniran da bude "jezično neutralan". ROS trenutno podržava četiri vrlo različita jezika: C++, Python, Octave i LISP, s dodacima za druge jezike različitih stanja dovršenosti [13]. Umjesto pružanja implementacije temeljene na C programskom jeziku s generiranim sučeljima za sve glavne jezike, ROS je implementiran izvorno u svakom ciljanom jeziku, kako bi se bolje pratila konvencija pisanja svakog programskog jezika [13]. Međutim, u nekim slučajevima svrsishodno je dodati podršku za novi jezik omatanjem postojeće knjižnice. Octave klijent je implementiran omatanjem ROS C++ biblioteke [13].

Kako bi podržao višejezični razvoj, ROS koristi jednostavan, jezično neutralan IDL (akronim od engl. *Interface Definition Language*: jezik definicije sučelja) za opisivanje poruka koje se šalju između modula [13]. ROS-ov IDL koristi kratke tekstualne datoteke koje opisuju polja koja se nalaze u porukama. Generatori koda pročitaju IDL datoteku i proširuju je u implementaciju napisanu u odabranom programskom jeziku, ako je taj programski jezik podržan od strane ROS-a, koja sliči izvornim objektima. ROS automatski serijalizira i deserializira te objekte kako se poruke šalju i primaju.

Krajnji rezultat je jezično neutralna obrada poruke u kojoj se različiti programski jezici mogu miješati i slagati po želji.

3.1.4. Tanak

Različit softver koji spada u neku disciplinu često ima dijelove (module, algoritme, itd.) koji bi se mogli ponovno iskoristiti u drugim projektima, ali je to nemoguće, zato što su ti dijelovi toliko vezani uz izvorni program da je nemoguće "izvući van" samo funkcionalnost tih dijelova bez velikih prilagodbi. Tako je i sa robotikom. U robotici, brojni upravljački programi (engl. *drivers*) ili algoritmi

bi se mogli ponovno iskoristiti da nisu toliko isprepleteni uz sami program za koji su proizvedeni. ROS rješava ovaj problem tako što potiče razvoj upravljačkih programa i algoritama unutar knjižnica koje nemaju ovisnosti jedna o drugoj ili o ROS-u.

ROS sustav izgradnje (engl. *build system*) izvodi modularne izgradnje unutar stabla izvornog koda, i njegova upotreba "CMake"-a omogućava praćenje ROS-ove "tanke" ideologije s relativnom lakoćom [13]. Stavljanje gotovo sve složenosti u knjižnice, i stvaranje samo malih izvršnih datoteka koje izlažu funkcionalnost knjižnice ROS-u, dopušta lakše izdvajanje koda za ponovno korištenje izvan svoje izvorne namjere [13].

Ovakav pristup donosi dodatnu pogodnost kod testiranja, koje je lakše izvoditi kada je kod faktoriziran u različite knjižnice, jer se njihove pojedinačne funkcionalnosti mogu testirati odvojeno.

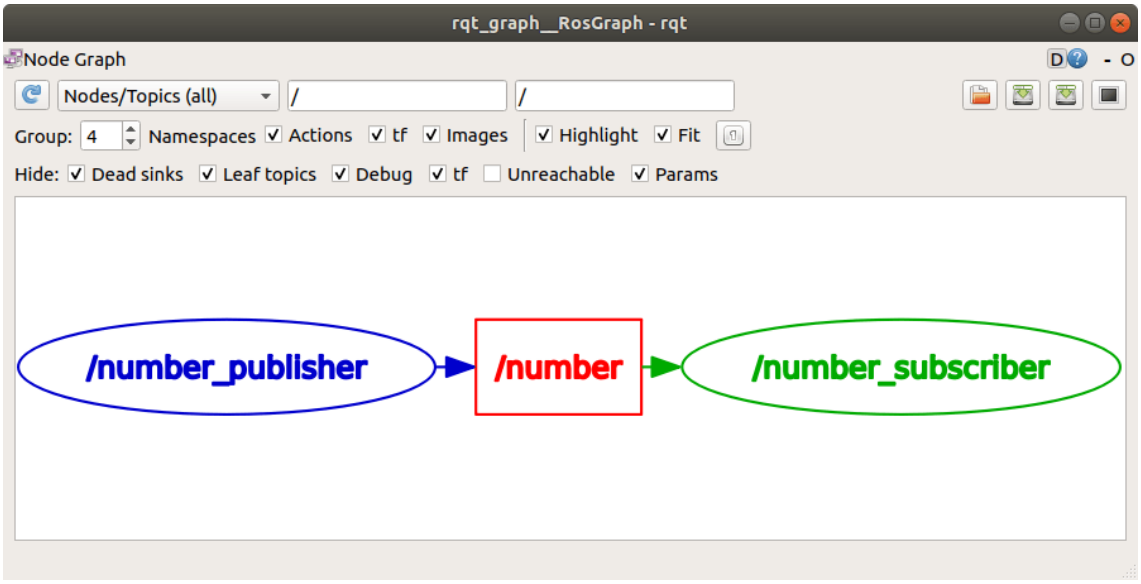
3.1.5. Besplatan i otvorenog koda

Izvorni kod ROS-a je javno dostupan. Na prvi pogled to se čini kao pogreška; ako je kod javno dostupan, to bi značilo da bilo tko može vidjeti i iskoristiti njegove ranjivosti. Dok je to istina, baš zato što je javno dostupan, svi mogu vidjeti i otkloniti bilo kakve probleme unutar ROS-a. Na primjer, puno je manja vjerojatnost da se problem ne zamijeti od strane 100 ljudi, nego od jedne osobe.

ROS se distribuira pod uvjetima licence BSD, koja dopušta razvoj i nekomercijalnih i komercijalnih projekata [13]. ROS prenosi podatke između modula koristeći komunikaciju među procesima i ne zahtijeva da se moduli međusobno povezuju u istu izvršnu datoteku, što je primjer ROS-ove "tanke" filozofije. Kao takav, sustavi izgrađeni oko ROS-a mogu posebno licencirati različite komponente. Pojedinačni moduli mogu sadržavati softver zaštićen različitim licencama od GPL do BSD do vlasničke (engl. *proprietary*) licence, ali "kontaminacija" licence završava sa granicama modula [13].

3.1.6. ROS struktura

Temeljni koncepti implementacije ROS-a su čvorovi (engl. *nodes*), poruke (engl. *messages*), teme (engl. *topics*) i usluge (engl. *services*) [13]. ROS ima strukturu grafa, gdje čvorovi predstavljaju čvorove, a teme predstavljaju bridove koji spajaju te čvorove. Čvorovi si međusobno šalju poruke ili zahtijevaju usluge putem tema. Na slici 3.3 vidimo primjer ROS grafa, vizualiziran pomoću "*rqt_graph*" čvora, koji služi kao grafičko sučelje za vizualiziranje ROS grafa.



Slika 3.3: Primjer ROS grafa vizualiziranog pomoću rqt_graph čvora (elipse su čvorovi, a pravokutnik je tema) [15]

Proces zvan "ROS Master" čini sve ovo mogućim; on registrira čvorove sebi, postavlja komunikaciju između čvorova za teme i kontrolira ažuriranja poslužitelja parametra (engl. *parameter server*). Poruke i usluge ne putuju kroz "ROS Master" već on postavlja *peer to peer* komunikaciju između svih procesa čvorova nakon što ih registrira.

Čvor predstavlja jedan pokrenuti proces na ROS grafu. Čvorovi međusobno komuniciraju pomoću poruka. Poruka je striktno klasificirana struktura podataka. Standardne primitivne vrste podataka (int, float, boolean, itd.) su podržane, isto kao i nizovi primitivnih vrsti podataka i konstante. Poruke mogu biti komponirane od drugih poruka i nizova poruka, ugniježđenih proizvoljno duboko [13].

Svaki čvor ima ime, koje registrira kod na "ROS Master"-u prije nego što može poduzeti bilo koju drugu radnju. Teme su imenovane sabirnice preko kojih čvorovi šalju i primaju poruke. Za slanje poruka na temu, čvor mora objavljivati na navedenoj temi, dok se za primanje poruka čvor mora pretplatiti na tu istu temu. Model objavljivanja/pretplate je anoniman: nijedan čvor ne zna koji čvorovi šalju ili primaju poruke s te teme, samo da se poruke šalju ili primaju.

Čvor također može oglašavati usluge. Usluga predstavlja radnju, koju čvor može poduzeti, koja će imati jedan rezultat. Kao takve, usluge se često koriste za radnje koje imaju definiran početak i kraj, kao što je snimanje slike jednog okvira, umjesto obrade naredbi za brzinu motoru kotača ili podataka brojača kilometara iz kodera kotača. Čvorovi reklamiraju usluge i pozivaju usluge jedni od drugih.

Poslužitelj parametara je baza podataka koja se dijeli između čvorova koja omogućuje zajednički pristup statičkim ili polustatičkim informacijama. Podaci koji se ne mijenjaju često i kao takvi će im se rijetko pristupati dobri su kandidati za pohranu u poslužitelju parametara.

3.1.7. RViz

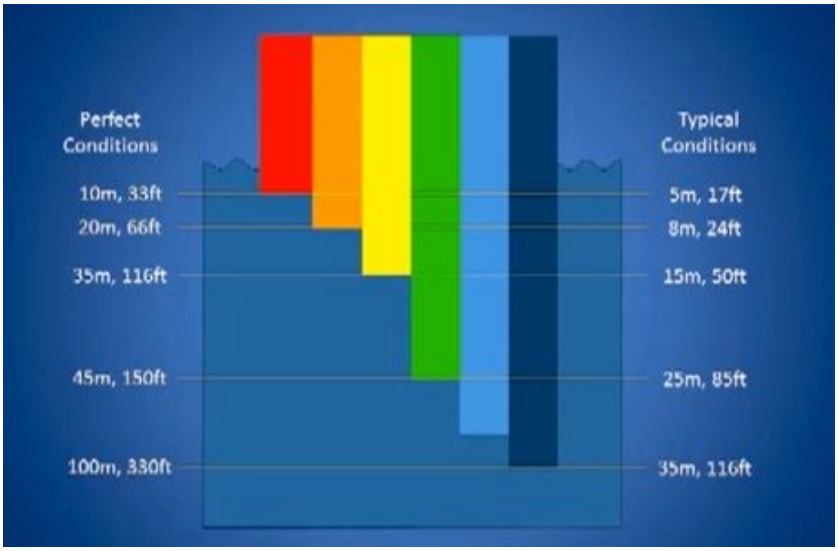
RViz je trodimenzionalni vizualizator koji se koristi za vizualizaciju robota, njihovih okruženja, podataka senzora itd. Svi oblaci točaka u ovome radu su vizualizirani u RViz-u

3.1.8. Video_stream_opencv

"Video_stream_opencv" je paket koji sadrži čvor za objavljivanje video snimke preko ROS tema. "Video_stream_opencv" se u ovome radu koristi za simuliranje podataka u stvarnome vremenu.

3.2. RGB kanali u podmorju

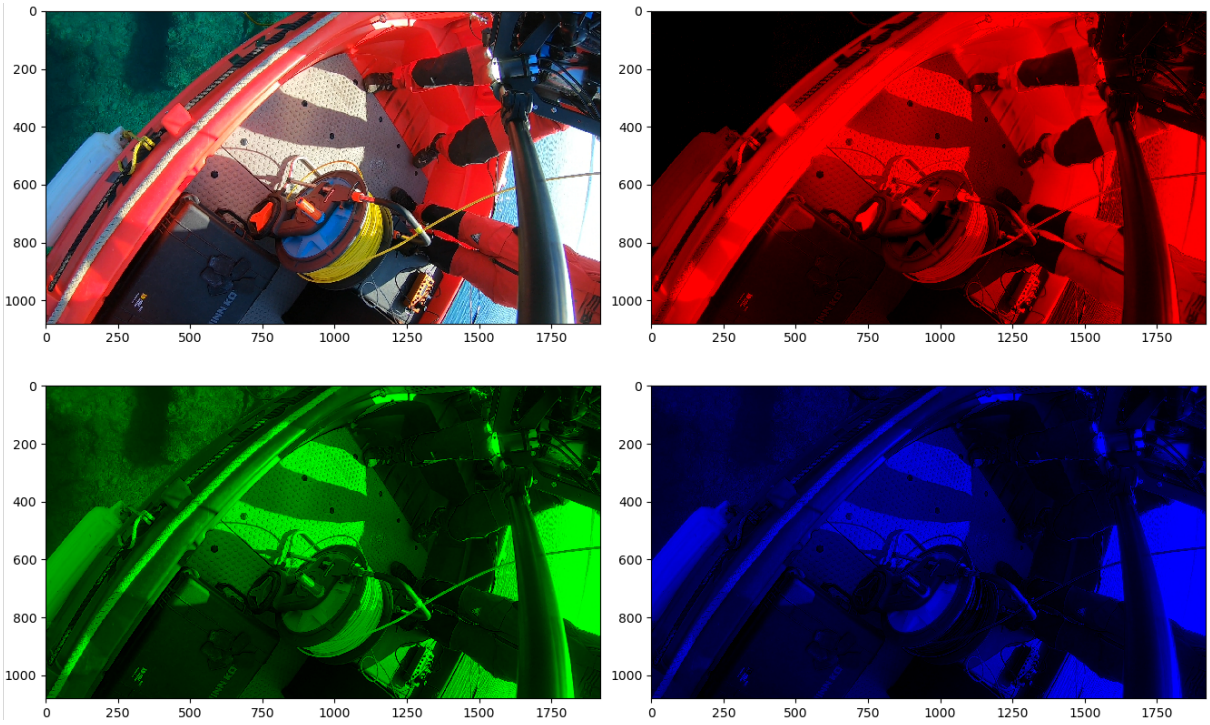
Vidljivo "bijelo" svjetlo sastoji se od spektra boja: ljubičaste, plave, zelene, žute, narančaste, crvene (poredane rastućim redoslijedom valnih duljina). Pod vodom dolazi do eksponencijalnog gubitka intenziteta svjetlosti koji ovisi o valnoj duljini svake boje. Taj se fenomen naziva selektivna apsorpcija (engl. *selective absorption*) mora i posljedica je vibracija i deformacija molekula vode pobuđenih apsorpcijom svjetlosti [16]. Ljudi ni ne primjećuju selektivnu apsorpciju vode. To je zato što naš mozak nadoknađuje izgubljene boje, ali ako probamo slikati kamerom vidjet će se razlika. Količina apsorbirane svjetlosti ovisi o valnim duljinama svjetla (slika 3.4) (svjetlo veće valne duljine se prije apsorbira), o prozirnosti vode (sediment, plankton, itd.), vremenu i položaju sunca. Valovito more odbija više svjetla nego mirno more, isto tako, ako je sunce na horizontu, puno više svjetla će se odbijati od mora nego ako se sunce nalazi ravno iznad. Crvena boja, koja ima najveću valnu duljinu, smanjuje se na trećinu svog intenziteta nakon jednog metra, i uglavnom se gubi nakon udaljenosti od 4 ili 5 metara [16].



Slika 3.4: Selektivna absorpcija mora [16]

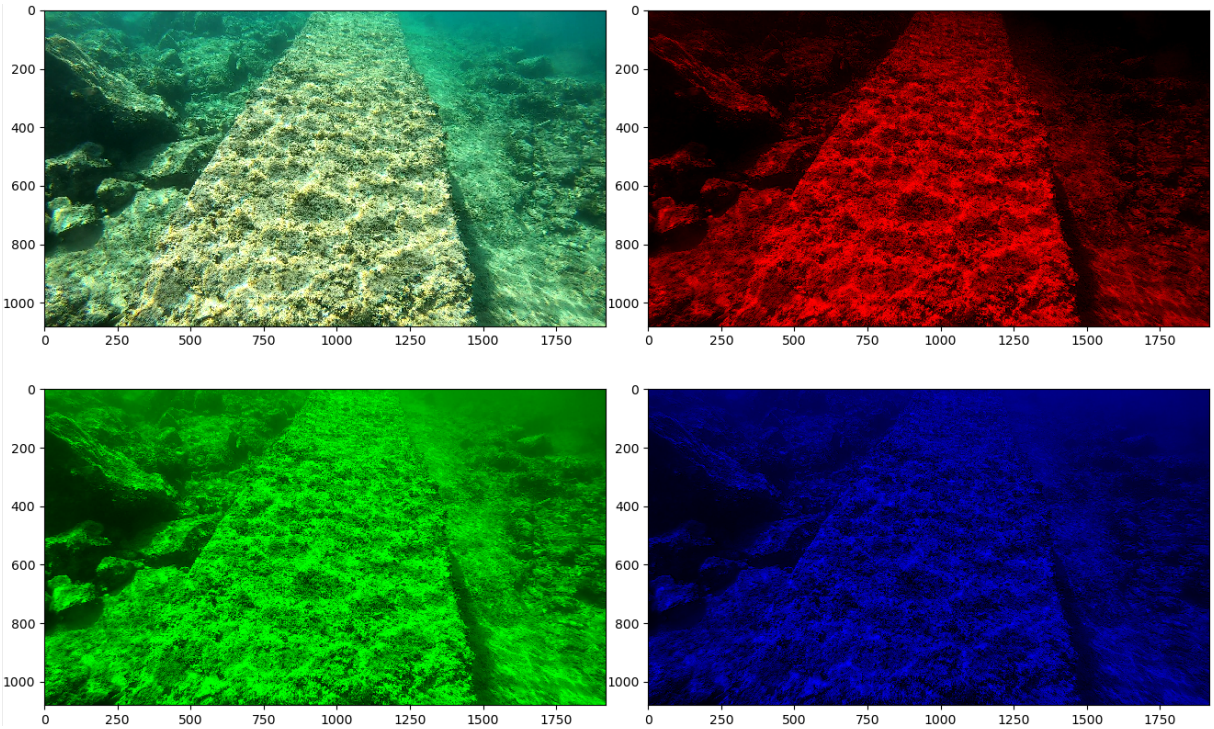
Opadanje boje pod morem se ne odnosi samo na dubinu, već i na udaljenost s koje gledamo objekt. Ako na dubini od 3 metra, kamerom snimamo crveni objekt udaljen 3 metra, svjetlo zapravo putuje 6 metara, pa će se crvena boja ipak izgubiti.

Slika 3.5 prikazuje sliku broda i R, G, B kanale te slike. Na njoj se može vidjeti standardna raspodjela na RGB kanale. Na crvenom kanalu primjećuje se da se more s gornje lijeve strane broda uopće ne vidi.



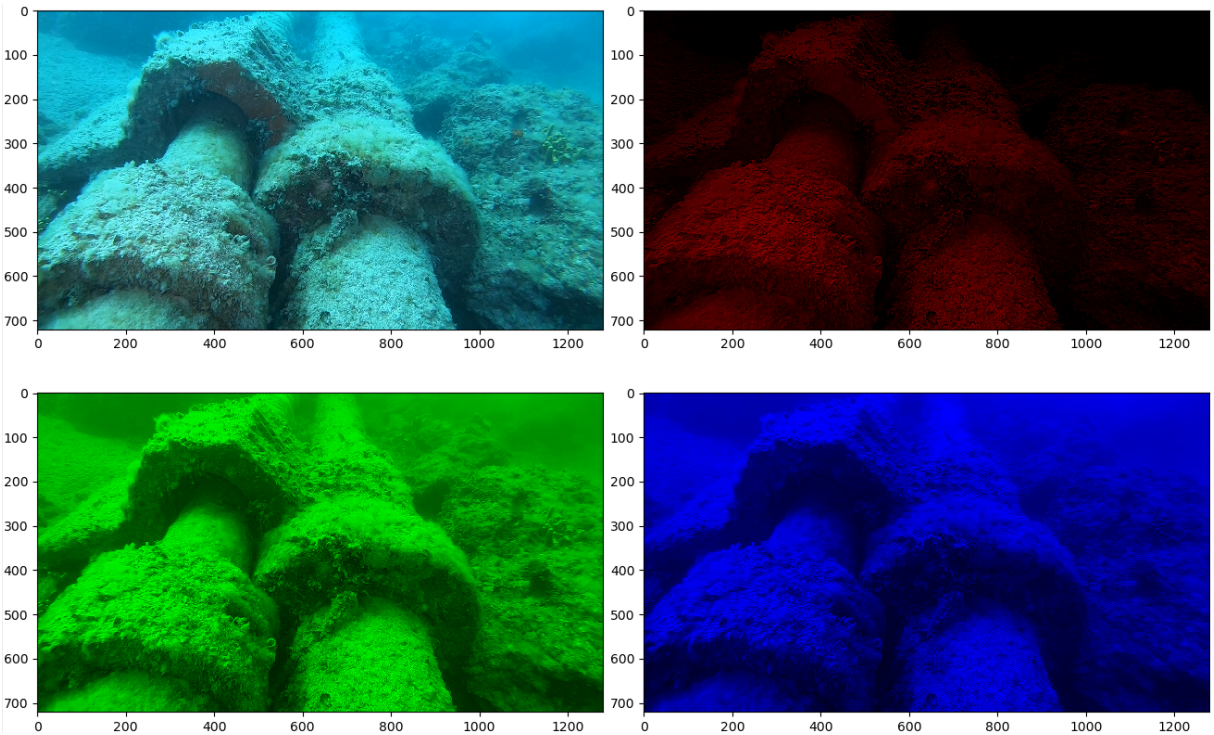
Slika 3.5: RGB kanali slike na površini

Ne zamjećujemo veliku razliku između slike 3.5 i slike 3.6 osim u što se crveni kanal počeo lagano gubiti. To je zato što se na slici 3.6 robot nalazi blizu površine. Ali, pošto svjetlo putuje duže, ne vidimo dobro na udaljenosti u crvenom kanalu.



Slika 3.6: RGB kanali slike u moru bliže površini

Na slici 3.7 vidimo R, G, B kanale slike na većoj dubini. Primjećuje se da se crveni kanal jedva vidi, te što je veća udaljenost od kamere, to vrijednost crvenog kanala opada.

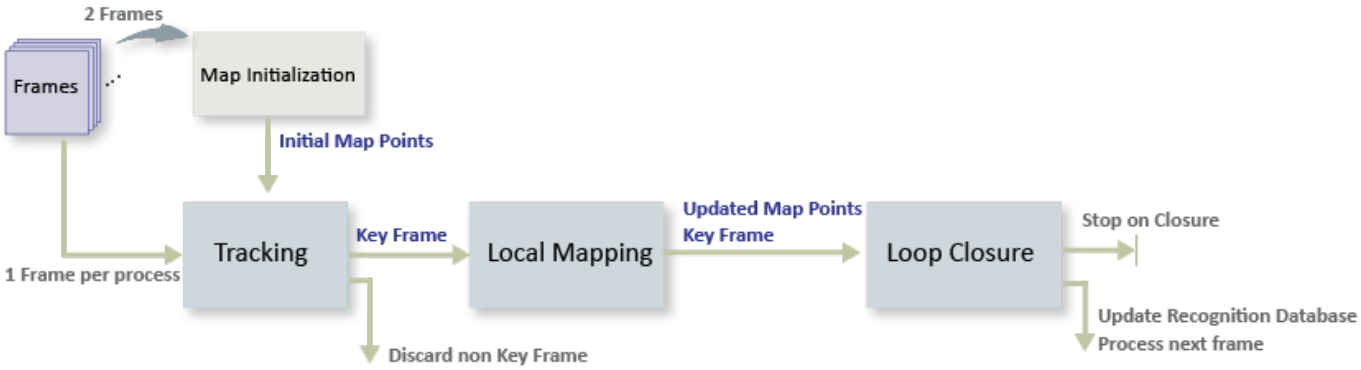


Slika 3.7: RGB kanali slike u moru na većoj dubini

Problem selektivne apsorpcije se može ublažiti korištenjem umjetnih svjetla koja nadoknađuju izgubljene boje.

3.3. Video SLAM

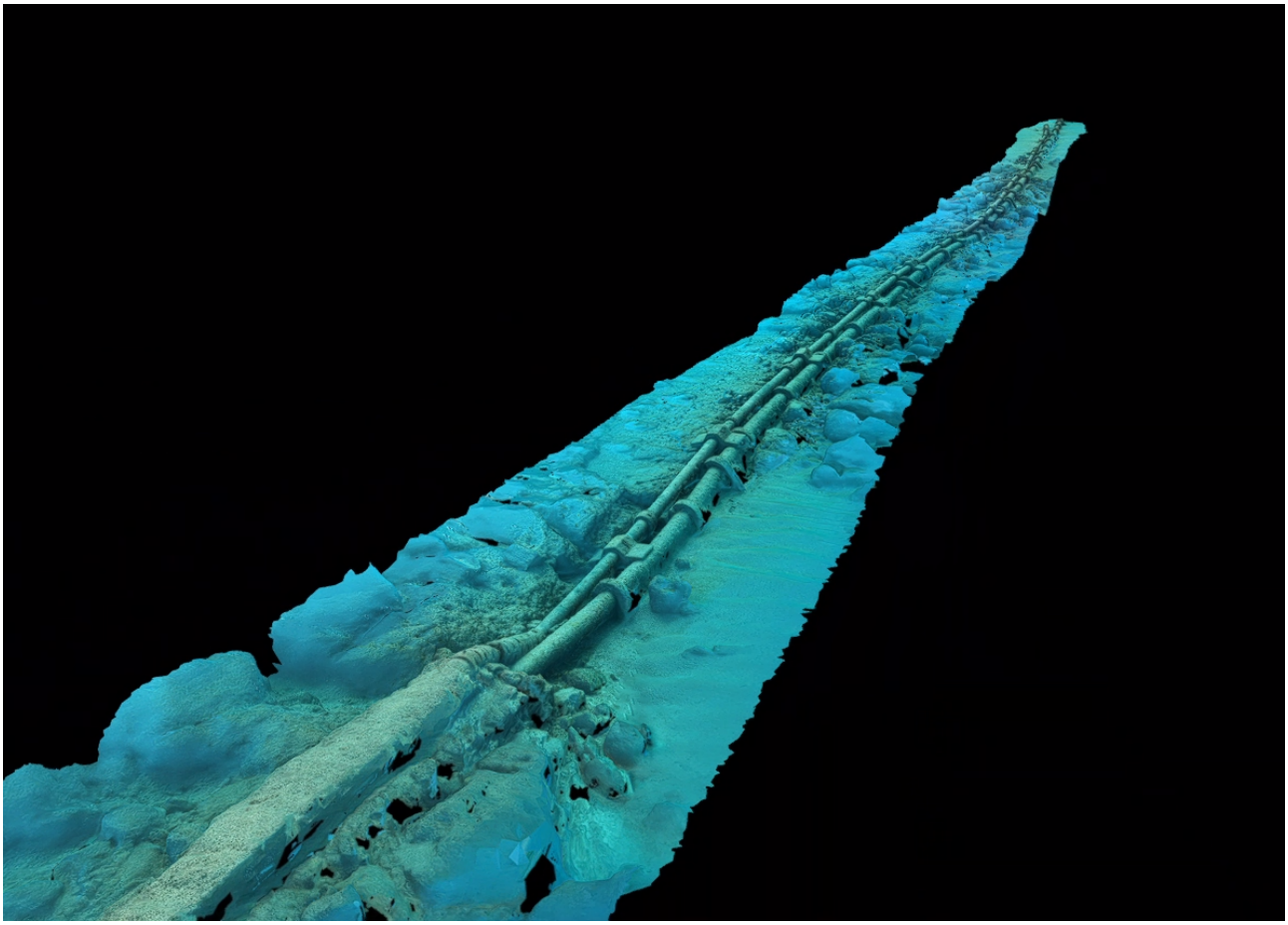
Za testiranje izgradnje podvodnih 3D modela razdvajanjem na R, G, B kanale u ovome radu, korišten je ORB-SLAM2. ORB-SLAM2 je biblioteka SLAM tehnologije u stvarnome vremenu za monokularne, stereo i RGB-D kamere. Izračunava putanju kamere i koristi rijetku (engl. *sparse*) metodu 3D rekonstrukcije [18]. ORB-SLAM2 ne podržava izradu karte s bojom (engl. *color map*).



Slika 3.8: "Pipeline" monokularnog "ORB-SLAM"-a [19]

Podaci su uključivali 3 seta snimki ispusta snimljeni sa svake od kamera ROV-a (akronim od engl. *remotely operated underwater vehicle*: daljinski upravljano podvodno vozilo). Taj ROV je na sebi imao 3 kamere: lijevu, srednju i desnu; orijentacija kamera je optimizirana za fotogrametriju.

Fotogrametrija je tehnologija pomoću koje se dobivaju pouzdane informacije fizičkih objekata i okoline kroz interpretaciju slika. Fotogrametriju možemo na neki način smatrati obrnutim od SLAM tehnologije, pošto je rad SLAM tehnologije namijenjen stvarnome vremenu, dok se fotogrametrija radi nakon što se dobiju snimke. 3D model istog ispusta dobiven fotogrametrijom možemo vidjeti na slici 3.9.

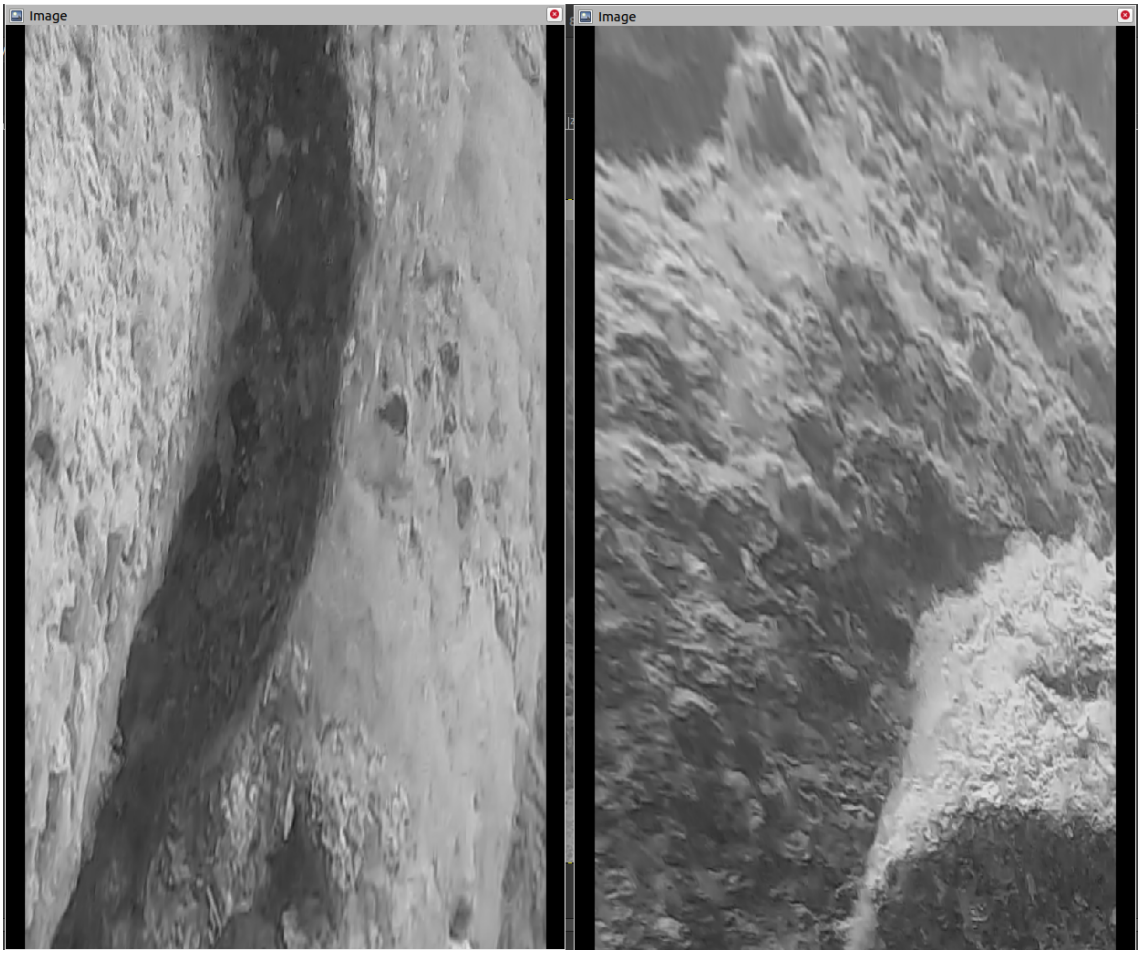


Slika 3.9: Model ispusta dobiven fotogrametrijom

Usprkos tome što je ROV opremljen s 3 kamere, za izradu podvodnih 3D modela korišten je monokularan SLAM, a ne stereo SLAM. To je zato što su orijentacije kamere optimizirane za fotogrametriju, a ne za SLAM.

Kamere su pod relativno velikim kutom, što čini rektifikaciju slika nepraktičnom. Naime, pošto rektifikacija dviju slika uključuje izobličenje slika tako da izgledaju kao da su snimljene sa samo horizontalnim pomakom, veliki kut između kamera znači da će se slike više izobličiti. Što više izobličujemo slike to je sve manje područje iskoristivih piksela za izvođenje stereo SLAM-a.

Na slici 3.10 nalaze se dvije slike (lijeva i srednja kamera) koje predstavljaju primjer izobličenja slike kod stereo rektifikacije. I lijeva i desna slika pokazuju na lijevi cjevovod iz različitih kutova. Lijeva slika predstavlja lijevu kameru, dok desna slika predstavlja srednju kameru. Može se primjetiti da se na slikama nalazi malo toga zajedničkog. Cjevovod u donjem desnom kutu desne slike odgovara cjevovodu na lijevoj strani lijeve slike. Stereo SLAM bi se jedino mogao izvoditi na tim malim dijelovima slike koje se podudaraju, to jest, sami broj neizobličenih piksela koje slike dijele je vrlo malen, što čini stereo SLAM nemogućim.



Slika 3.10: Primjer izobličenja slike kod stereo rektifikacije

3.3.1. Izrada modela sa RGB snimkama

Najveća prepreka u izradi podvodnih modela su dinamična područja s malo detalja. ORB-SLAM2 koristi rijetku metodu 3D rekonstrukcije. To znači da podudaranjem značajka na slikama nalazi svoj položaj i kreira kartu okoline. Ako imamo problema s pronalaženjem značajki nećemo moći koristiti SLAM.

Na slikama 3.11 i 3.12 vidimo primjer problema. Na slici 3.11 SLAM se ne može inicijalizirati radi malenog broja podudarnih značajki između okvira. Dok se na slici 3.12 vidi mnogo (označeno zelenim točkicama) podudarnih značajki između okvira.

Slika 3.11: Primjer pokušavanja inicijalizacije SLAM-a

Važno je istaknuti da je uzrok malom broju značajki na slici 3.11 svjetlo. Naime, zaštita cjevovoda koju vidimo na slici se nalazi na malenoj dubini. To omogućava svjetlosti da, refrakcijom kroz valove, ostavlja odsjaj na zaštiti, koji vidimo na slici kao bijele linije koje se nalaze na površini zaštite cjevovoda. Pošto more nikada nije potpuno mirno, taj odsjaj na zaštiti će se nasumično "kretati", što stvara velike probleme u prepoznavanju značajki.

Nažalost ne postoji puno što se može napraviti u ovakvoj situaciji. Kada se SLAM koristi za mapiranje vanjskih ili unutrašnjih prostora, često se to radi u gradovima, uredu ili na ostalim mjestima gdje se može naći puno značajki. Isto pomaže to što većina predmeta stvorena ljudskom rukom imaju definirane oblike s kojima je lakše izvoditi SLAM. Na primjer, puno je lakše prepoznati stol, koji ima definirane rubove pomoću kojih možemo lakše naći značajke, nego cijev koja je zaobljena i bez rubova, gdje se moramo osloniti na traženje značajki na samoj površini cijevi, što je puno teže. U ovome slučaju dosta često okolina ima više detalja nego objekt koji pokušavamo snimiti (slike 3.13 i 3.14).

Slika 3.12: Primjer dijela snimke sa puno značajki

Slika 3.13: Dio modela kojem nedostaje dio desnog cjevovoda

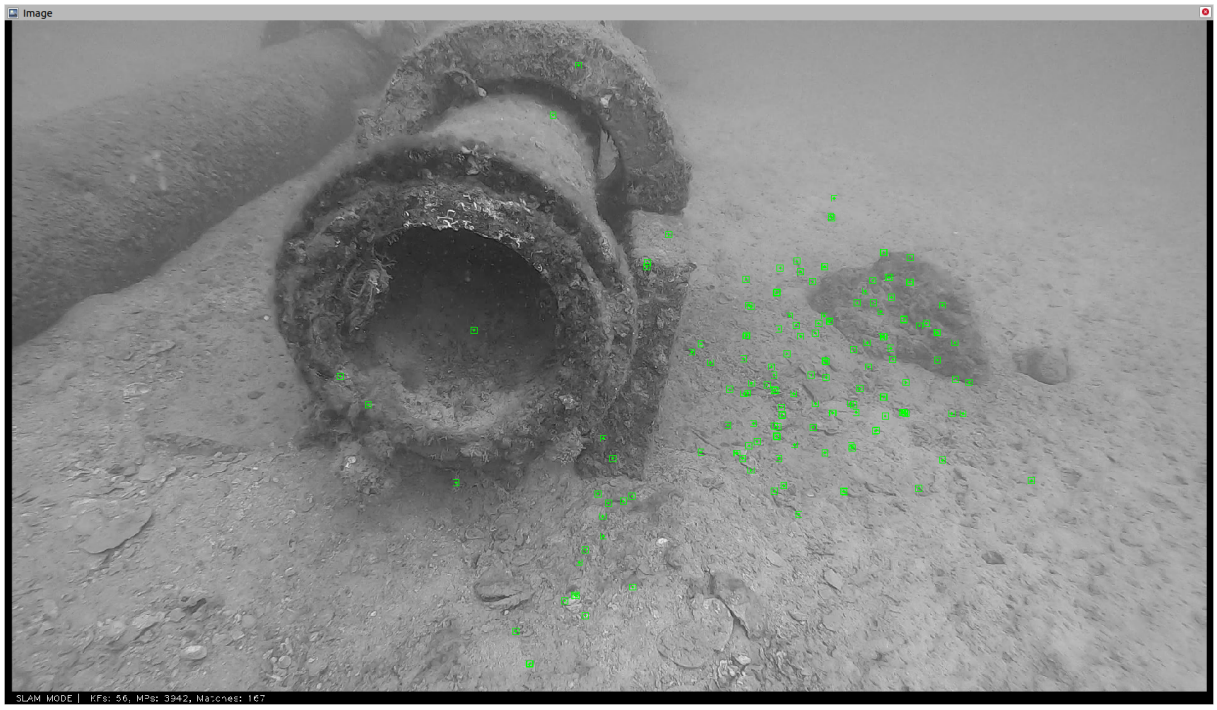
Na slici 3.13 može se vidjeti da nedostaje dio modela, zbog malog broja podudarnih značajki ORB-SLAM2 ni "ne vidi" da taj dio okoline postoji. Postoji znatna mogućnost gubitka lokalizacije u ovakvim slučajevima, a pošto je relokalizacija nemoguća jer ne prolazimo kraj istog orijentira, mapi-

Slika 3.14: Primjer: više značajki u okolini nego na objektu kojeg želimo modelirati

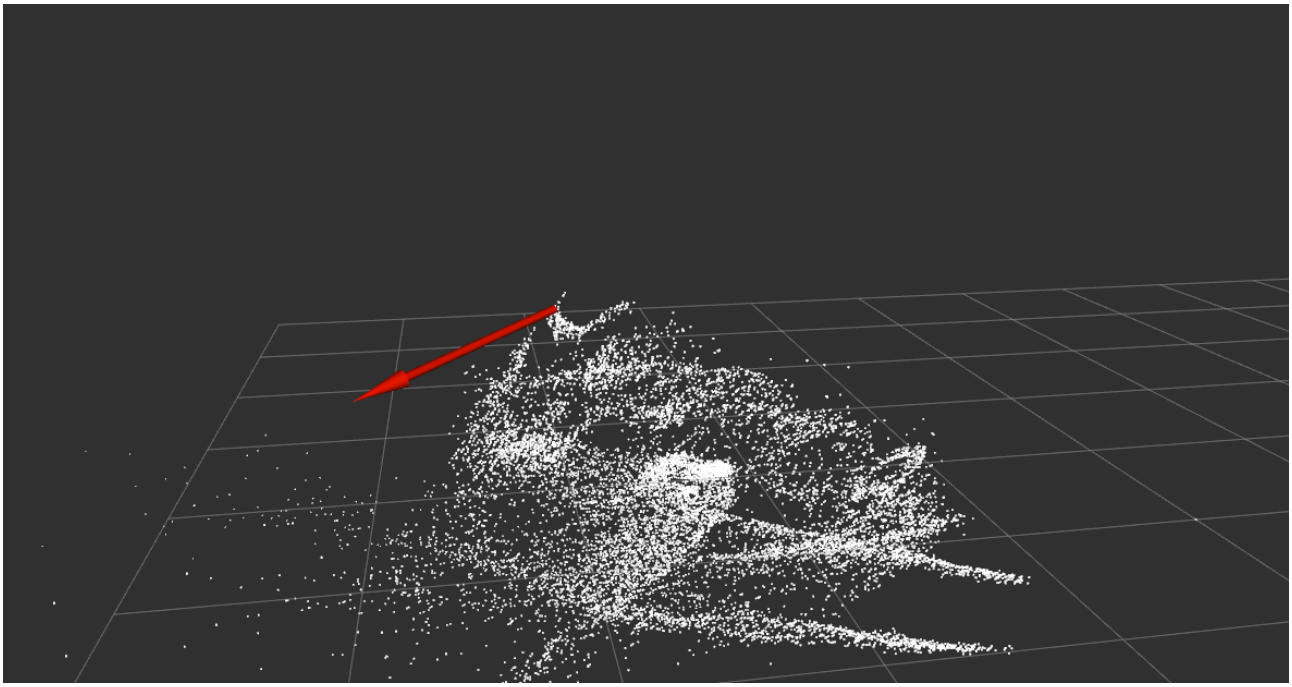
ranje se mora započeti iznova.

Zatvaranje petlje isto predstavlja prepreku. Na polovici snimke, kada ROV dođe do kraja ispusta, vraća se istim putem kojim je i došao. ORB-SLAM2 bi trebao prepoznati okret kamere, ali ne može. Zbog već spomenutog malog broja prepoznatljivih značajki (slika 3.15), prebrzo se pronalaze nove i gube starije značajke.

To dovodi do problema u lokalizaciji, ORB-SLAM2 ne može procijeniti svoj položaj, što rezultira pogrešnom mapiranju okoline. Na slici 3.16 se vidi mapiranje u ORB-SLAM2-u, dok se na slici 3.17 vidi isti taj dio ispusta modeliran fotogrametrijom.



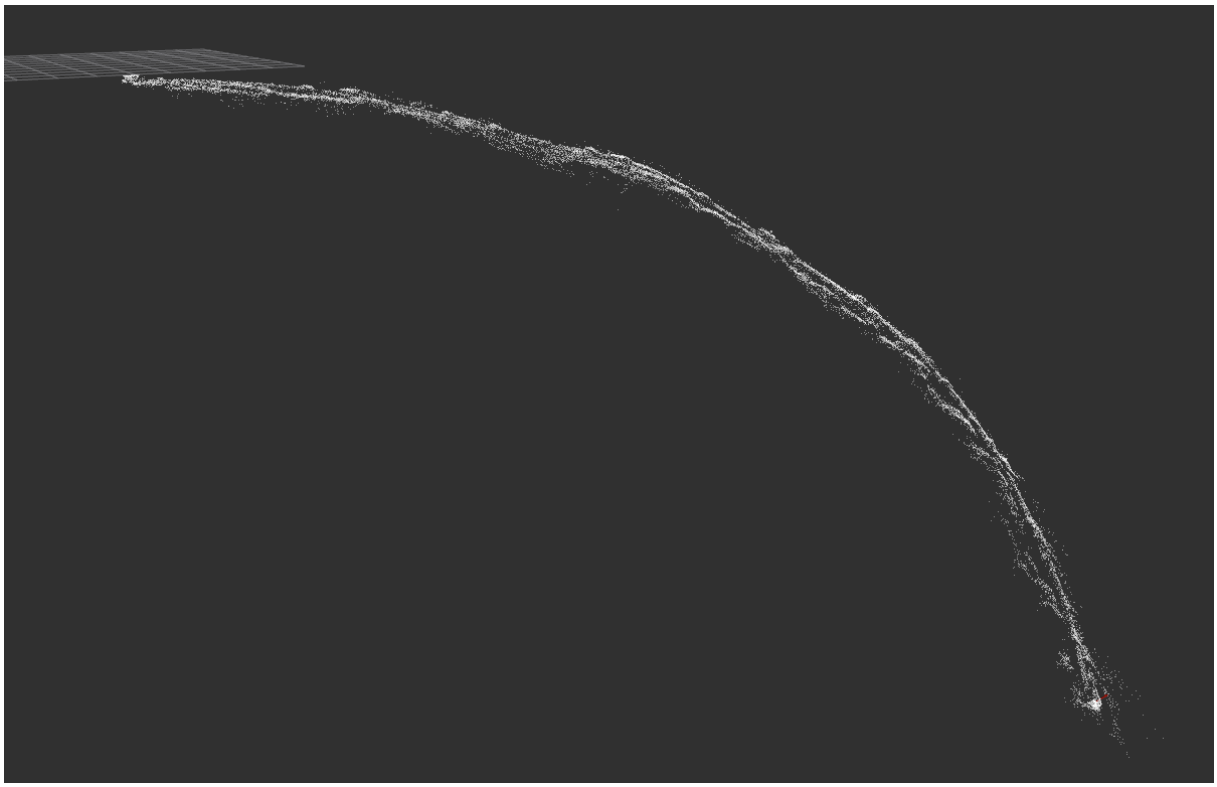
Slika 3.15: Više značajki u okolini nego na cijevi



Slika 3.16: Kriva lokalizacija pri okretu ROV-a

Slika 3.17: Točno mapiranje dijela okreta

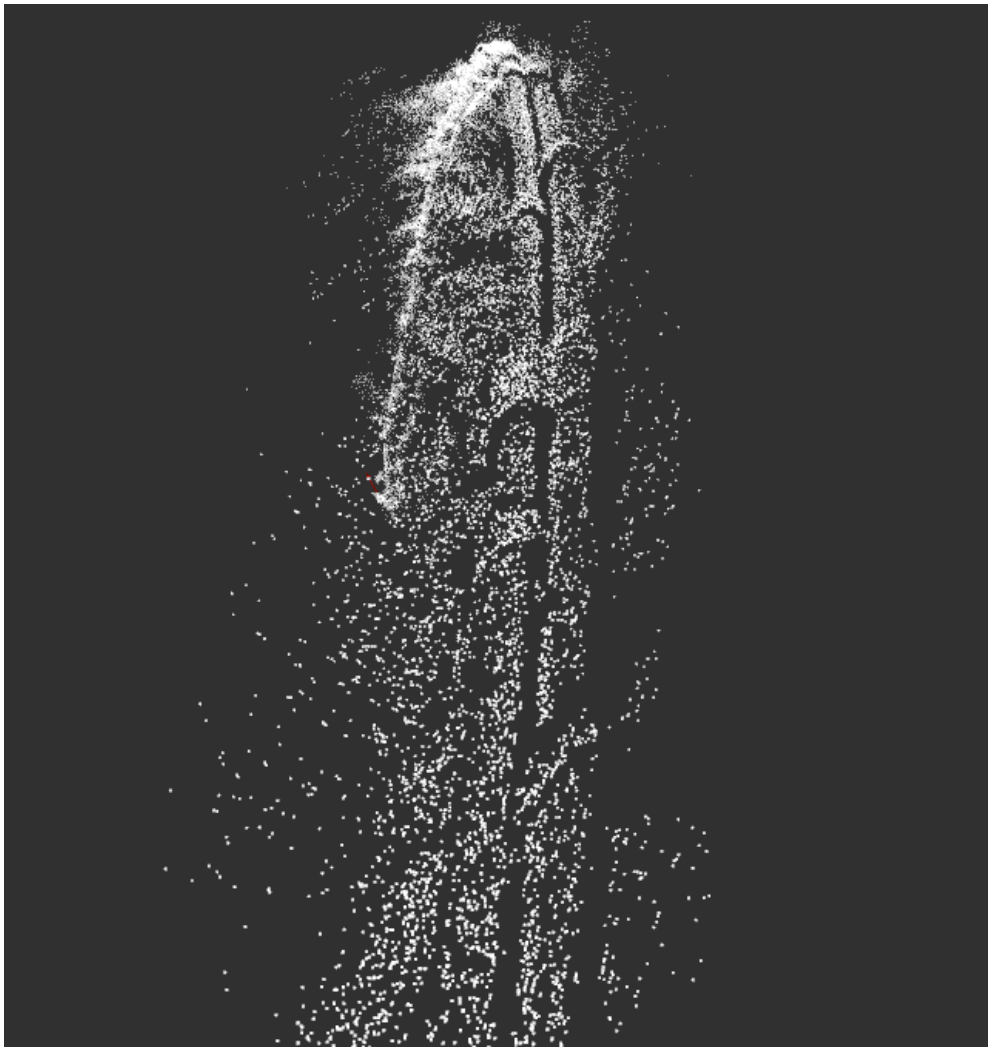
Izrada cijelog modela ispusta u oba smjera je moguća, ali zbog nemogućnosti točnog mapiranja okreta ROV-a, izgrađen je model samo od početka snimke do okreta (slika 3.18).



Slika 3.18: RGB model ispusta od početka snimke do okreta

Model na slici 3.18 se najviše razlikuje od modela na slici 3.9 u općenitom obliku. Detalji, kao što

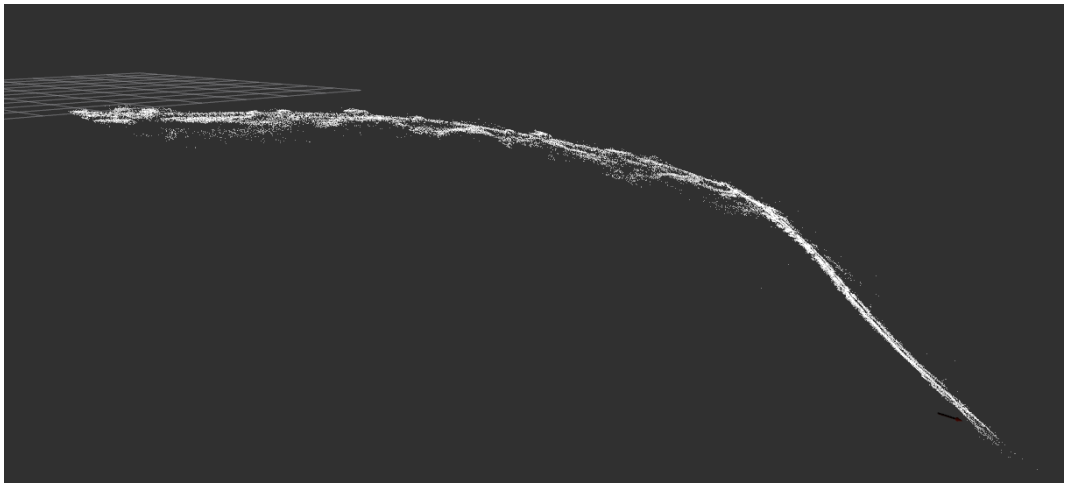
se vidi na slici 3.19, unutar su očekivanja za monokularan video SLAM.



Slika 3.19: Dio RGB modela izbliza

Model na slici 3.9 je skoro potpuno ravan, dok je model na slici 3.18 zakrivljen prema dolje. Same greške nastaju nestabilnim podudaranjem značajki između okvira. Ako se značaje puno mijenjaju, SLAM pretpostavlja da se kamera kreće, dok se u stvarnosti kamera nije pomakla, već su značajke nestale. Te greške u lokalizaciji se gomilaju kroz vrijeme, što uzrokuje zakrivljenost modela.

Snimka je snimljena u 120 slika po sekundi, a to je prebrzo za obradu u stvarnome vremenu. ROS paketom "video_stream_opencv" mogu se puštati snimke s različitim vrijednostima slika po sekundi (engl. *frames per second*). To znači da ako je snimka snimljena u 120 slika po sekundi i traje 20 minuta, ako puštamo snimku u 60 slika po sekundi, snimka će biti usporena i trajat će 40 minuta. Testirano je građenje modela na više različitih postavki i što je usporenija snimka to je pomak izraženiji. Model na slici 3.18 građen je sa snimkom od 45 slika po sekundi, dok je model na slici 3.20 građen sa 60 slika po sekundi.



Slika 3.20: Model izgrađen sa snimkom od 60 slika po sekundi

3.3.2. Izrada modela razdvajanjem slika na R, G, B kanale

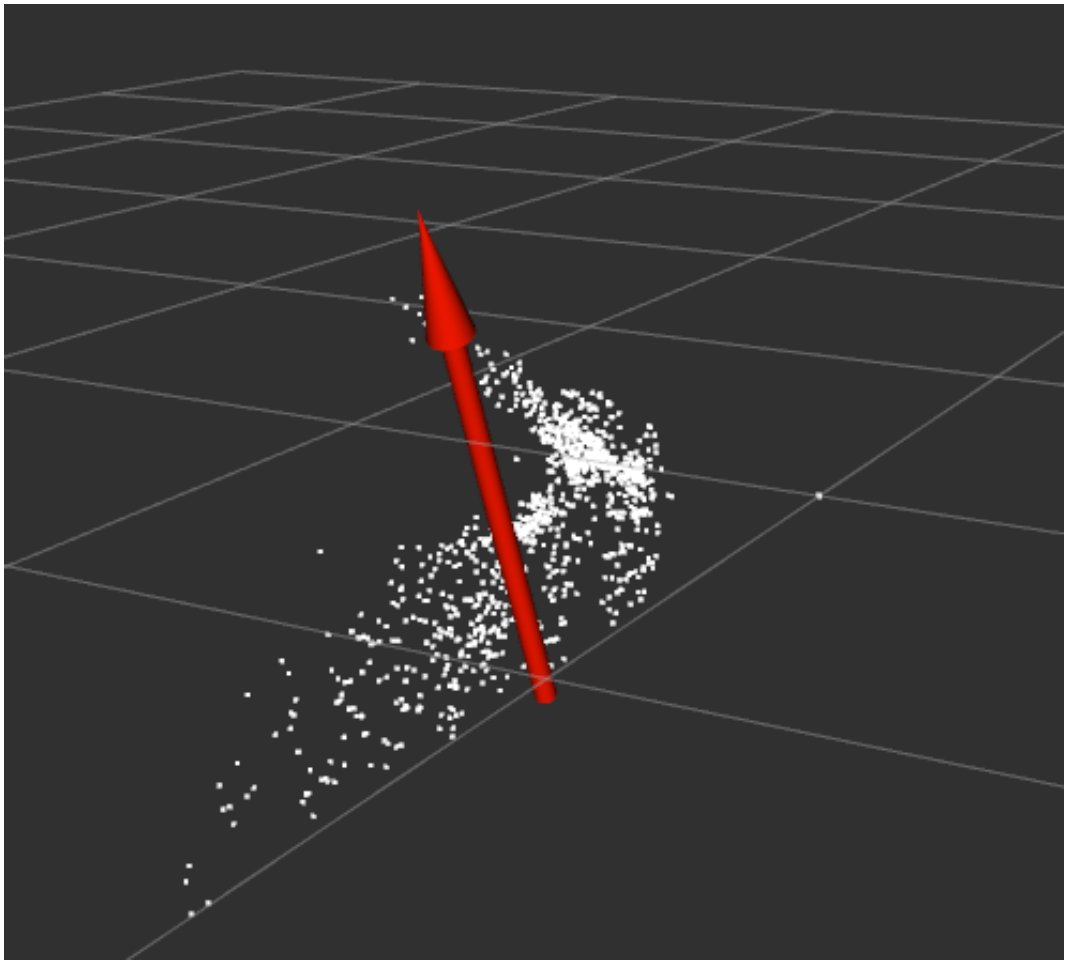
Za testiranje izrade modela razdvajanjem slika na R, G, B kanale, originalna snimka je pretvorena u 3 snimke nijansi sive (engl. *grayscale*), gdje "grayscale" vrijednost piksela na svakoj snimci odgovara vrijednosti R, G ili B kanala. Kao što je već spomenuto, crvena boja prva nestaje pod morem, a zatim zelena, i na kraju plava. Testiranje izrade modela će se zato raditi tim redoslijedom.

Crvena boja nestaje već nakon 5 metara. Na slici 3.21 možemo vidjeti taj gubitak. U blizini kamere još možemo vidjeti, ali u daljinu se sve teže vidi. To je zato što svjetlo putuje duže, pa se boja ipak gubi.

Slika 3.21: "Grayscale" slika R kanala pod morem

Tamna slika odmaže u izgradnji modela. Podudaranje značajki je puno teže sa ovakvim podacima, jer se efektivno koristi samo donja polovica slike koja je vidljiva. Sama inicijalizacija karte "ORB-SLAM2"-a se teško postiže zbog malo pronalazivih značajki. Te pronađene značajke su nestabilne što je uzrok krivoj lokalizaciji R kanala. Zbog premalo podudarnih značajki između okvira gubitak lokalizacije je neizbježan.

Na slici 3.22 se vidi primjer loše lokalizacije tijekom korištenja izdvojenoga R kanala (trebao bi se vidjeti oblik modela kao na slici 3.19), uzrokovane nestabilnim podudarnim značajkama među okvirima. ORB-SLAM2 misli da se kamera kreće u raznim smjerovima, no ta kretnja je samo iluzija, od trenutka kada se dogodi velika pogreška u lokalizaciji, mapiranje postaje nemoguće.



Slika 3.22: Kriva lokalizacija pri korištenju R kanala

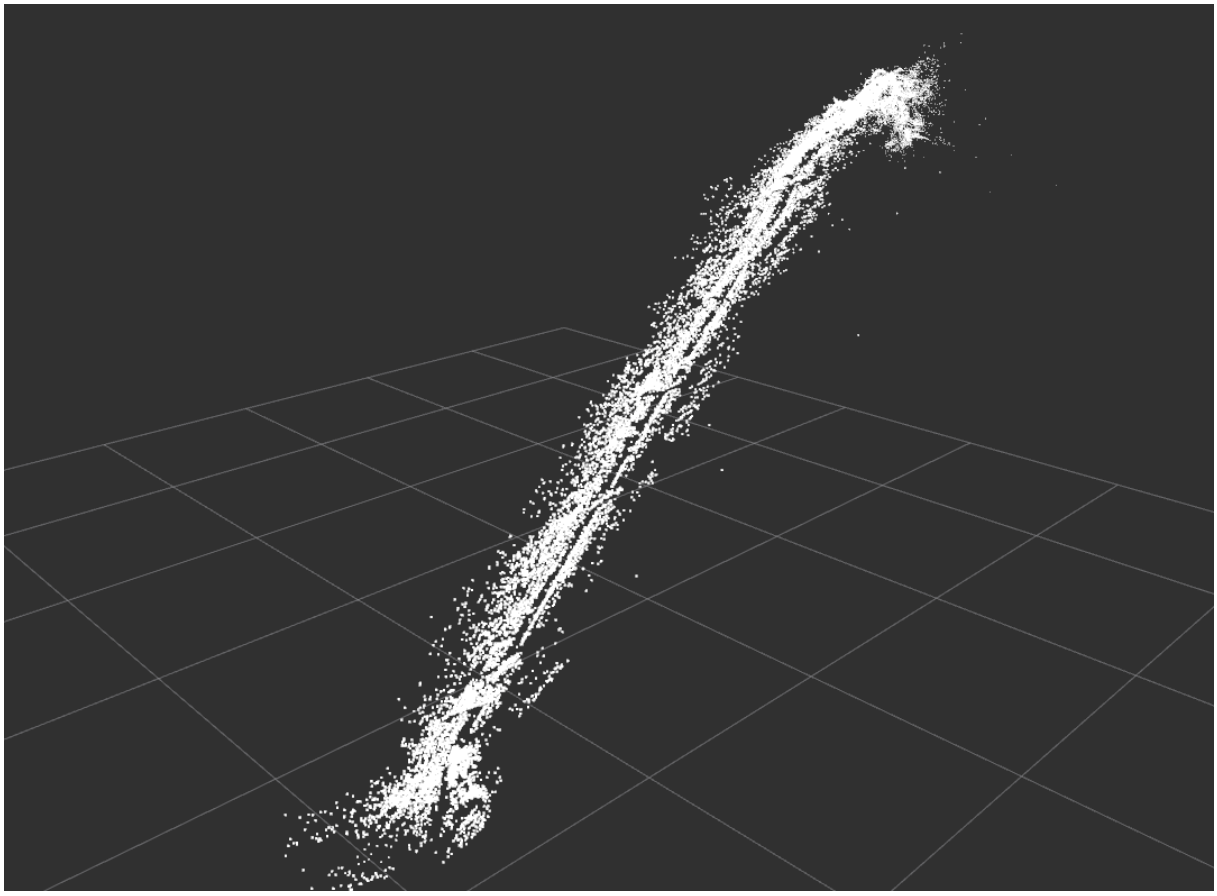
Kao što je i očekivano, izdvojeni crveni kanal se ne može koristiti u izgradnji podmornih modela. Gubitak crvenog svjetla je toliki da je teško uopće inicijalizirati kartu video SLAM-a, a kamo li izgraditi smislen 3D model.

Apsorpcija zelene boje puno je manja od crvene (slika 3.23). Odmah se primjećuje razlika, lakše se vidi slika i, što je još važnije, slika ima više detalja, što nam povećava mogućnosti u izgradnji 3D modela.

Slika 3.23: "Grayscale" slika G kanala pod morem

Zeleni kanal je puno pogodniji za SLAM nego crveni kanal, no i dalje postoji problem nestabilnih podudarnih značajki između okvira, kao što možemo vidjeti na slici 3.24. Na dnu slike vidi se problematičan početak, koji se često javlja u izradi modela G kanalom. Naime, pošto je inicijalizacija SLAM-a teža zbog manje detalja koji se nalaze u zelenome kanalu, što se događa je da se kod inicijalizacije prepozna puno nestabilnih značajki odjedanput. Te značajke se brzo izgube što stvara iluziju kretnje kamere.

Isti problem se javljao kod izrade modela pomoću R kanala, samo što je količina detalja istog puno manja nego kod G kanala. Dok se u crvenom kanalu izgubila lokalizacija, u zelenom se uspijeva oporaviti, kao što se i vidi na slici. No G kanal idalje ima malo detalja, što ubrzava gomilanje pogreški lokalizacije koja se potpuno izgubi nakon nekog vremena.



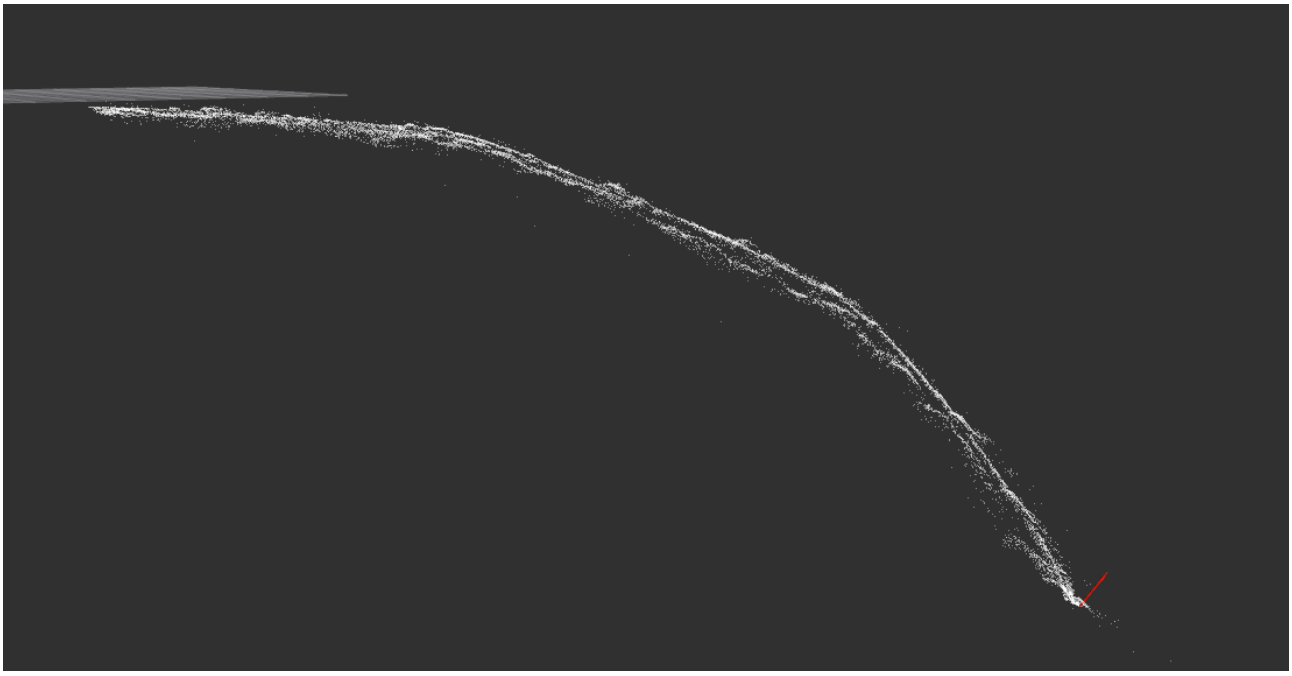
Slika 3.24: Primjer loše lokalizacije kod izrade modela G kanala

Velika nestabilnost podudarnih značajki između okvira znači da se greške u lokalizaciji puno brže gomilaju, ne može se doći do kraja ispusta jer se lokalizacija izgubi. Nije moguće izgraditi potpuni model ispusta koristeći G kanal u ovome slučaju, no u boljim uvjetima zeleni kanal bi se mogao koristiti.

Plavi kanal je najizraženiji kanal pod vodom (slika 3.25). Na prvi pogled, ne zamjećuje se razlika između slika B i G kanala, osim u tome što se kod B kanala teže vidi na udaljenost. No razlika postoji, i nalazi se u detaljima kojih plavi kanal sadrži više.

Kao i u cijeloj RGB slici najveći problem stvaraju nestabilne podudarne značajke. Iako problem nije toliko uočljiv kao kod crvenog ili zelenog kanala. Usprkos tome na slici 3.26 vidimo da je moguće izraditi veći dio modela ispusta (u daljnjem tekstu: B model).

Slika 3.25: "Grayscale" slika B kanala pod morem



Slika 3.26: B model ispusta izgrađen snimkom od 60 slika po sekundi

B model je građen snimkom od 60 slika po sekundi, ali usprkos tome vidimo veliku zakrivljenost na dole kao i kod RGB modela na slici 3.18 koji je građen na 45 slika po sekundi. Zato što SLAM radimo samo na izdvojenome plavom kanalu, u teoriji radimo na slici sa samo trećinom detalja. U stvarnosti, zato što je crveno svjetlo toliko slabije pod vodom, zelena i plava boja čine većinu detalja,

no i dalje ne radimo s potpunom slikom. S manje detalja greške u lokalizaciji će se brže gomilati nego na cijeloj slici, što uzrokuje većoj zakrivljenosti i što nas sprječava u izgradnji potpunog modela.

Na slici 3.27 vidi se dio B modela izbliza. Zamjećuje se da je B model skoro detaljan kao i RGB model, što je i očekivano s obzirom na to da se radi samo o B kanalu.

Slika 3.27: Dio B modela iz bliza

Važno je spomenuti da je B kanal manje stabilan za izradu modela nego kada se koristi cijela slika. U dobrome dijelu slučajeva lokalizacija će se ili izgubiti odmah na početku, što dovodi do rezultata sličnijima crvenom ili zelenom kanalu, ili će se greške lokalizacije nagomilati prije nego što kamera dođe do kraja modela i izgubit će se lokalizacija.

Općenito, izdvojeni plavi kanal se može koristiti za izradu 3D modela u podmorju. Najveća razlika izrade modela između izdvojenog plavog kanala i cijele RGB slike je konzistencija. Pošto plavi kanal ne može imati istu količinu detalja kao što i puna RGB slika, greške u lokalizaciji će se brže gomilati.

Naravno, izrada 3D modela razdvajanjem slika na R, G, B kanale se može koristiti samo na određenim dubinama. Kamera se mora nalaziti na tolikoj dubini da ovisi ponajviše o suncu za osvjetljavanje

područja koji će se mapirati. Ako smo preblizu površine crveno svjetlo će idalje činiti dobar dio informacija cijele slike, što znači da ćemo raditi sa efektivno trećinom podataka, što nije dovoljno za uspješno izvođenje SLAM-a. Isto tako, kako se kamera spušta na sve veće i veće dubine, gubi se prirodno svjetlo, te se moramo sve više oslonjavati na umjetno svjetlo, koje možemo podesiti tako da kompenzira za gubitak boja, te nema potrebe da razdvajamo slike na zasebne kanale.

4. Zaključak

SLAM je jako kompleksna i svestrana tehnologija. Može se koristiti uz pomoć praktično bilo kojeg senzora daljine ili slike. Od svih vrsta SLAM-a, video SLAM je najfleksibilnija tehnologija, zato što je najbliža načinu kojim se ljudi orijentiraju u prostoru.

Video SLAM u podmorju je općenito puno teži za izvesti nego na površini. Podmorje ima slabije definirane objekte nego površina, gdje se video SLAM često testira u okolinama ureda ili ulica gdje su svi objekti statički i dobro definiranih rubova. Daleko najveći nedostatak korištenja video SLAM tehnologije u podmorju je to što je okolina dinamična, a najveći problem predstavljaju alge, koje mijenjaju svoj oblik s morskom strujom. Kao što se vidjelo u radu, to je uzrok nestabilnim značajkama koje su uvijek stvarale problem u lokalizaciji video SLAM-a.

Rezultati izrade 3D modela razdvajanjem slika na R, G, B kanale su očekivani. Zbog selektivne apsorpcije mora crvena boja se gubi prva (već na 5 metara), i kao takva ne može se koristiti za izradu 3D modela. I zeleni i plavi kanali se mogu koristiti u izradi, no plavi se preferira. Zbog količine plave boje pod morem, najviše detalja slike se nalazi u tom kanalu. U usporedbi s punom RGB slikom, najveći nedostatak plavog kanala leži u konzistenciji. Sami plavi kanal ne može imati detalja koliko i puna RGB slika, što odmaže lokalizaciji.

5. Literatura

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," in IEEE Robotics & Automation Magazine, vol. 13, no. 2, pp. 99-110, June 2006, doi: 10.1109/MRA.2006.1638022.

[2] "SLAM (Simultaneous Localization and Mapping)", s Interneta, https://www.mathworks.com/discovery/slam.html

[3] "Design and use Kalman filters in MATLAB and Simulink", s Interneta, https://www.mathworks.com/discovery/kalman-filter.html

[4] Ribeiro, Maria Isabel. "Kalman and extended kalman filters: Concept, derivation and properties." Institute for Systems and Robotics 43 (2004): 46.

[5] Sobczak, Ł.; Filus, K.; Domański, A.; Domańska, J. LiDAR Point Cloud Generation for SLAM Algorithm Evaluation. Sensors 2021, 21, 3313. https://doi.org/10.3390/s21103313

[6] "What Is Camera Calibration?", s Interneta, https://www.mathworks.com/help/vision/ug/camera-calibration.html

[7] "Camera Calibration and 3D reconstruction", s Interneta, https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

[8] Maimone, M.; Cheng, Y.; Matthies, L. (2007). "Two years of Visual Odometry on the Mars Exploration Rovers" (PDF). Journal of Field Robotics. 24 (3): 169–186. CiteSeerX 10.1.1.104.3110. doi:10.1002/rob.20184. Retrieved 2008-07-10.

[9] He, Ming & Zhu, Chaozheng & Huang, Qian & Ren, Baosen & Liu, Jintao. (2020). A review of monocular visual odometry. The Visual Computer. 36. 10.1007/s00371-019-01714-6.

[10] Turan, Mehmet & Almalıoğlu, Yasin & Araujo, Helder & Konukoglu, Ender & Sitti, Metin. (2017). Deep EndoVO: A Recurrent Convolutional Neural Network (RCNN) based Visual Odometry Approach for Endoscopic Capsule Robots. Neurocomputing. 275. 10.1016/j.neucom.2017.10.014.

[11] Gu, Feifei & Song, Zhan & Zhao, Zilong. (2020). Single-Shot Structured Light Sensor for 3D Dense and Dynamic Reconstruction. Sensors. 20. 1094. 10.3390/s20041094.

[12] "Why ROS", s Interneta, https://www.ros.org/blog/why-ros/

[13] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009, May). ROS: an open-source Robot Operating System. In ICRA workshop on open source software (Vol. 3, No. 3.2, p. 5).

[14] Barkai, David. "An introduction to peer-to-peer computing." Intel Developer update magazine (2000): 1-7.

[15] "rqt graph - Visualise and Debug Your ROS Graph", s Interneta, https://roboticsbackend.com/rqt-graph-visualize-and-debug-your-ros-graph/

[16] "The Science of Color Underwater", s Interneta, https://connieimboden.com/2013/12/the-science-of-color-underwater-2/

[17] R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," in IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, Oct. 2015, doi: 10.1109/TRO.2015.2463671.

[18] Mur-Artal, Raul, and Juan D. Tardós. "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras." IEEE transactions on robotics 33.5 (2017): 1255-1262.

[19] "Monocular Visual Simultaneous Localization and Mapping", s Interneta, https://www.mathworks.com/help/vision/ug/monocular-visual-simultaneous-localization-and-mapping.html

6. Sažetak i ključne riječi na hrvatskom i engleskom jeziku

6.1. Hrvatski

Korištenje SLAM tehnologije u podmorju podrazumijeva izazove zbog dinamičnosti podvodnih okolina. Zbog selektivne apsorpcije svjetla u moru, na većim dubinama se vidi sve manje boja. Cilj ovog rada je proučiti izradu podvodnih 3D modela video SLAM tehnologijom razdvajanjem slike na R, G, B kanale. Osim proučavanja izgradnje 3D modela, potrebno je opisati glavne značcajke video SLAM tehnologije i prednosti i mane korištenja iste u podvodnim okruženjima. Zbog selektivne apsorpcije mora prva se gubi crvena boja (već na 5 metara dubine) i, kao takva, ne može se koristiti za izradu 3D modela. Dakle, mogu se koristiti zeleni i plavi kanal, no plavi se obično preferira. Zbog količine plave boje pod morem, najviše detalja slike se nalazi upravo u tom kanalu. U usporedbi s punom RGB slikom, najveći nedostatak plavog kanala leži u konzistenciji. Sami plavi kanal ne može imati detalja koliko i puna RGB slika, što odmaže lokalizaciji.

Ključne riječi: SLAM, video SLAM, SLAM u podmorju

6.2. Engleski

The usage of SLAM technology in underwater environments presents unique challenges due to its dynamic nature. Because of the selective absorption of light by the sea, greater depths equate to a further loss of color. This thesis aims to study the creation of underwater 3D models using visual SLAM technology by splitting the image into separate R, G, B channels. In addition to studying the construction of 3D models, it is necessary to describe the main features of video SLAM technology and the advantages and disadvantages of using it in underwater environments. Due to the selective absorption of the sea, the red color is lost first (already at 5 meters), and as such, it cannot be used for creating 3D models. Both green and blue channels can be used, but blue is preferred. Because of the amount of blue colors under the sea, most details of the image are found in that channel. Compared to a full RGB image, the biggest drawback of the blue channel lies in consistency. The blue channel alone cannot achieve as much detail as a full RGB image, which corrupts the localization.

Key words: SLAM, visual SLAM, SLAM in underwater environments