Gaussove trodimenzionalne reprezentacije za istovremenu lokalizaciju i mapiranje vizualnim senzorima

Sladić, Ante

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 462

GAUSSOVE TRODIMENZIONALNE REPREZENTACIJE ZA ISTOVREMENU LOKALIZACIJU I MAPIRANJE VIZUALNIM SENZORIMA

Ante Sladić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 462

GAUSSOVE TRODIMENZIONALNE REPREZENTACIJE ZA ISTOVREMENU LOKALIZACIJU I MAPIRANJE VIZUALNIM SENZORIMA

Ante Sladić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 462

Pristupnik:	**Ante Sladić (0036527330)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	prof. dr. sc. Ivan Marković

Zadatak: **Gaussove trodimenzionalne reprezentacije za istovremenu lokalizaciju i mapiranje vizualnim senzorima**

Opis zadatka:

Autonomna navigacija u nepoznatim prostorima ključna je za primjenu mobilnih robota u svakodnevnim problemima. Istovremena lokalizacija i mapiranje vizualnim senzorima zanimljiv je pristup rješavanju tog problema zbog jednostavnosti senzorskog sustava i visoke razine semantičke informacije. Glavni nedostatak tradicionalnih algoritama je nemogućnost guste fotorealistične rekonstrukcije, uglavnom zbog memorijskih i računskih ograničenja odabrane reprezentacije prostora. Nedavno uvođenje Gaussove trodimenzionalne reprezentacije dovelo je do drastičnog napretka u sintezi novih pogleda u trodimenzionalnoj sceni. Cilj ovoga diplomskog rada je istražiti prednosti Gaussove trodimenzionalne reprezentacije prostora za istovremenu lokalizaciju i mapiranje te potencijalna proširenja, uključujući semantičko mapiranje te fuziju s drugim senzorima. Implementirani algoritam potrebno je testirati u simulacijskom i po mogućnosti stvarnom okruženju, uz kvalitativnu i kvantitativnu usporedbu s metodama stanja tehnike.

Rok za predaju rada: 28. lipnja 2024.

Sadržaj

1. Uvod

SLAM sustavi su jedna od temeljnih tehnologija u robotici i autonomnim sustavima koja omogućava stvaranje mape prostora u kojem se robot nalazi dok ujedno prati i njegovu lokaciju. Tradicionalni SLAM sustavi koji su implementirali Kalman filtere te posljednje optimizacijske metode temeljene na teoriji grafova uvelike su napredovali, no i dalje SLAM sustavi imaju problema s efikasnosti, prilagodljivosti i skalabilnosti. Uz nabrojene nedostatke, još jedan dugostojeći problem u području je fotorealistična rekonstrukcija u stvarnom vremenu, gdje su dosadašnji sustavi uvijek morali birati između brzine izvođenja odnosno efikasnosti ili fotorealističnosti scene odnosno kvaliteti.

Gaussove trodimenzionalne reprezentacije su posljednji trend u području rekonstrukcije scene. Glavne prednosti metode su njena fotorealističnost koja konkurira sustavima sa neuronskim radijalnim poljima i brzina izvođenja koja nadmašuje sve prethodne sustave. Gaussove trodimenzionalne reprezentacije prikazuju scenu kao skupinu točaka odnosno elipsoida koji se podvrgavaju Gaussovoj funkciji, što osigurava kontinuiranost i diferencijabilnost reprezentacije. Naspram eksplicitnog oblika i prijašnje spomenutih svojstva, prednost metode je modeliranje nesigurnosti odnosno šuma koristeći probabilističko svojstvo Gaussove distribucije čime se osigurava robustnost sveukupnog sustava.

Ovaj rad istražuje integraciju Gaussovih trodimenzionalnih reprezentacija unutar SLAM radnog okvira. Za početak će se dati uvod o SLAM sustavima, njihovoj arhitekturi, posljednjim trendovima i mjestima primjene. Prikazat će se neuronska radijalna polja (NeRF), njihova teorijska pozadina i neki od modernijih sustava, kao motivacija za Gaussovim trodimenziolanim reprezentacijama, te onda same trodimenzionalne Gaussove reprezentacije. Naposljetku prikazat ćemo implementaciju SLAM sustava sa Gaussovim trodimenzionalnim reprezentacijama i rezultate dobivene u eksprimentima.

2. SLAM

U području automatizacije robota, simultana lokalizacija i mapiranje (skr. SLAM) jedna je od ključnih tehnologija koja omogućava robotu kretanje u prijašnje nepoznatim prostorima. Zadaća SLAM-a je stvaranje mape prostora unutar kojeg se robot nalazi te određivanje pozicije robota unutar tog prostora, pritom prikupljajući podatke sa vanjskih senzora poput kamera ili radara. Inkrementalna konstrukcija mape i kontinuirano praćenje lokacije robota su temelji percepcije okruženja robota i automatizacije[1]. U usporedbi s radarom, sonarom i drugim prostornim senzorima, vizualni senzori imaju prednost što su manjeg volumena, manje potrošnje i podatci sadržavaju veliku količinu informacija. No, vizualni SLAM ima i svoje nedostatke. Degradacija slike uzrokovana šumom, nagle promjene osvjetljenja ili nagli pokreti samo su neki od razloga zašto su za vizualni SLAM potrebni veoma složeni sustavi. Uz napredak tehnologije u području računalnog vida, ponajviše u promjenjenoj odnosno umjetnoj realnosti i autonomnosti robota u unutrašnjim prostorima, vidimo i sve veću zainteresiranost za vizualni SLAM[2].

2.1. Podjela po ulaznim senzorima

Kad pričamo o vizualnom SLAM-u, u većini slučajeva mislimo na SLAM gdje su ulazni senzori kamere. No postoji više vrsta kamera i načina na koji se one mogu koristiti. SLAM sustav koji koristi jednu kameru zove se Monokularni SLAM[3]. Senzori koji su sačinjeni od samo jedne kamere su jeftini i jednostavni za implementaciju i integraciju, što ih čini vrlo zanimljivima istraživačima. Kamera kao tip podatka vraća sliku, a slika je u suštini dvodimenzionalna reprezentacija trodimenzionalnog prostora. Očigledno je da u tom procesu gubimo jednu dimenziju, u ovom slučaju dubinu. Iako ovo zvuči na prvu trivijalno, pokazat će se da je u potpunosti suprotno. Mi kao ljudi lako percipiramo dubinu iz više razloga. Kao prvo, oči možemo smatrati kao stereo kamerama, pa

samim time imamo dvije slike za percepciju umjesto samo jedne. Drugi razlog koji nam ujedno omogućava da shvatimo relativnu dubinu i sa slika je iskustvo. Kroz život, čovjek se susreće sa mnoštvom objekata i stječe saznanja o njihovim geometrijskim svojstvima (visini, širini i slično) te kad ih vidimo na slici lako možemo pretpostaviti njihov položaj u prostoru. Također su tu i neka znanja koja robotu nisu unaprijed poznata, na primjer ako je na slici Sunce ili Mjesec, čovjek zna da je to objekt koji je udaljen tisućama kilometara i nije relevantan za odnose objekata na slici, dok robot toga nije svjestan ako se eksplicitno ne nauči. No iz samo jedne slike je nemoguće odrediti stvarnu veličinu objekata, čak i za čovjeka. Za to nam je potrebna još jedna slika, u monokularnom slučaju još jedna slika sa iste kamere iz drugog kuta. Na istom principu funkcionira i Monokularni SLAM. Pomičemo kameru i kontinuirano uzimamo slike te iz njih računamo pomak same kamere i relativne odnose između objekata na slici. Intuitivno znamo da ako pomičemo kameru ulijevo, objekti na slici će se pomicati prema desno. Razlika između piksela objekata na dvije slike naziva se disparitet i na temelju tog dispariteta moguće je izračunati koji objekti su bliži od drugih. No to je i dalje samo relativan odnos objekata, ne stvarne dimenzije. Taj relativan odnos se razlikuje od stvarnog odnosa za određeni faktor tj. skalu te s obzirom na nemogućnost određivanja stvarne dubine sa monokularnom kamerom, to se također naziva "neodređenost skale" (eng. scale ambiguity). Da objasnimo na primjeru radi lakšeg shvaćanja, na slici 2.1. čovjeku je lako odrediti da je lijeva osoba bliže kameri nego desna, ali do tog zaključka dolazimo intuitivno jer znamo iz iskustva veličinu prosječnog čovjeka. Računalo nema to saznanje te iz samo ove jedne slike za njega nije moguće odrediti je li lijeva osoba u stvarnoj veličini pa desna umanjena ili je desna osoba u stvarnoj veličini, a lijeva uvećana.

Stereo senzori i dubinske kamere za razliku od monokularnih mogu izračunati odnosno izmjerit točnu udaljenost između objekata i kamere. Kad su poznate te udaljenosti odnosno dubine, moguće je konstruirati trodimenzionalnu reprezentaciju iz samo jedne slike i riješen je problem neodređenosti skale. No, iako imaju istu ulogu, stereo senzori i dubinske kamere se uvelike razlikuju. Stereo kamera se sastoji od dvije sinkronizirane monokularne kamere koje su udaljene jedna od druge za poznatu vrijednost, "baseline". Na temelju tog unaprijed poznatog baseline-a, moguće je izračunati dubinu objekta na temelju razlika između lijeve i desne slike, slično kako čovjek percipira dubinu. Dubina ovisi o baseline-u na način da što veći baseline, veća je udaljenost na kojoj stereo kamera

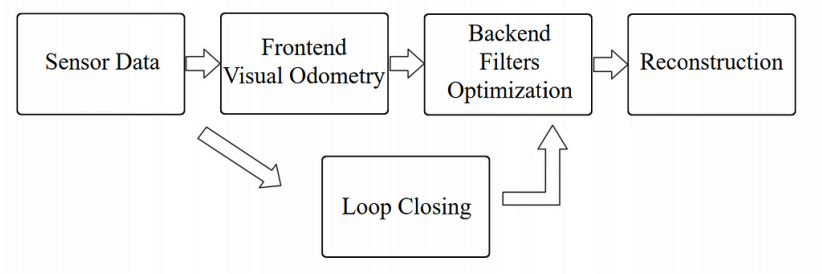
Slika 2.1. Nemoguće je na temelju jedne slike odrediti jesu li dvije osobe različito udaljene ili su dvije osobe stvarno različitih veličina. Slika preuzeta sa [4]

.

može sa određenom sigurnosti izračunati dubinu. To može biti problematično pri autonomnoj vožnji i sličnim problemima na otvorenom jer bi zahtjevalo da stereo kamere budu vrlo udaljene jedna od druge, što ne bi bilo praktično. Drugi nedostatak je računarska zahtjevnost problema. Naime, u većini slučajeva potrebni su GPU ili posebni FPGA sklopovi kako bi se izračuni mogli izvršavat u stvarnom vremenu, što stvara dodatne probleme prilikom implementacije. Nadalje, proces kalibracije i konfiguracije senzora je kompliciran te dubina i točnost izračuna su ograničeni ovisno o baseline-u, npr. nije moguće snimati scenu na udaljenosti od 100 metara ako nam je baseline 1 centimetar. Za razliku od stereo kamera, dubinske kamere nisu ograničene baseline-om. Dubinske kamere su monokularne kamere koje uz standardnu sliku koriste Time-of-Flight ili neku sličnu tehnologiju mjerenja dubine. Dakle, kod dubinskih kamera potpuno smo preskočili izračun nego dubinu dobivamo izravno od senzora. No, iako smo izbjegli problem zahtjevnog izračuna, to ne znači da dubinske kamere nemaju svojih nedostataka. Neki od njih su ograničen domet pouzdanog mjerenja, prisustvo šuma, manji kut gledišta, podležnost promjenama osvjetljenja te nemogućnost razlikovanja transparetnih materijala poput stakla. Zbog svih tih nedostataka, SLAM sa dubinskim kamerama ograničen je samo na unutarnje prostore.

2.2. Arhitektura tipičnog SLAM sustava

Sad kad smo se upoznali sa vrstama kamera koje se pretežno koriste kod vizualnog SLAM-a, vrijeme je da objasnimo tipičan SLAM sustav. *Frontend*, *backend* i *loop closure* su glavni dijelovi tipičnog SLAM cjevovoda(2.2.).



Slika 2.2. Tipičan SLAM cjevovod. Slika preuzeta iz [3].

Frontend, odnosno vizualna odometrija, sustav je čija je glavna zadaća određivanje pomaka kamere između slika te inicijalna procjena mape prostora. Na prvu problem ne izgleda kompliciran jer za čovjeka to je poprilično intutivan zadatak. Na temelju dvije slike lako je prepoznati da je jedna iz drukčijeg kuta ili sa druge lokacije, no skoro je nemoguće samo gledajući u dvije slike kvantitativno odrediti za koliko je jedna slika pomaknuta. No, računalo mora kvantitativno odrediti razlike jer tako funkcionira u suštini, ali kako to postići? Postoje dva pristupa u vizualnoj odometriji: pomoću točaka značajki (eng. *feature point*) i direktni pristup. Metode koje koriste točke značajki funkcioniraju na način da iz više slika izvuku manju skupinu specifičnih točaka i povežu ih, odnosno pronađu koresponendencije između njih. Pozicije kamere i dubine piksela procjenjuju se računajući pogrešku reprojekcije (eng. *reprojection error*) tih označenih parova točaka. Moderni deskriptori značajki (Harris [5], SURF [6], ORB [7]) su se pokazali robustinijima nego direktne metode, ponajviše zbog svoje otpornosti odnosno invarijatnosti na promjene u osvjetljenju i točki gledišta [8]. U drugu ruku pokazali su se nepouzdanima u jednoličnim scenama gdje dolazi do gubitka praćenja putanje unatoč nedostatku točaka značajki te su ograničeni što se tiče broja točaka koje se mogu uzimat u obzir prilikom izračuna, a da bi se izračun izvršio u stvarnom vremenu. No, za razliku od metoda točaka značajki, direktne metode ne uparuju točke između više slika i računaju pogrešku reprojekcije nego koriste sve piksele slike [9], piksele sa dovoljno velikim intezitetom gradijenta [10] ili rijetko izabrane piksele [11](ovisno o metodi) te minimiziraju fotometrični gubitak. Pozicija kamere i dubine piksela se procjenjuju mi-

nimizirajući taj gubitak pomoću nelinearnih optimizatora. Kako se koristi više piksela nego kod metoda točaka značajki, direktne metode pokazale su se robustnijima u scenama s manje teksture te mogu pružiti puno gušću odnosno detaljniju trodimenzionalnu strukturu prostora. No, zbog korištenja većeg broja piksela, direktne metode sklonije su većim pogreškama u prisustvu artefakata koji nastaju zbog *rolling shutter*-a, automatske ekspozicije kamere i slično. Također, još bitnije od dosad navedenih nedostaka, direktne metode pretpostavljaju konstantno osvijetljenje u sceni te pri naglim promjenama osvijetljenja performanse ovakvih metoda uvelike opadaju. Zasad se pokazuje kako bi najefikasnije bilo prvo inicijalnu poziciju izračunat pomoću točaka značajki i onda ih optimizirat direktnim metodama [12].

Očigledno je vizualna odometrija ključna u rješavanju SLAM-a, no vizualna odometrija je sklona akumulativnom odmaku (eng. *accumulative drift*). Vizualna odometrija računa pomak između svake dvije slike odnosno kadra (*frame*-a) te svaki izračun u sebi sadrži neki gubitak. Kako se izračunava inkrementalno, svaki gubitak se prenosi na sljedeći izračun te se kroz duži vremenski period akumulira. Kako bi riješili problem akumulativnog pomaka, potrebna su nam ostala dva segmenta sustava, *backend* i *loop closure*.

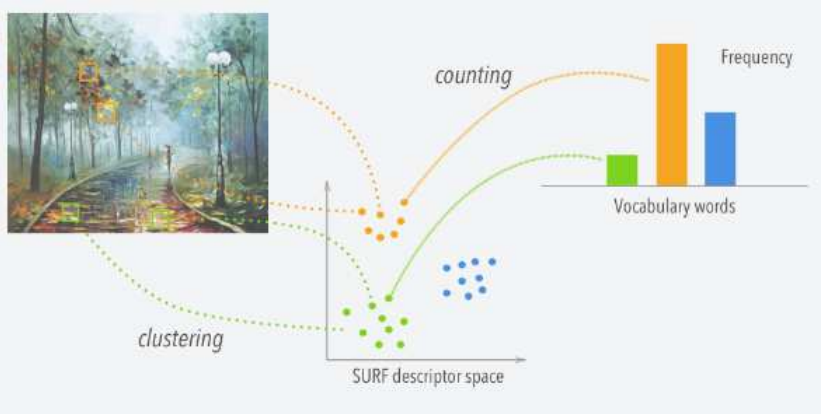
Backend je u suštini zadužen za uklanjanje šuma i gubitka iz vizualne odometrije. S obzirom da nam je u interesu dobiti rezultate s visokom preciznošću, potrebno je promatrati cijelu putanju,a ne samo između dva kadra, te je potrebno odrediti pouzdanost tih rezultata s obzirom na prisustvo šuma. Naivni pristup bio bi ukloniti šum, no to se pokazalo nemogućim čak i za najskuplje senzore. Zato se koristi teorija vjerojatnosti kako bi se procijenila pouzdanost rezultata, točnije pozicije kamere i rekonstrukcija prostora u kojem se nalazi. Koristeći Bayesovo pravilo, cijeli sustav je prikazan kao skup stanja te optimalna procjena stanja i prostor dobivaju se kombinirajući informacije o kretnji robota i informacije o prostoru. Ovakvom formulacijom problema vidimo da se ovo svodi na klasičnu Maximum-a-Posteriori (MAP) estimaciju. U početcima vizualnog SLAM-a, filter metoda bila je primaran način rješavanje optimizacije. Prva filter metoda primjenjena u SLAM-u bila je Kalman filtriranje (KF) [13], koja se bazirala na rješavanju linearnih sustava minimizirajući RMSE (eng. *root mean squared error*). Kako se ubrzo pokazalo da su SLAM sustavi pretežno nelinearni, prezentirana je proširena Kalman fil-

ter metoda koja je izvršavala linearnu optimizaciju Taylorovog reda linearnog sustava [14]. No, u posljednjem desetljeću pokazalo se kako su metode bazirane na teoriji grafova preciznije te su one postale standard u razvoju SLAM sustava. Ove metode rade estimaciju za sve promatrane podatke te ih nanovo lineariziraju kad dođe do promjene konačne estimacije. Kako u svakom koraku optimizacije optimiziramo sve parametre, ove metode su vrlo računarski zahtjevne. Moderna istraživanja u području optimizacije SLAM sustava se ponajviše bave poboljšavanjem metoda koje se baziraju na teoriji grafova ([15], [16]).

Loop closure je glavni način kako riješiti problem akumulativnog pomaka. Pretpostavimo situaciju gdje promatramo robota i pokušavamo procijeniti njegovu poziciju. U jednom trenutku robot će se vratit na početnu poziciju, no naša estimirana pozicija se neće podudarati sa stvarnom zbog akumulativnog pomaka. Očigledno najbolje rješenje u ovakvoj situaciji bila bi mogućnost da robot može prepoznati lokacije na kojima je već bio, a to je svrha *loop closure*-a. Temeljna provjera *loop closure* algoritma je provjera korespondencije trenutne slike odnosno kadra sa prijašnjim slikama. Postoji više načina usporedba korespondencije (usporedba na bazi piksela [17], na bazi korespondencija kompleksijih značajki [18]), te nakon korespondencije se provjerava je li ona iznad određenog praga, ako je onda se smatra da smo pronašli već posjećenu lokaciju. No, za veću pouzdanost rezultata potrebno je usporediti trenutnu sliku sa što većim brojem prijašnjih. S obzirom da je kompleksnost tog zadatka minimalno linearna i ovisi o broju prijašnjih slika, lako vidimo da ovdje imamo glavno ograničenje jer u cilju nam je da se izračun izvršava u stvarnom vremenu [19]. Modernije metode pokušavaju riješiti ovaj problem boljom selekcijom prijašnjih slika za usporedbu [20] ili korištenjem Bag-of-Words principa [21]. U tom pristupu, slike su prikazane kao vektor "riječi" gdje riječi opisuju sliku i predstavljaju značajke. Ti vektori riječi odnosno histogrami se onda koriste za pronalazak korespondencija između slika. Slika 2.3. vizualno prikazuje takav proces generiranja riječi odnosno histograma.

2.3. Trendovi

Iako smo naizgled riješili problem SLAM-a, potreba za boljim i efikasnijim metodama i dalje raste. U jednoličnim prostorima sa malo teksture, pri naglim promjena osvijetljenja



Slika 2.3. Generiranje vektora odnosno histograma riječi iz značajki.
Slika preuzeta sa [22].

te pri naglim pokretima kamere, većina sustava ne daje prihvatljive rezultate. Tu vidimo potrebu za sve robustnijim sustavima koji bi bili otporni na te probleme.

U prostorima bez teksture i sa malo teksture, klasični deskriptori značajki ne mogu konzistentno pronalaziti točke značajki između kadrova te dolazi do gubitka praćenja putanje ili velikog akumulativnog pomaka. U tu svrhu predložene su linije značajki. Linije značajki imaju prednost što su u njima enkodirane informacije o globalnoj orijentaciji i smjeru [23]. Također su se pokazale efikasnima pri snimanju unutrašnjih prostora sa više linearnih značajki te u situacijama kad dolazi do zamućenja slike [24]. Klasični deskriptori tad često stvaraju krive korespondencije te dobivamo nekonzistente rezultate, no linije značajki zbog svojih svojstva lako rade tu distinkciju.

Drugi pristup kojem se u posljednje vrijeme posvećuje najviše pažnje je korištenje dubokog učenja. U duhu razvoja tehnologije dubokog učenja i njegove superiornosti nad tradicionalnim metodama strojnog učenja, sve više znanstvenika počinje implementirati duboke modele u razne dijelove SLAM sustava. Pri procjeni pozicije kamere između kadrova koriste se proširene i detaljnije točke značajki dobivene od unaprijed treniranih dubokih modela. Sustav LIFT-SLAM [25] koristi takve točke značajki te ih koristi uz standardne sustave temeljene na klasičnim deskriptorima. Proširuje se klasični SLAM sustav sa ORB deskriptorom značajki [26] tako da se koriste točke značajki dobivene od konvolucijske neuronske mreže bazirane na LIFT (eng. *Learned Invariant Feature Transform*) dubokoj neuronskoj mreži. Sustav RWT-SLAM [27] predložen je za scene sa nedostatkom teksture. Također baziran na ORB-SLAM sustavu, konvolucijska neuron-

ska mreža je korištena uz LoFTR [28] algoritam gdje je konvolucijska neuronska mreža zadužena za grublje, a LoFTR za detaljnije značajke. Oba sustava pokazala su se superiornijima i efikasnijima naspram prijašnjih sustava koji su koristili tradicionalne pristupe. No duboko učenje ne koristi se samo za ekstrakciju značajki, već i u druge svrhe. Sustav znanstvenika Nex i Steenbeck [29] koristi ResNet-50 [30]. Njihovo rješenje koristi monokularnu kameru postavljenu na dron te koristi duboku neuronsku mrežu za precizniju procjenu dubine i samim time, precizniju mapu prostora. Sustav Dynamic-SLAM [31] koristi prednosti dubokih modela za bolju procjenu pozicije i okruženja. Konvolucijska neuronska mreža koristi se za segmentaciju objekata na slici kako bi se napravila distinkcija koji objekti su dinaminični i ne zanimaju sustav te koji su dio statičke scene relevantne za izračun i rekonstrukciju. Sustav DeepVO [32] implementira RCNN (eng. *Recurrent Convolutional Neural Network*) unutar monokularnog SLAM sustava. Sustav koristi duboko učenje za automatsku ekstrakciju točaka značajki, koje unaprijed trenirani model nauči kao najboljima, izravno iz slika sa senzora. Nakon toga se koristi FlowNet [33] konvolucijska neuronska mreža za izračun optičkog toka i procjenu temporalnih promjena u sceni.

2.4. Primjena

Vidimo da veliki broj istraživača se bavi SLAM sustavima, no može se postavit pitanje zašto, odnosno otkud tolika potreba za tako robustnim sustavima? Dat ćemo uvid u samo neke od primjena SLAM sustava kako bi shvatili važnost SLAM-a te razlog tolikog interesa istraživača.

Kao glavna primjena vizualnog SLAM-a se može navesti autonomna vožnja koja je trenutno i dalje u začetcima. Tvrtke poput Buggati Rimac i Muskove Tesle su najveće tvrtke koje se aktivno bave time, Buggati Rimac ima planove pustiti svoje "robotaksije" do kraja 2025. godine na području Zagreba, dok Tesla planira izvršiti javna testiranja u Kini 2024. godine. Stefan Milz i suradnici u svom radu [34] rade podjelu autonomne vožnje na tri ključna segmenta: parkiranje, vožnja po urbanom području i vožnja po autocesti. Najveć zahtjev pri parkiranju je stvaranje precizne mape okruženja u neposrednoj blizini vozila pri manjim brzinama. Najčešće situacije su parkiranje na javnim parkiralištima i parkiranje u području vlastite kuće odnosno garaže. Za parkiranje na

javnim mjestima potrebna je precizna lokalizacija no samo na malom području, neposredno oko vozila. No parkiranje kod vlastite kuće zahtjeva dodatne funckionalnosti kako bi autonomna vožnja bila moguća. Za početak, vozilo nauči određenu putanju te kreira inicijalnu mapu. Na povratku, promatranu mapu i putanju se optimizira te vozilo je u mogućnosti voziti naučenu putanju te obnavljati mapu svaki put kad prođe putanjom. Vožnja autocestom ima potpuno drukčije zahtjeve od parkiranja. S obzirom na velike brzine, potrebna je velika brzina izvođenja kako bi se što češće mapa mogla nadograđivati. No, olakšica je što za razliku od parkiranja i vožnje urbanim područjem sam prostor kretnje je puno jednostavniji. Druga vozila nisu nasumično poredana u svim orijentacijama oko nas, nego su paralelna s nama i imaju istu orijentaciju. Naposljetku, vožnja urbanim područjem je sredina između druge dvije situacije te je ona ujedno i najzahtjevnija iz perspektive vizualnog SLAM-a. Brzine su i dalje relativno velike te samo okruženje je puno kompliciranije nego u prijašnje objašnjenim situacijama. Potreban je sustav koji u stvarnom vremenu (barem 30 FPS-a) može stvarati preciznu mapu te odrediti koji objekti su dinamični, a koji statični. Da stavimo u perspektivu koliko je točno zahtjevnije, prije spomenuti ORB-SLAM testiran na KITTI skupu podataka [35], na scenarijima sa autocestom ostvarili su RMSE (eng. *Root Mean Squared Error*) od 1.79 metara, a na scenarijima na urbanom području ostvarili su RMSE od 46.36 metara.

Slika 2.4. DJI Phantom bespilotna letjelica. Slika preuzeta sa [36].

Druga veća primjena vizualnog SLAM-a je kod bespilotnih letećih vozila (eng. UAV, *Unmanned aerial vehicles*). U posljednje vrijeme vidimo porast njihove primjene u raznim područjima poput inspekcije dalekovoda, geoloških istraživanja i prevencije požara u šumi, ponajviše zbog njihove fleksibilnosti i mnoštva mogućnoti koje nude. Iako zasad i dalje se koriste samo u područjima sa dobrom pokrivenosti GPS satelitima, vidi se sve veća potreba za njihovom autonomnosti. Većinu UAV sustava dodatno upravlja ili nadgleda operater što može dovesti do nekih propusta. DJI Phantom 2.4. je letjelica sa implementiranim sustavom upravljanja koja omogućava letjelici samostalno preživljavanje i prolaženje kroz složenija okruženja [37]. Brzi napredak vizualnog SLAM-a je omogućio ovakve inovacije u posljednjem desetljeću, no i dalje nisu riješeni prije spomenuti problemi. Kod ovakvih letjelica posebno je izraženo zamućenje slike uzrokovano naglim pokretima. Također je potrebno naglasiti kako uz napredak tehnologije ometanja satelita, vojne bespilotne letjelice sve više pokušavaju se odmaknuti od satelitskog navođenja te prebaciti se na vizualni SLAM sustav [38].

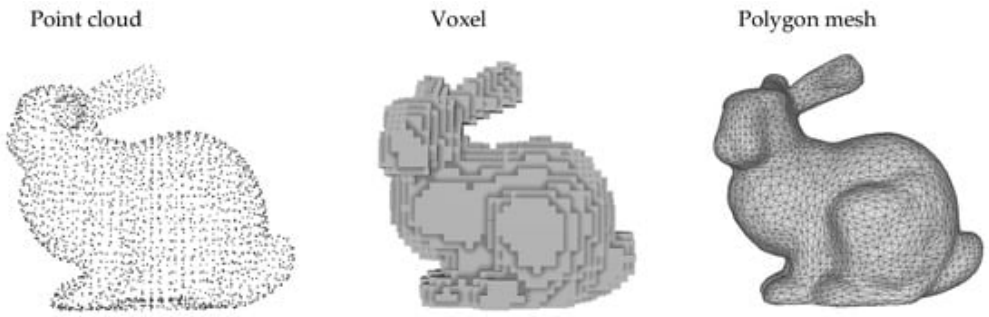
3. Reprezentacije scene

Kako bi robota mogli smatrati autonomnim, robot mora imati sposobnost navigacije u nepoznatim prostorima. U SLAM sustavima, za rekonstrukciju je zaslužno mapiranje, a ono uvelike ovisi o izboru reprezentacije scene. Složenije reprezentacije scene rezultiraju preciznijim, no memorijski i vremenski zahtjevnijim sustavima što ih čini neprikladnima za širu upotrebu u stvarnom vremenu. Iako je moguće ostvariti SLAM sustav bez reprezentacije scene odnosno uzimanjem informacije izravno iz RGB slika, Sifferman [39] tvrdi kako implementiranje reprezentacije scene je korisno iz nekoliko razloga: služe kao prostorna memorija, efikasno spremište za starije informacije, dopuštaju dugoročno planiranje te mogu služiti kao regularizacija i enkodirati prostorna a priori znanja za učenje sustava. Također, radi podjelu reprezentacije scena po zadaćama u kojima se koriste: izbjegavanje sudara, manipulacija robotom i upravljanje na daljinu.

Izbjegavanje sudara jedna je od ključnih prepreka u području robotike. Za manje, sporije robote ovaj zadatak nije toliko zahtjevan jer nije potrebna velika brzina izvođenja, no za brže robote dosadašnji SLAM sustavi nisu bili u mogućnosti izvršavati se u stvarnom vremenu. Postoje pristupi ([40]) koji uzimaju podatke izravno sa senzora i ne stvaraju internu reprezentaciju prostora, no taj nedostatak memorije je nepremostiv nedostatak jer postavlja se pitanje kako isplanirati putanju robota odnosno provesti algoritam zatvaranja petlje. Iako se moderni sustavi za izbjegavanje sudara mogu podijeliti na dvije kategorije, planiranje pokreta i inverznu kinematiku, po uzoru na [39] nećemo raditi distinkciju jer rješenja su ista u obje kategorije. Neke od reprezentacija scena koje su se dosad koristile su SDF (eng. *Signed Distance Fields*, mapiranje između 3D točke u prostoru i skalar udaljenosti d sa najbližom preprekom), kolekcije primitiva (skupina primitiva poput sfera i kvadrata zadanih parametarski) i kolekcije konveksnih ljusaka (slično skupini primitiva, umjesto oblika spremaju se konvekse ljuske prigodne

za moderne optimizatore).

Ako i uspijemo napraviti sustav za autonomno kretanje robota, nema smisla da taj robot samo prolazi zadanom putanjom. Točnije, želimo da naš robot može izvršavati neke zadatke, na primjer ugasiti svijetlo ili donijeti kutiju. Kako bi se to omogućilo potrebno je imati sustav za manipulaciju robota. Sustavi za manipulaciju se razlikuju od sustava za izbjegavanje sudara ponajviše u razini detalja potrebnoj u reprezentaciji scene. Uobičajeno, objekti koje bi robot, recimo prenosio, biti će manji od cijelih okruženja gdje se robot nalazi. Također, ako želimo da robot sigurno može manipulirat tim objektima potrebna je i određena razina detalja koja nije potrebna pri snalaženju u prostoru. Neke od dosad korištenih reprezentacija su *meshevi* (koriste se 3D mreže trokuta za reprezentaciju eksterijera objekta), oblaci točaka (eng. *point clouds*, velika nakupina 3D točaka koja promatrana u cjelini predstavlja objekt odnosno scenu) i mrežu voksela (eng. *voxel grid*, može se zamisliti kao skupina 3D piksela). Na slici 3.1. možemo vizualno usporediti sva tri pristupa.



Slika 3.1. Razlike između *point cloud*, *voxel grid* i *mesh* reprezentacija.
Slika preuzeta iz [41].

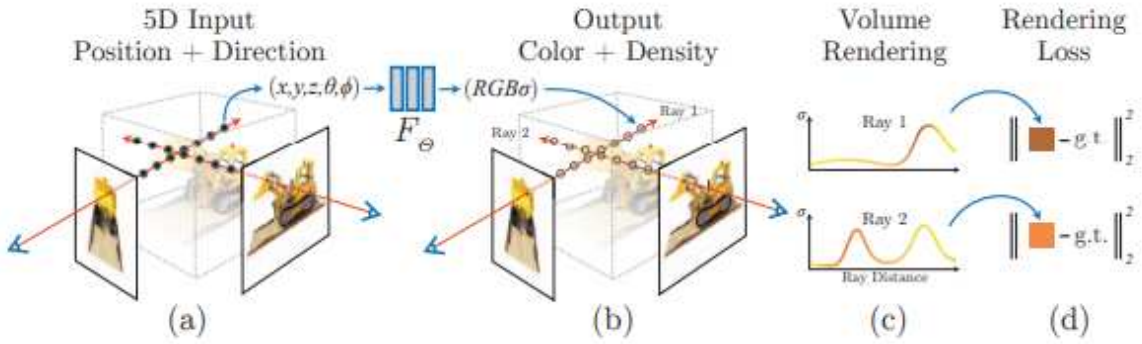
Posljednja zadaća koju ćemo razmotrit je udaljeno upravljanje odnosno teleoperacija robotima. Uglavnom, ti sustavi se temelje na neposrednom sučelju kojim korisnik može upravljati u stvarnom vremenu. U posljednje vrijeme posebno se istražuje takvo upravljanje unutar VR (eng. *Virtual Reality*) okruženja [42]. No, reprezentacije scene korištene u tim okruženjima mogu se primijeniti i na zadatke računalnog vida. Osim, očiglednog, izravnog korištenja statičnih kamera, koriste se prijašnje objašnjeni *point cloud*, *mesh* i mreže okupiranosti koje su predložili Omarali i suradnici u svom radu [43] te se one koriste uz dubinske kamere kao hibridni pristup VR teleoperiranju robotima.

Ukratko smo objasnili samo neke od dosad učestalo korištenih reprezentacija. U ok-

viru ovog rada detaljno ćemo objasniti dvije moderne reprezentacije koje su u posljednje vrijeme prikupile mnogo pažnje u području računalnog vida, neuronska radijalna polja (eng. *Neural Radiance Fields*, NeRF) i reprezentacije pomoću Gaussovih krivulja (eng. *gaussian splatting*).

3.1. Neuronska radijalna polja

Neuronska radijalna polja (skraćeno NeRF), inicijalno predložena u radu [44], su tehnika reprezentacija scene gdje je scena prikazana kao kontinuirana funkcija u 5 dimenzija koja vraća odsjaj u svim smjerovima na svim lokacijama u sceni. Za prikaz te funkcije predloženo je da se koristi potpuno povezana neuronska mreža tako da radi regresiju od 5D koordinate do boje i gustoće. Za prikaz NeRF-a uzorkuju se trodimenzionalne točke duž zraka iz kamere, te točke i kuteve gledanja kamere dajemo onda neuronskoj mreži kao ulaz za estimaciju boje i gustoće te naposljetku koristimo standardne tehnike prikaza kako bi saželi dobivenu boju i gustoću u dvodimenzionalne piksele na slici. Zbog diferenecijabilnosti predloženog procesa, koristi se gradijentni spust za optimizaciju modela minimizirajući pogrešku između stvarnih i procijenjenih vrijednosti pojedinih piksela [44]. Slika 3.2. prikazuje pregled cjelokupnog predloženog sustava. U nastavku ćemo objasniti svaki dio ovog sustava u detalje.



Slika 3.2. Pregled NeRF reprezentacije scene i postupka prikaza. Uzorkujemo 5D točke, koje sadrže koordinate i kuteve gledanja, duž zraka iz kamere (a), prosljeđujemo uzorkovane točke u potpuno povezanu neuronsku mrežu (b) i naposljetku sažimamo dobivene podatke u 2D reprezentaciju slike koristeći standardne tehnike prikaza (c). Sustav je diferencijabilan pa za optimizaciju se koristi minimizacija gubitka između stvarnih i procjenjenih vrijednosti piksela (d). Slika preuzeta iz [44].

Kao što smo prijašnje spomenuli, reprezentiramo scenu kao funkciju u pet dimenzija:

$$F_\Theta(\mathbf{x}, \mathbf{d}) \longrightarrow (\mathbf{c}, \sigma) \tag{3.1}$$

gdje $\mathbf{x}$ predstavlja trodimenzionalnu koordinatu točke $\mathbf{x} = (x, y, z)$, $\mathbf{d}$ predstavlja kut gledanja (dva najčešća načina su kao kut između koordinatnih osi $\mathbf{d} = (\theta, \phi)$ i kao trodimenzionalni vektor u Kartezijevom sustavu $\mathbf{d} = (d_x, d_y, d_z)$. Optimiziramo parametre Θ potpuno povezane neuronske mreže tako da 5D ulaz mapira na odgovarajuću gustoću volumena i boju vidljivu iz danog kuta gledanja. Ova funckija je ograničena da bude konzistenta sa više kamera odnosno više istovremenih kuteva gledanja tako da gustoća volumena σ bude neovisna o kutu gledanja $\mathbf{d}$, dok boja ovisi i o kutu gledanja $\mathbf{d}$ i o 3D koordinati $\mathbf{x}$. U izvornom radu [44], to je ostvareno tako da je neuronska mreža odnosno perceptron dizajniran da bude u dvije faze. U prvoj fazi kao ulaz se uzima $\mathbf{x}$ i na izlazu je σ i vektor značajki velike dimenzije. U drugoj fazi taj vektor značajki kombinira se sa kutom gledanja $\mathbf{d}$ i prosljeđuje se drugom perceptronu koji kao izlaz onda daje boju $\mathbf{c}$. Iako u originalnom radu ova dva perceptrona su smatrana kao dvije grane iste mreže, noviji autori smatraju kako trebaju biti dvije odvojene mreže [45].

Dobivenu gustoću σ i boju $\mathbf{c}$ prosljeđujemo u sustav za prikaz volumena [46] kako bi dobili konačnu boju C($\mathbf{r}$) u proizvoljnoj zraci iz kamere $\mathbf{r}$(t) $= \mathbf{o} + t\mathbf{d}$, gdje $\mathbf{o}$ predstavlja trenutnu poziciju kamere i $\mathbf{d}$ smjer gledanja, koristeći izraz

$$C(\mathbf{r}) = \int_{t_1}^{t_2} T(t) \cdot \sigma(\mathbf{r}(t)) \cdot \mathbf{c}(\mathbf{r}(t, \mathbf{d})) \cdot dt \tag{3.2}$$

gdje $\sigma(\mathbf{r}(t)$ i $\mathbf{c}(\mathbf{r}(t, \mathbf{d}))$ predstavljaju gustoću i boju u točki $\mathbf{r}(t)$ duž smjera gledanja kamere sa kutom $\mathbf{d}$ i dt predstavlja inkrementalni pomak duž zrake u smjeru gledanja kamere koji je prijeđen u svakom koraku integracije. $T(t)$ predstavlja vjerojatnost da je zraka došla od točke $\mathbf{r}(t_1)$ do točke $\mathbf{r}(t)$ bez prekida. Ta vjerojatnost određena je izrazom

$$T(t) = exp(-\int_{t1}^{t} \sigma(\mathbf{r}(u)) \cdot du) \tag{3.3}$$

Taj integral moguće je izračunati numerički. Originalni rad [44] koristio je nedeterministički pristup gdje je zraka bila isprekidana u N ravnomjerno raspoređenih segmenata i uzorkovana je točka u svakom segmentu. Zbog toga jednadžba 3.2 se može aproksimirati

sa

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} \alpha_i T_i \mathbf{c}_i, \quad \text{gdje} \quad T_i = exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j) \tag{3.4}$$

δ_i je udaljenost između uzorkovanih točaka i i $i+1$, σ_i i $\mathbf{c}_i$ su gustoća volumena i boja dobivene iz zrake iz kamere i α_i predstavlja transparentnost dobivenu slaganjem alfe u uzorku i

$$\alpha_i = 1 - exp(-\sigma_i \delta_i) \tag{3.5}$$

Dubina se može procijeniti pomoću akumulirane transparentnosti [45] izrazom

$$d(\mathbf{r}) = \int_{t_1}^{t_2} T(t) \cdot \sigma(\mathbf{r(t)} \cdot t \cdot dt \tag{3.6}$$

što se može aproksimirati na isti način kao jednadžbe 3.2, 3.3 i 3.4

$$\hat{D}(\mathbf{r}) = \sum_{i=1}^{N} \alpha_i t_i T_i \tag{3.7}$$

Pokazalo se kako uzorkovanje u svakom segmentu nije efikasno te su autori predložili hijerarhijsku strukturu sustava. Prvo uzorkujemo skup od N_c primjeraka te evaluiramo "grubu" mrežu na tim pozicijama po jednadžbi 3.4 Pomoću tih grubih procjena uzorkujemo novu skupinu pozicija gdje su uzorci više priklonjeni relevantnim dijelovima volumena. To je postignuto preformulacijom izraza za transparetnost alfe

$$\hat{C}_c(\mathbf{r}) = \sum_{i=1}^{N_c} w_i c_i, \quad \text{gdje} \quad w_i = T_i(1 - exp(-\sigma_i \delta_i)) \tag{3.8}$$

Normalizacijom tih težina w_i dobivamo novu distribuciju vjerojatnosti duž zrake. Uzorkujemo skupinu pozicija veličine N_f te ih evaluiramo na skupu koji sadržava sve primjere iz grube i fine procjene. Naposljetku izračunavamo konačnu boju pomoću izraza 3.4 Za optimizaciju parametara Θ neuronske mreže koristi se ukupni kvadratni gubitak grubog i detaljnog prikaza

$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} [||\hat{C}_c(\mathbf{r}) - C(\mathbf{r})||_2^2 + ||\hat{C}_f(\mathbf{r}) - C(\mathbf{r})||_2^2] \tag{3.9}$$

gdje $\mathcal{R}$ predstavlja skupinu zraka u svakoj seriji, C($\mathbf{r}$) stvarnu poziciju, $\hat{C}_c(\mathbf{r})$ grubu pro-

cjenu volumena i $\hat{C}_f(\mathbf{r})$ detaljnu procjenu volumena.

U originalnom radu [44], autori su istaknuli da koristeći ulaz u obliku (x, y, z, θ, ϕ) nisu dobivali prihvatljive rezultate u scenama sa visokofrekventnim varijacijama boja, što je poduprijeto rezultatima u [47]. Rahaman i suradnici su u [47] pokazali da su duboke neuronske mreže sklonije učenju funkcija nižih frekvencija. Mildenhall i suradnici su [44] pokazali kako rastav funkcije F_Θ na kompoziciju dvije funkcije $F_\Theta = F'_\theta \circ \gamma$, jednu unaprijed naučenu i jednu ne, uvelike poboljšava rezultate. Uloga γ je mapiranje $\mathcal{R}$ u prostor više dimenzije. Formalno enkodiranje, nazvano pozicijsko enkodiranje, glasi

$$\gamma(p) = (sin(2^0\pi p), cos(2^0\pi p), ..., sin(2^{L-1}\pi p), cos(2^{L-1}\pi p)) \tag{3.10}$$

i primjenjuje se na svaku vrijednost točke (x, y, z) i na svaku dimenziju vektora pogleda (d_x, d_y, d_z). L predstavlja parametar dimenzionalnosti kojeg zadaje korisnik, u radu [44] $L = 10$ za $\gamma(\mathbf{x})$ i $L = 4$ za $\gamma(\mathbf{d})$.

Neuronska radijalna polja su od iznimne važnosti za robotiku i računalni vid zbog svojih prednosti nad prijašnjim metodama, poput prostorne kompaktnosti memorije, kontinuirane reprezentacije scene i mogućnosti korištenja matematičkih modela. Wang i suradnici [48] u svom radu vrše podjelu NeRF sustava u dvije kategorije, zadužene za razumijevanje okoline i zadužene za interakciju s okolinom.

Sustave zadužene za razumijevanje okoline možemo podijeliti na sustave zadužene za rekonstrukciju scene, sustave zadužene za segmentaciju i sustave za uređivanje scena. Nadalje, sustave za rekonstrukciju možemo podijeliti na sustave za statičnu rekonstrukciju i sustave za dinamičnu rekonstrukciju. Kad gledamo statičnu rekonstrukciju razlikujemo dva slučaja, rekonstrukciju unutarnjih i rekonstrukciju vanjskih prostora. S obzirom da razmatramo samo statične rekonstrukcije, pri rekonstrukciji unutarnjih scena možemo imati pretpostavke koje uvelike pojednostavljuju problem rekonstrukcije, poput konstantnog osvijetljenja. Sustav iMAP [49] je sustav također baziran na MLP strukturi i gustoći volumena koji uspjeva dobiti prihvatljive rezultate pri rekonstrukciji unutarnjih prostora iz samo 2D slika. No, zbog svojstava MLP strukture, sustav je ograničen samo na prostore manje skale. Motivirani tim ograničenjima, Kruzhov i suradnici predložili su MeSLAM [50], memorijski efikasan SLAM sustav baziran na NeRF reprezentacijama.

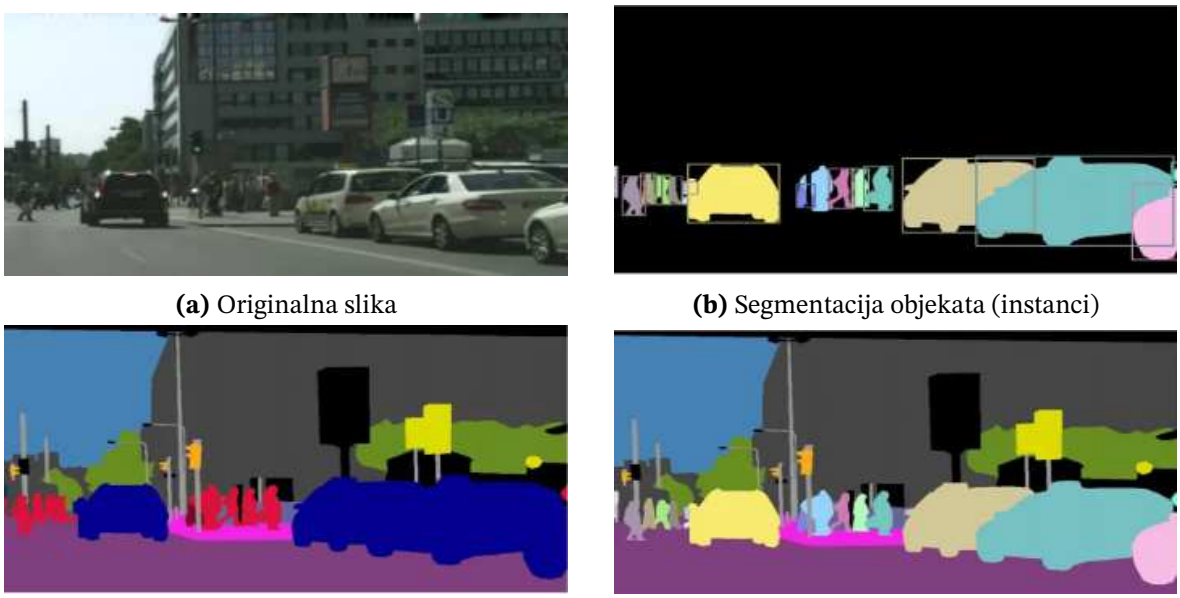
MeSLAM omogućava tu memorijsku efikasnost tako da se scena podijeli u regije koje se tek spajaju u trenutku rekonstrukcije. Dakle, za vrijeme prolaska robota, rekonstrukcija se obavlja samo u vidljim dijelovima scene, dok nevidljivi dijelovi scene su spremljeni u segmente te time zauzimaju manje memorije. U drugu ruku, sustavi poput NeRF-W [51] i Block-NeRF [52] namijenjeni su za rekonstrukciju vanjskih prostora te rješavaju probleme poput različitog osvijetljnja i prolaznih objekata poput ljudi i automobila. NeRF-W to ostvaruje tako da slike iz scene rastavlja na komponente koje su vezane uz slike i koje se ponavljaju, dok Block-NeRF to postiže markerima u sceni, optimizacijom naučenih pozicija i varijabilnim *exposure*-om.

Roboti koji su namjenjeni dugotrajnoj uporabi susretat će se sa naglim promjenama u kompleksnom okruženju. Za rješavanje takvih dinamičnih okruženja, u praksi se koriste tri reprezentacije dinamičnog pokreta. U prvoj reprezentaciji, dodaju se dodatna ograničenja u sustave bazirana najčešće na vremenskim i prostornim promjenama. Na primjer, STaR sustav [53] koristi dinamični NeRF za reprezentaciju kretajućeg objekta u sceni te optimizira pozicije tog objekta ovisno o vremenskim oznakama, dok sustav DyNeRF [54] koristi vremenski ovisno enkodiranje za dinamičko polje kako bi postigli bolje rezultate sa varijacijama u površini i prolaznim objektima u sceni. Druga reprezentacija je reprezentacija deformacijom. Pokret je prikazan deformacijom statičnog prostora koji je već opažen i za kojeg je stvorena mapa. Ta mapa mora biti konzistentna sa svim ostalim senzorima, čime se osigurava njena točnost. Razlike od tog statičnog prostora odnosno polja predstavljaju pokrete u lokalnim područjima scene. Sustavi poput D-NeRF [55] i RoDynRF [56] implementiraju takve reprezentacije deformacijom, dok sustav RoDynRF još nadograđuje koncept sa paralelnim praćenjem putanje za ostvarivanje zadovoljavajućih rezultata pri scenama sa dinamičnim pokretima. Posljednja reprezentacija dinamičnog pokreta je sa tokom. Za razliku od deformacija, tok služi za prikazivanje pokreta na cijeloj sceni, a ne samo u lokalnim regijama. Sustav NSFF [57] kao izlaz MLP mreže uz boju i gustoću, vraća i 3D pokret kao tok u sceni te se za optimizaciju mreže koristi gubitak tih tokova uspoređujući i prethodni i sljedeći kadar što osigurava da rekonstrukcija bude vremenski konzistentna. Drugi sustav koji je bitno spomenuti je sustav od Gaoa i suradnika [58]. Ovaj sustav je od iznimnog značaja zato što su predložili da se koriste dvije odvojene reprezentacije za istu monokularnu scenu. Statični NeRF neovisan o vremenu korišten je za reprezentaciju nepromjenjivog dijela prostora, dok je dodatan

dinamičan NeRF korišten za varijabilne dijelove u prostoru.

Druga skupina sustava za razumijevanje okoline su sustavi za segmentaciju. Segmentacija scene odnosi se na proces raspodjele scene u različite komponente. Te komponente se određuju ovisno o zadaći sustava, te ovisno o zadaćama, dosadašnje sustave možemo podijeliti na sustave za segmentaciju objekata, sustave za semantičku segmentaciju i sustave za panoptičku segmentaciju. Cilj sustava za segmentaciju objekata, odnosno instanci, je precizno razlikovanje statične pozadine od dinamičnih prolaznih objekata. Primjer takvog sustava je sustav ONeRF [59] koji izvršava automatsku nenadziranu segmentaciju gdje je svaki objekt prikazan odvojenim NeRF-ovima. Nenadzirana segmentacija je ostvarena iterativnim algoritmom za maksimizaciju očekivanja kako bi se učinkovito 2D značajke mapirale na 3D točke dobivene iz više kamera. Za razliku od sustava za segmentaciju objekata, semantička segmentacija pridjeljuje svakoj 3D točki njenu semantičku oznaku. Sustav Semantic-NeRF [60] proširuje standardnu NeRF reprezentaciju tako da enkodira podatke o semantici uz podatke o izgledu i geometriji. Drugi bitan sustav za spomenut je iLabel [61] koji implementira SLAM sustav u stvarnom vremenu. Sustav nadograđuje prijašnje spomenuti iMAP sustav [49] sa NeRF reprezentacijama te se semantički razredi zadaju u trenutku izvođenja od strane korisnika. To omogućava sustavu da preskoči dio treniranja prije korištenja i odmah krene *online* učenje. Naposljetku imamo panoptičku segmentaciju koja je kombinacija prethodne dvije. U procesu panoptičke, svi pikseli se segmentiraju te im se pridjeljuju semantičke značajke i značajke instanci. Sustav Panoptic NeRF [62] osmišljen je za scene vožnje na vanjskim prostorima koristeći 2D semantičke oznake i 3D okvire za objekte u sceni. Stvaraju se dva odvojena semantička polja, jedno fiksno koje poboljšava geometriju prilikom rekonstrukcije i jedno unaprijed istrenirano polje za dodjelu semantičkih značajki. Za lakše razumijevanje, na slici 3.3. možemo vidjeti vizualne razlike u rezultatima dobivenim od različitih segmentacija.

Naposljetku imamo sustave zadužene uređivanju scena. Iako naizgled irelevantne iz perspektive SLAM sustava, uređene slike mogu služiti kao podatci za treniranje sustava. Naime, snimanje scena za treniranje može biti ili nemoguće ili vremenski prezahtjevno da se obavlja ručno. Uređivanjem scena se taj proces uvelike olakšava odnosno ubrzava što je od velikog značaja tijekom testiranja razvijenih sustava. NeRF tehnologija



(a) Originalna slika

(b) Segmentacija objekata (instanci)

(c) Semantička segmentacija

(d) Panoptička segmentacija

Slika 3.3. Vizualne razlike u različitim segmentacijama. Slike preuzete sa [63].

tu igra ključnu ulogu poboljšavajući realnost i konzistenciju. Kad pričamo o uređivanju scene, moguće je uređivati pojedinačne objekte, dodavati odnosno brisati objekte i uređivati izgled cijele scene. CodeNeRF [64] uči odvojiti geometrijsku strukturu i teksture koristeći odvojene ugrađene informacije. Iako se takav pristup pokazao efikasnim, Xu i suradnici u svom radu [65] tvrde kako eksplicitne reprezentacije poboljšavaju skalabilnost i performanse sustava. Njihov sustav enkapsulira NeRF-ove u "kaveze" te se deformacije postiže promjenama u vrhovima tih kaveza. Iako prethodni sustavi su imali samo mogućnost manipulacije objektima, Ost i suradnici su u svom radu [66] predložili da se scena dekomponira kao graf scene. Premda je njihov rad originalno osmišljen kao rješenje za dinamične scene, njihova eksplicitna reprezentacije scene omogućava dodavanje odnosno brisanje objekata jednostavnim umetanjem ili micanjem čvorova iz grafa scene. Kao zadnju ulogu uređivanja slika imamo uređivanje cijele scene. To se najčešće odnosi na proces stilizacije scena koji nije od velike važnosti za računalni vid, no jedan sustav iskače. Sustav ClimateNeRF [67] kombinira klasični NeRF sustav sa fizičkim simulatorima za vremenske prilike čime se postiže precizna i konzistentna rekonstrukcija prostora za vrijeme utjecaja prirodnih fenomena poput poplava i požara.

Kao što smo prije spomenuli, druga kategorija NeRF sustava su sustavi zaduženi za interakciju robota s okolinom. To se primarno odnosi na procese navigacije i manipulacije prostorom u kojem se nalaze. Navigacija odgovara na dva pitanja: gdje se robot

nalazi i kako robot dolazi do cilja. Na prvo pitanje odgovor nam daje lokalizacija, a na drugo planiranje puta. Lokalizacija određuje lokaciju robota u 6 stupnjeva slobode (end. *Degree-of-Freedom*, DoF), 3 stupnja za poziciju i 3 za orijentaciju, te se na temelju a priori znanja o mapi mogu podijeliti na lokalizaciju s unaprijed poznatom mapom i unaprijed nepoznatom mapom. Sustavi s unaprijed poznatim mapama, poput INeRF-a [68] i Direct-PoseNet [69], uobičajeno koriste unaprijed trenirane NeRF modele kao reprezentacije scene. INeRF ima arhitekturu inverznog NeRF sustava i koristi fotometrični gubitak na razini piksela za optimizaciju pozicije i orijentacije kamere, dok sustav Direct-PoseNet koristi izlaze iz NeRF modela kao ulaz u mrežu za regresiju apsolutne pozicije kamere (eng. *Absolute Pose Regression*). Za razliku od takvih sustava, sustavi poput NeRF– [70] i GARF [71] nemaju unaprijed poznatu mapu prostora. NeRF– paralelno optimizira poziciju kamere i parametre NeRF modela zaduženog za reprezentaciju scene. Sustav GARF jedan je od prvih sustava koji kreće u smjeru istraživanja Gaussovih funckija u rekonstrukciji prostora koristeći Gaussove aktivacijske funkcije pri čemu su ostvarili veću preciznost estimacije pozicije i bolju generalizaciju sustava. Planiranje puta odnosno trajektorije robota druga je zadaća navigacije. NeRF reprezentacije su tu od iznimne važnosti jer su prigodne zadatku, točnije geometrija i oblik scene koje je NeRF naučio mogu se promatrati kao objekti u prostoru, odnosno prepreke. Time se omogućava izravna integracija standardnih postupaka planiranja putanje. Sustav koji su osmislili Adamkiewicz i suradnici [72] uspješno planira putanju bez sudara tako da kažnjava slučajeve kad dolazi do sudara između oblaka točaka robota i polja gustoće volumena dobivene od NeRF modela. Sustav CATNIPS [73] koristi gustoću volumena kao skupinu točaka u kontinuiranom prostoru koje podliježe Poissonovoj distribuciji za izračun kolizija u sceni. Marza i suradnici predložili su sustav [74] koji uči *online* strukturalne i semantičke parametre. Strukturalne informacije se koriste za izbjegavanje prepreka dok se semantičke koriste za estimaciju lokacija objekata u sceni.

Manipulacija se odnosi na proces gdje je robot pokretač interakcije s okolinom. Glavna razlika naspram dosadašnjih upotreba je potreba za preciznim izračunom odnosno procjenom orijentacije i pozicije objekta. Dok je lokalizacija računala samo 6 stupnjeva slobode robota, pri manipulaciji potrebno je izračunati 6 stupnjeva slobode drugih objekata, ovisno o zadatku. Sustav NeRF-Pose [75] prvo radi rekonstrukciju objekta pomoću OBJ-NeRF modela te se onda regresijom dobivaju 6D pozicije iterativnim PnP i RANSAC

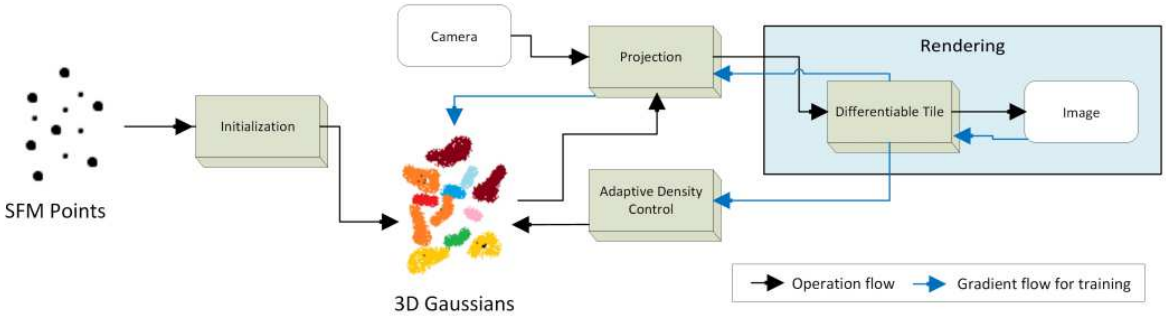
postupkom. Nadalje, NeRF modeli su zbog svog trodimenzionalog strukturalnog oblika bogatiji informacijama nego standardni dvodimenzionalni pristupi te se zato mogu iskoristiti pri raznim operativnim zadatcima[48]. Hu i suradnici su predložili sustav [76] gdje se NeRF modeli koriste za učenje ciljanih objekata no bez zadane kategorije. To rezultira generiranjem velikog broja sličnih slika koje se onda koriste pri detekciji objekata. Također, NeRF sustavi su se pokazali učinkovitijima u situacijama gdje standardni modeli i senzori ne daju prihvatljive rezultate. Sustav Dex-NeRF [77] koristi gustoću volumena kako bi ostavarili globalno konzistentnu mapu koja im omogućava manipulaciju i interakciju s prozirnim objektima, nešto što prijašnji sustavi nisu bili u mogućnosti. Naposljetku, bitno je napomenuti kako se NeRF pokazao efikasnijom reprezentacijom pri nadziranom učenju [78]. Sustav NeRF-RL [78] osmišljen je u više dijelova kako bi se ovo pokazalo. Prvo se trenira enkoder koji preslikava nekoliko opažanja na slici u skriveni prostor koji opisuje objekte u sceni. Nadalje, napravljeni dekoder ovisan o tom skrivenom prostoru koristi se kao nadzor za učenje tog skrivenog prostora. Naposljetku, algoritam nadziranog učenja onda se primjenjuje na taj prostor i služi kao reprezentacija stanja.

U ovom potpoglavlju prikazali smo teoriju i neke od sustava koji implementiraju NeRF tehnologiju. Cilj je bio pokazati postupak razvoja i motivacije za pojedine sustave te njihove nedostatke. Inspirirani time, istraživači su došli do trenutno najnovije reprezentacije scene koja zasad pokazuje obećavajuće rezultate, Gaussovih trodimenzionalnih reprezentacija.

3.2. Trodimenzionalne Gaussove reprezentacije

Trodimenzionalne Gaussove reprezentacije su nova reprezentacija scene, originalno predstavljena u radu [79], koja se koriste u novim sustavima za Gaussovo rasipanje (eng. *Gaussian splatting*). U radu ćemo nadalje koristiti izvorno ime radi lakšeg razumijevanja. *Gaussian splatting* metoda omogućava prikazivanje (eng. *rendering*) radijalnih polja i sintezu pogleda u stvarnom vremenu pritom ne koristeći nikakve neuronske mreže, a ostvarivajući jednake ili bolje rezultate nego dotad najbolji sustavi. Kao ulaz se koristi skup slika sa odgovarajućim kalibracijama kamera dobivenim od SfM (eng. *Structure-from-Motion*) što rezultira oblakom točaka. Te točke se koriste za inicijalizaciju trodimenzi-

onalnih Gaussovih sferoida koje su određene svojom pozicijom (srednjom vrijednosti), matricom kovarijance i neprozirnosti. To omogućava relativno kompaktnu,a preciznu reprezentaciju scene zato što vrlo anizotropni Gaussovi sferoidi omogućavaju detaljan prikaz scene. Po uzoru na [80], komponenta boje u pojedinim smjerovima prikazana je sfernim harmonicima. Nadalje, metoda stvara reprezentaciju radijalnog polja slijedom optimizacijskih koraka parametara trodimenzionalnih Gaussa (srednje vrijednost, matrice kovarijance i neprozirnosti) pritom koristeći tehnike za adaptivno upravljanje gustoćom tih Gaussa. Naposljetku, autori tvrde kako je ključan čimbenik efikasnosti njihove metode prikazivač baziran na regijama (eng. *tile-based renderer*) što omogućava stapanje neprozirnosti anizotropnih sfera poštivajući njihov redoslijed zahvaljujući brzom sortiranju. Slika 3.4. prikazuje arhitekturu predloženog sustava.

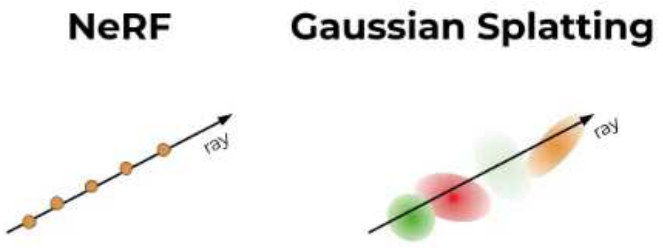


Slika 3.4. Arhitektura sustava za *Gaussian splatting*. Slika preuzeta iz [81].

Implicitne reprezentacije scene, poput NeRF, pokazale su se prikladnima za detaljnu sintezu pogleda, no nisu bile u mogućnosti to raditi u stvarnom vremenu. Takvi modeli su ostvarivali neprihvatljive rezultate što se tiče vremena izvođenja uz činjenicu da su se trebali trenirati i do nekoliko dana. Gaussove trodimenzionalne reprezentacije predstavljene su kao eksplicitna alternativa i dalje zadržavajući diferencijalna volumetrijska svojstva implicitnih reprezentacija te omogućavajući brzo stapanje neprozirnosti za prikaz. Slika 3.5. prikazuje konceptualnu razliku između NeRF-a i trodimenzionalnih Gaussa. Trodimenzionalni Gaussi definirani su punom matricom kovarijance Σ centiranom oko točke (srednje vrijednosti) μ:

$$G(x) = e^{-\frac{1}{2}(x)^T \Sigma^{-1}(x)} \tag{3.11}$$

Svojstvo neprozirnosti α koristi se koristi se u procesu stapanja, objašnjeno kasnije. S obzirom da su nam potrebne koordinate iz pogleda kamere, potrebno je trodimenzionalne



Slika 3.5. Razlike u prikazu između NeRF reprezentacija i trodimenzionalnih Gaussovih reprezentacija. Slika preuzeta sa [82]

Gausse projicirati u dvodimenzionalan prostor kamere. Sa danom matricom transformacije pogleda W, matrica kovarijance Σ' u kamerinim koordinatama glasi

$$\Sigma' = JW\Sigma W^T J^T \tag{3.12}$$

gdje J je Jakobijan afine aproksimacije projekcijske matrice. S obzirom da matrice kovarijanci trebaju biti pozitivno semidefinitne kako bi imale fizičko značenje, one nisu prikladne za reprezentaciju radijalnog polja. Naime, gradijentni spust koji se koristi za optimizaciju nije moguće ograničit da ne proizvodi takve rezultate te bi matrice postale nestabilne u samo nekoliko koraka optimizacije. Kao rješenje predložena je sljedeća pretpostavka, matrice kovarijanci trodimenzionalnih Gaussa su analogne opisivanju elipsoida. Sa danom matricom skaliranja S i matricom rotarije R možemo dobiti matricu kovarijance:

$$\Sigma = RSS^T R^T \tag{3.13}$$

Kako bi se omogućila neovisna optimizacija, matrice su spremljene odvojeno. Vrijednosti skaliranja su spremljene kao trodimenzionalni vektor $s \in R^3$ i vrijednosti rotacije kao kvaternion $q = (x, y, z, w)$. Matrica skaliranja dana vektorom s glasi

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \tag{3.14}$$

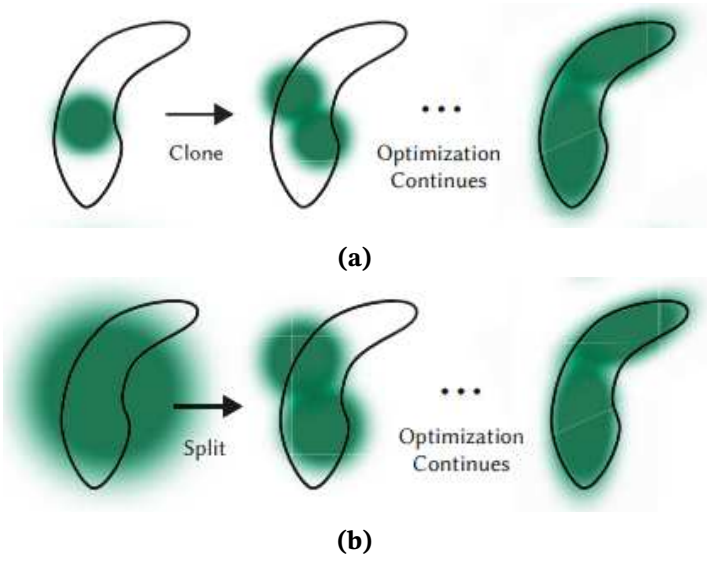
te izraz za matricu rotacije R dobivenu parametriziranim kvaternionom q glasi

$$R = \begin{bmatrix} 1-2(y^2+z^2) & 2(xy-wz) & 2(xz+wy) \\ 2(xy+wz) & 1-2(x^2-z^2) & 2(yz-wx) \\ 2(xz-wy) & 2(yz+wx) & 1-2(x^2+y^2) \end{bmatrix} \tag{3.15}$$

Optimizaciju možemo podijeliti u dva segmenta: optimizaciju parametera i adaptivno upravljanje gustoćom Gaussa. Uz pozicije μ, matrice kovarijanci Σ i neprozirnosti α optimiziraju se i koeficijenti sfernih harmonika koji predstavljaju boju Gaussa c. Ta optimizacija bazirana je na uzastopnim iteracijama prikazivanja i usporedbe rezultantne slike sa pogledima iz skupa za učenje. Po uzoru na prijašnje radove [80][83], koristi se stohastični gradijentni spust kako bi se omogućilo korištenje radnih okvira prilagođenih izvođenju na grafičkim karticama i dodavanje vlastitih CUDA jezgara. Za α se koristi sigmoidalna aktivacijska funkcija i za skalu matrice kovarijance eksponencijalna aktivacijska funckija kako bi se ograničili na intervalu [0-1) i osigurali glatki gradijenti. Inicijalna matrica kovarijacije se estimira kao isotropni Gauss sa osima jednakim prosjeku udaljenosti do tri najbliže točke. Naposljetku funkcija gubitka komibinirana sa D-SSIM [84] glasi

$$\mathcal{L} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SIMM} \tag{3.16}$$

Adaptivna kontrola gustoćom Gaussa zadužena je za bolju reprezentaciju scene. Na inicijalne točke dobivene SfM algoritmom primjenjuje se kontrola gustoćom kako bi se od rijetkih točaka dobili gusti Gaussi koji bolje opisuju scenu. Nakon početnog "zagrijavanja" sustava, Gaussi se zgušnjavaju svakih 100 iteracija i miču se transparentni Gaussi (Gaussi s α manjim od praga ϵ_α). U područjima u kojima nema geometrijskih značajki i u područjima gdje Gaussi zauzimaju većinu prostora, očitavaju se veliki prostorni gradijenti. To označava da za sustav ta područja još nisu do kraja obrađena te će pokušati micati Gausse u ta područja. Zbog toga ta područja sa gradijentima iznad praga τ_{pos} se isto zgušnjavaju Gaussima. Područja gdje nema geometrijskih značajki odnosno prazan je prostor zgušnjavaju se tako da se kloniraju već postojani Gaussi u blizini i pomiču do tamo. U područjima gdje je velik broj Gaussa sa visokim varijancama, zgušnjavanje se obavlja na način da se Gaussi podijele u dva manja Gaussa sa skalom podijeljenom



Slika 3.6. Zgušnjavanje Gaussa u: a) praznim prostorima gdje nema Gaussa klonira se već postojeći najbliži Gauss, b) prostorima gdje jedan Gaussa zauzima veći prostor on se dijeli na dva manja Gaussa. Slike preuzete iz [79].

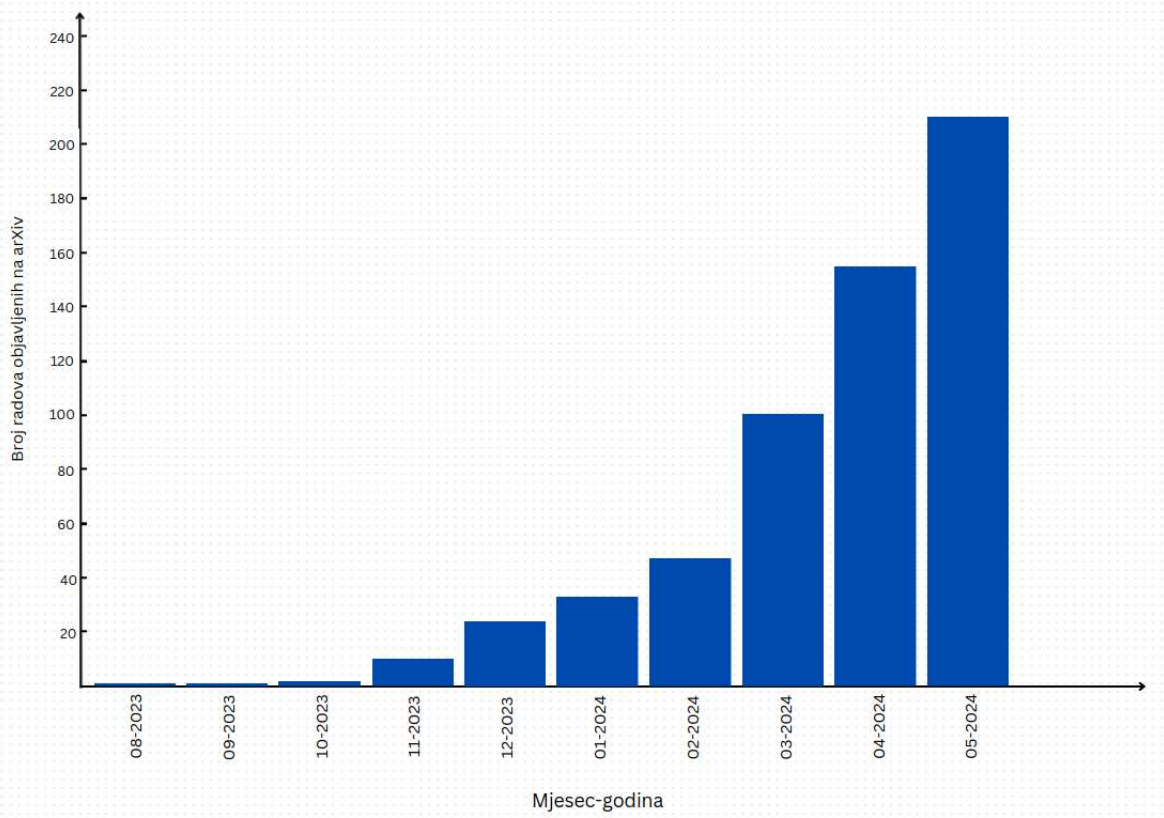
koeficijentom ϕ. U prvom slučaju dolazi do povećanja volumena i broja Gaussa, dok u drugom volumen ostaje isti no povećamo broj Gaussovih elipsoida. No u nekim prostorima, poput blizu kamere, sustav može kreirati nepotrebno velik broj Gaussa. Kako bi se spriječilo gomilanje takvih Gaussa u sceni, svakih 3000 tisuće iteracija smanjuje se α svim Gaussima blizu 0. U sljedećim koracima optimizacija će povećati α tamo gdje je potrebno i odbacit će one Gausse kojima je α ispod praga, kako je objašnjeno ranije. Proces kontrole gustoće na primitivnom primjeru možemo vidjeti na slici 3.6.

Kako bi se postiglo brzo prikazivanje odnosno rasterizacija, po uzoru na [85], kreiran je *tile-based* prikazivač. On je omogućavao brzo sortiranje primitiva cijele scene istovremeno, time izbjegavajući zahtjevno sortiranje na razini svakog piksela, te implementacije unatražne propagacije preko neograničenog broja Gaussa. Proces rasterizacije počinje podijelom ekrana u regije od 16×16 piksela te miče Gausse koji ispadaju van prostora gledanja (eng. *view frustum*), točnije svi Gaussi čije je preklapanje sa prostorom gledanja manje od 99%. Također, Gaussi na ekstremnim pozicijama odnosno Gaussi koji su preblizu ili predaleko se isto miču. Nakon toga se preostalim Gaussima dodjeljuje oznaka na temelju njihove dubine i broja regija koji zauzimaju. Ti Gaussi se onda sortiraju koristeći brzi Radix sortirajući algoritam na grafičkog kartici [86]. Nakon sortiranja, stvaraju se liste za pojedine regije gdje je prvi član najbliži Gauss, a zadnji najudaljeniji. Pri rasterizaciji pokreće se blok dretvi za svaku regiju te se onda paralelno izvršava stapanje α za

sve regije. Rasterizacija prestaje kada α u svim pikselima postane 1. Kako je to jedino ograničenje odnosno prekid rasterizacije, ne postoji ograničenje na broj Gaussa koji rasterizacija može podnijet. Konačna boja c pojedinog piksela se može dobiti na isti način kao kod NeRF metode

$$c_i = \sum_{n \leq N} c_n \alpha_n T_n \quad \text{gdje} \quad T_n = \prod_{m<n} (1 - \alpha_m) \tag{3.17}$$

Trodimenzionalne Gaussove reprezentacije su se pokazale superiornijima naspram prijašnjih metoda zbog svoje sposobnosti za preciznom i relativno brzom rekonstrukcijom scene. Kao što možemo vidjeti na slici 3.7., u posljednjih godinu dana, odnosno otkako je objavljen prvi rad [79], vidimo kontinuiran porast broja radova na tu temu. Iako je originalni sustav [79] prvi ostvario rekonstrukciju u stvarnom vremenu, to ne znači da je on bio idealan. Daljnji radovi istražili su poboljšanja za pojedine dijelove sustava u svrhu ubrzanja i što realističnijeg prikaza scene.



Slika 3.7. Broj arXiv radova objavljenih na temu *Gaussian splatting*.

Kako smo originalni sustav podijelili na tri komponente (reprezentacija trodimenzionalnim Gaussom, optimizacija i prikazivanje) tako možemo podijeliti i radove. Radovi

poput DNGaussian [87] i PixelSplat [88] su mijenjali odnosno dodavali stvari na postojeću trodimenzionalnu Gaussovu reprezentaciju, radovi poput [89] i [90] poboljšavali su prikazivanje scene i radovi poput [91] i [92] bavili su se poboljšavanjem optimizacijskog algoritma.

Sustav DNGaussian [87] je uz trodimenzionalne Gausse koristio i informacije o dubini dobivene od monokularne kamere za rektifikaciju geometrijskih oblika. To je predstavljalo i slabiju i jaču regularizaciju čime se poboljšala rekonstrukcija u situacijama sa malo ulaznih informacija odnosno broja pogleda. Nadalje, optimizacija pozicije trodimenzionalnih Gaussa nije mijenjala njihov oblik čime se osigurala ravnoteža između detaljnosti scene i koherencije geometrije. Sustavi PixelSplat [88] i SplatterImage [93] su preskočili proces optimizacije te izravno prikazivali trodimenzionalne Gausse koristeći duboke neuronske modele. PixelSplat je koristio guste vjerojatnosne distribucije za uzorkovanje Gaussa. Problem lokalnih minimuma koji nastaje koristeći regresije funkcija baziranih na primitivima je riješen djelmično tom parametrizacijom i djelomično reparametrizacijom gradijenata za vrijeme gradijentnog spusta. SplatterImage je iskoristio neuronski model kako bi mapirao ulazne dvodimenzionalne slike na korespondentne Gausse na bazi pojedinih piksela.

Sustav LightGaussian [94] koristio je kompaktniju reprezentaciju scene micajući Gausse sa malim utjecajem u scenu. Utjecaj je računat sa udjelom koliko pojedini Gauss pridodaje vrijednosti piksela za pojedinu zraku. Lee i suradnici su u svom radu [95] kreirali masku kojom se omogućava micanje nepotrebnih Gaussa, gdje se maska stvara ovisno o volumenu Gaussa i njihovim transparentnostima.

Sustavi poput [96] i [97] proširili su trodimenzionalne Gaussove reprezentacije dodavajući semantičke informacije. Shi i suradnici u svom radu [96] predstavljaju trodimenzionalne Gausse sa ugrađenim jezičnim informacijama dizajniranim za upite otvorenog vokabulara (eng. *open-vocabulary query*). Na Gausse se ugrađuju semantične značajke pritom održavajući minimalno memorijsko zauzeće te kako bi se kontrolirala semantička nekonzistencija preko različitih ulaza odnosno kamera, predlaže se zaglađivanje semantičkih značajki koje smanjuje njihovu prostornu frekvenciju. To sve zajedno rezultira detaljnom reprezentacijom i visokom preciznošću prilikom upita. Yang i suradnici u svom radu [98] su za razliku od jezičnih semantičnih značajki, iskoristili četiridimenzi-

onalne Gausse kako bi reprezentirali prostorni i vremenski volumen kao jedan entitet, ugrađujući na model temporalne značajke. Uz te značajke predložena je i strategija zaglađivanja zadužena za smanjenje pretreniranje koje se često pojavljuje pri ugrađivanju temporalnih značajki, što je u konačnici omogućilo modeliranje proizvoljnih rotacija u vremenu i prostoru održavajući detaljan prikaz.

Trenutni cjevovod za prikazivanje je primitivan te sadrži nekoliko nedostaka [99]. Na primjer, jednostavan algoritam vidljivost mogao bi rezultirati nedosljednostima u slici odnosno artefaktima, iz čega je očigledno da ima mjesta za napredak cjevovoda. Za početak, zbog svoje diskretne forme, trodimenzionalne Gaussove reprezentacije podležne problemu preklapanja (eng. *aliasing*) pri različitim rezolucijama, što rezultira zamućenjima i nejasnoćama u sceni. Yan i suradnici [89] su kao rješenje tome predložili Gausse više skali gdje su se u prikazu scene koristili trodimenzionalni Gaussi različitih veličina, dok su Yu i suradnici u svom radu [90] uveli dodatne filtere (trodimenzionalni filter zaglađivanja i dvodimenzionalni Mip filter) što je rezultiralo efikasnim uklanjanjem zamućenja. Nadalje, trodimenzionalni Gaussi inicijalno nisu rješili problem odsjaja s kojim se muči većina pristupa rekonstrukciji. Sustav GaussianShader [100] riješio je taj problem implementirajući jednostavne funkcije sjenačanja. Također implementirana je i metoda predikcije normala koje dodatno poboljšavaju sjenčanje uzimajući u obzir stvarnu geometriju scene. Naposljetku, taj problem stvarne geometrije također je prisutan kod trodimenzionalnih Gaussovih reprezentacija. Gaussi pretežito zanemaruju stvarnu geometriju scene, posebno u složenijim situacijama. Sustav ScaffoldGS [101] uveo je mrežu točaka sidrenja kako bi se organizirali lokalni Gaussi čija se svojstva, poput boje i prozirnosti, konstantno optimiziraju. To je rezultiralo poboljšanom reprezentacijom scene koja se podvrgnula stvarnoj geometriji pomoću hijerarhijske strukture geometrija u sceni.

Slično sustavu ScaffoldGS, GeoGaussian [91] također čuva globalnu konzistentnost geometrije scene, no za razliku od mreže točaka sidrenja, Gaussima se dodaju geometrična svojstva. Ta geometična svojstva kasnije su optimizirana potičući Gausse u susjedstvu da budu komplanarni čime se ostvaruje bolja kvaliteta rekonstrukcije scene. Sustav FreGS [92] je u optimizaciju uveo regularizaciju baziranu na frekvenciji. Točnije, FreGS ostvaruje grubo i fino zgušnjavanje Gaussa koristeći komponente niskih i visokih frekvencija. Kako bi se pronašle i razlikovale te komponente koriste se filteri visokog i niskog

propusta u Fouriverovom prostoru. Sustav koji su napravili Chung i suradnici [102] je umjesto frekvencije koristio dubinu pri regularizaciji. Dubine se dobivaju od monokularnog modela estimacije dubine te se one prilagođavaju skali dubine pomoću SfM točaka. Sustav je osmišljen za situacije s nedostatkom ulaznim informacija te u tim situacijama funkcionira bolje nego drugi postojeći sustavi, rekonstuirajući relativno točnu geometriju iz samo nekoliko kadrova.

S obzirom na svoja svojstva i fleksibilnost, trodimenzionalne Gaussove reprezentacije imale su utjecaj u mnogo različitih područja istraživanja. Sustavi poput sustava od Lia i suradnika [103] te sustav Caia i suradnika [104] pokazali su njihovu primjenu na rendgenskim i CT snimkama, sustavi EndoGSLAM [105] i EndoGaussian [106] primjenili su ih za rekonstrukcije scene pri endoskopskim operacijama, sustavi DreamGaussian [107] i GaussianAvatars [108] koristili su ih za efikasno generiranje trodimenzionalnih proizvoljnih modela i naposljetku sustavi poput SplaTAM [109], MonoGS[110] i GS-SLAM [111] prikazali su njihovu prednost u SLAM sustavima. U okviru ovog rada, posljednja stavka je od najvećeg značaja te ćemo ju detaljnije objasnit i proučit u sljedećem poglavlju.

4. SLAM pomoću Gaussovih trodimenzionalnih reprezentacija

Kao što smo već spomenuli u poglavlju 2., SLAM sustavi su zbog kompleksnosti i obujma zadatka vrlo zahtjevni za izvođenje. S obzirom da je cilj ostvariti SLAM sustave koji se mogu izvodit u stvarnom vremenu, Gaussove trodimenzionalne reprezentacije pokazale su se prikladnima zbog svoje efikasnosti i kvalitete rekonstrukcije scene. U ovom poglavlju ćemo prikazati detaljno sustav MonoGS [110] i objasnit svaku pojedino komponentu.

Sustav MonoGS [110] predstavljen je kao prvi *online* vizualni SLAM sustav koji koristi Gaussove trodimenzionalne reprezentacije. Gaussove trodimenzionalne reprezentacija predstavljaju kombinaciju prijašnjih reprezentacije, imaju efikasnost, dobru lokalizaciju i mogućnost modifikacije poput točaka dok su također i kontinuirana i diferencijabilna volumna reprezentacija poput NeRF-a. Kako bi omogućili *online* SLAM, uvedeni su izračun Jakobijana pozicije kamere u zatvorenom obliku koristeći Lie algebru u odnosu na trodimenzionalnu Gaussovu mapu, regularizacija Gaussovih izotropičnih oblika čime se osigurava geometrijska konzistencija te alokacija resursa trodimenzionalnih Gaussa i nova metoda micanja tih Gaussa kako bi se osigurala precizna rekonstrukcija i praćenje kamere. Važno je naglasiti kako sustav za ulazne podatke može koristiti RGB ili RGB-D kamere koje omogućavaju očitavanje dubine.

Gaussove trodimenzionalne reprezentacije prikazuju scenu velikim skupom trodimenzionalnih Gaussa $\mathcal{G}$ koji izgledaju kao elipsoidi. Svaki Gauss $\mathcal{G}^i$ sadrži svojstva boje c^i i neprozirnosti α^i. Srednja vrijednost μ^i i kovarijanca Σ^i_W određena u globalnom koordinatnom prostoru W predstavljaju poziciju Gaussa i izgled njegovog pripadnog elipsoida. S obzirom da su trodimenzionalni Gaussi i volumna reprezentacija, nije potrebna eksplicitna ekstrakcija površine. Boja pojedinog piksela se računa isto kao i u jednadžbi

3.17:

$$C_p = \sum_{i \in \mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{4.1}$$

gdje $\mathcal{N}$ predstavlja skup Gaussa koji opisuju taj piksel. Tijekom rasterizacije utjecaj α se raspada korištenjem Gaussove funkcije dobivene stlačivanjem trodimenzionalnog Gaussa u dvodimenzionalni prostor. Trodimenzionalni Gaussi $\mathcal{N}(\mu_W, \Sigma_W)$ se preslikavaju u dvodimenzionalne Gausse $\mathcal{N}(\mu_I, \Sigma_I)$ projekcijskom transformacijom:

$$\boldsymbol{\mu}_I = \pi(\boldsymbol{T}_{CW} \cdot \boldsymbol{\mu}_W), \boldsymbol{\Sigma}_I = \mathbf{JW\Sigma}_W \mathbf{W}^T \mathbf{J}^T, \tag{4.2}$$

gdje π je projekcijska funkcija, $\boldsymbol{T}_{CW} \in \mathbf{SE}(3)$ je pozicija kamere u njenim koordinatama, $\mathbf{J}$ je Jakobijan linearne aproksimacije projekcijske transformacije i $\boldsymbol{W}$ je rotacijska komponenta $\boldsymbol{T}_{CW}$. Takva formulacija omogućava deriviranje dok stapanje Gaussa omogućava tok gradijenta. Za poboljšanje optičkih i geometrijskih svojstava Gaussa koristi se gradijent prvog reda.

Kako bi se poboljšale performanse, originalni *gaussian splatting* sustav implementirao je rasterizaciju pomoću CUDA jezgara gdje se derivacije svih parametara računaju eksplicitno. Sličnim principom, u sustavu MonoGS se Jakobijani kamere također računaju eksplicitno. Autori tvrde kako su predstavili prvi analitički Jakobijan pozicije kamere u odnosu na trodimenzionalne Gausse. Koristeći Lie algebru deriviraju se minimalni Jakobijani, pritom osiguravajući istu dimenzionalnost Jakobijana i stupnjeva slobode kako bi se eliminirali nepotrebni izračuni. Koristeći lančano pravilo derivacije, iz jednadžbe 4.2 možemo izraziti

$$\frac{\partial \boldsymbol{\mu}_I}{\partial \boldsymbol{T}_{CW}} = \frac{\partial \boldsymbol{\mu}_I}{\partial \boldsymbol{\mu}_C} \frac{\mathcal{D}\boldsymbol{\mu}_C}{\mathcal{D}\boldsymbol{T}_{CW}}, \tag{4.3}$$

$$\frac{\partial \boldsymbol{\Sigma}_I}{\partial \boldsymbol{T}_{CW}} = \frac{\partial \boldsymbol{\Sigma}_I}{\partial \mathbf{J}} \frac{\partial \mathbf{J}}{\partial \boldsymbol{\mu}_C} \frac{\mathcal{D}\boldsymbol{\mu}_C}{\mathcal{D}\boldsymbol{T}_{CW}} + \frac{\partial \boldsymbol{\Sigma}_I}{\partial \boldsymbol{W}} \frac{\mathcal{D}\boldsymbol{W}}{\mathcal{D}\boldsymbol{T}_{CW}}. \tag{4.4}$$

Koristeći derivacie na razdjelniku dobivamo minimalnu parametrizaciju te koristeći notaciju gdje $\boldsymbol{T} \in \mathbf{SE}(3)$ i $\tau \in \mathbf{se}(3)$ definiramo parcijalnu derivaciju na razdjelniku sljedećim limesom:

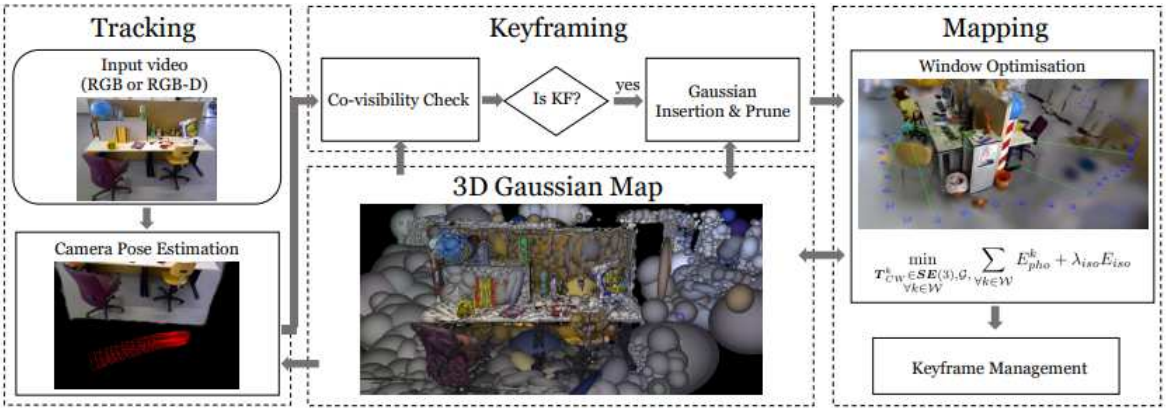
$$\frac{\mathcal{D}f(\boldsymbol{T})}{\mathcal{D}\boldsymbol{T}} \triangleq \lim_{\tau \longrightarrow 0} \frac{Log(f(Exp(\tau) \circ \boldsymbol{T}) \circ f(\boldsymbol{T})^{-1})}{\tau}, \tag{4.5}$$

gdje su Log i Exp logaritamska i eksponencijalna mapiranja između Lie grupe i Lie algebre i $\circ$ predstavlja kompoziciju grupa. Pomoću toga dalje deriviramo

$$\frac{\mathcal{D}\boldsymbol{\mu}_C}{\mathcal{D}\boldsymbol{T}_{CW}} = \begin{bmatrix} I & -\boldsymbol{\mu}_C^{\times} \end{bmatrix}, \quad \frac{\mathcal{D}\mathbf{W}}{\mathcal{D}\boldsymbol{T}_{CW}} = \begin{bmatrix} 0 & -\mathbf{W}_{:,1}^{\times} \\ 0 & -\mathbf{W}_{:,2}^{\times} \\ 0 & -\mathbf{W}_{:,3}^{\times} \end{bmatrix}, \tag{4.6}$$

gdje $^{\times}$ predstavlja kosu simetričnu matricu trodimenzionalnog vektora i $\mathbf{W}_{:,i}$ predstavlja i-ti stupac te matrice.

Sad je vrijeme da objasnimo SLAM komponente sustava: praćenje kamere, određivanje ključnih kadrova i mapiranje. Arhitekturu cijelog sustav možemo vidjeti na 4.1.



Slika 4.1. Arhitektura MonoGS sustava. Slika preuzeta iz [110].

Tijekom praćenja samo se trenutna pozicija kamere optimizira, ne dirajući reprezentaciju mape. U monokularnom slučaju minimizira se fotometrični rezidual

$$E_{pho} = ||I(\mathcal{G}, \boldsymbol{T}_{CW}) - \bar{I}||_1, \tag{4.7}$$

gdje $I(\mathcal{G}, \boldsymbol{T}_{CW})$ prikazuje Gausse $\mathcal{G}$ sa pozicije $\boldsymbol{T}_{CW}$ i $\bar{I}$ predstavlja stvarnu promatranu sliku. U slučaju kad imamo opažanja dubine definiramo geometrijski rezidual

$$E_{geo} = ||D(\mathcal{G}, \boldsymbol{T}_{CW}) - \overline{D}||_1, \tag{4.8}$$

gdje $D(\mathcal{G}, \boldsymbol{T}_{CW}1)$ je rasterizacije dubine i $\overline{D}$ je stvarna dubina. Umjesto da se stvarna dubina koristi samo za inicijalizaciju Gaussa, minimizira se i geometrijski rezidual $\lambda_{pho}E_{pho}+$

$(1 - \lambda_{pho})E_{geo}$ gdje λ_{pho} je hiperparametar. Uz to se dodatno optimiziraju parametri svijetline za scene sa različitim *exposure*-om i kažnjavamo piksele koji ne predstavljaju rub ili su visoke neprozirnosti.

Drugi dio sustava je modul zadužen za određivanje ključnih kadrova. On uz analitički Jakobijan pozicije kamere igra ključnu ulogu u performansama sustava. Kako bi praćenje i uspoređivanje svih kadrova bilo računarski prezahtjevno i skupo, za izračune se održava manji podskup kadrova $\mathcal{W}_k$ sačinjen od samo rijetkih kadrova odabranih gledajući njihovu vidljivost preko nekoliko kadrova. Cijeli proces određivanja ključnih kadrova možemo podijeliti u tri dijela: selekcija i upravljanje, vidljivost Gaussa između kadrova i njihovo dodavanje odnosno micanje. U selekciji gleda se vidljivost preko više kadrova gledajući presjek unije (IOU) i koeficijent preklapanja (OC) između trenutnog kadra i i posljednjeg ključnog kadra j

$$IOU_{cov} = \frac{|\mathcal{G}_i^v \cup \mathcal{G}_j^v|}{|\mathcal{G}_i^v \cap \mathcal{G}_j^v|}, \tag{4.9}$$

$$OC_{cov} = \frac{|\mathcal{G}_i^v \cup \mathcal{G}_j^v|}{min(|\mathcal{G}_i^v|, |\mathcal{G}_j^v|)}, \tag{4.10}$$

gdje $\mathcal{G}_i^v$ predstavlja Gausse vidljive u ključnom kadru i. Novi kadar i postaje ključni kadar ako sveukupna vidljivost preko više kadrova padne ispod određenog praga ili ako je relativna translacija t_{ij} dovoljno velika u odnosu na medijan dubine scene. S obzirom da se čuva samo manji broj kadrova radi efikasnosti, potrebno je i micat ključne kadrove. Ključni kadar se miče iz skupa ključnih kadrova ako prijašnje spomenut koeficijent preklapanja padne ispod određenog praga. Vidimo kako efikasna procjena vidljivosti preko više kadrova može uvelike pojednostavit selekciju i upravljanje ključnim kadrovima. Trodimenzionalni Gaussi su prigodni za takvu procjenu jer zbog svojstva redanja po vidljivosti automatski rješavaju probleme s okluzijama. Gauss se smatra vidljivim ako se prikazuje u procesu rasterizacije i ako sveukupna neprozirnost α nije prešla 0.5 za tu zraku. Naposljetku, u svakom ključnom kadru se dodaju novi Gaussi kako bi se mogli prikazati prijašnje neviđeni dijelovi scene. Novi Gaussi se inicijaliziraju ovisno o tome je li dostupna dubina. Ako imamo dubinu, Gaussi se inicijaliziraju sa srednjom vrijednosti μ_W postavljenom na tu dubinu sa malom varijancom Σ_W. U slučaju kad dubina nije dostupna, Gaussi imaju srednju vrijednost μ_W postavljenu na medijan dubine

sa velikom varijancom Σ. S obzirom da u monokularnom slučaju većina tih novih Gaussa nije točna, sama optimizacija nije dovoljna da ih sve ukloni. Zato se uvelo dodatno uklanjanje Gaussa na način da ako Gaussi opaženi u posljednja tri ključna kadra nisu opaženi u barem tri druga kadra, oni miču.

Naposljetku imamo i modul zadužen za mapiranje čije su zadaće održavanje koherentne trodimenzionalne strukture scene i optimiziranje novih Gaussa. Trenutni ključni kadrovi korsite se za rekonstrukciju trenutno vidljivih regija dok se nasumično odabrana dva prethodna ključna kadra koriste kako bi se zadržala globalna mapa. Rasterizacija trodimenzionalnih Gaussa nema ograničenja, no za vrijeme dinamičnih scena poput naše primjene vizualnog SLAMA dolazi do artefakta. Kako bi se to spriječilo, u mapiranje je dodana izotropna regularizacija

$$E_{iso} = \sum_{i=1}^{|\mathcal{G}|} ||\boldsymbol{s}_i - \tilde{\boldsymbol{s}}_i \cdot \mathbf{1}||_1. \tag{4.11}$$

Regularizacija sprječava artefakte kažnjavajući parametre skaliranja $\boldsymbol{s}_i$ u odnosu na njihovu srednju vrijednost $\tilde{\boldsymbol{s}}_i$ što potiče sferičnost Gaussa i sprječava njihovo izduženje duž osi. Naposljetku, konačan oblik problema mapiranja glasi

$$\min_{\boldsymbol{T}_{CW}^k \in \mathbf{SE}(3), \mathcal{G}, \forall k \in \mathcal{W}} \sum_{\forall k \in \mathcal{W}} E_{pho}^k + \lambda_{iso} E_{iso}, \tag{4.12}$$

gdje $\mathcal{W}$ predstavlja uniju svih ključnih kadrova $\mathcal{W}_k$ i nasumično odabranih kadrova $\mathcal{W}_r$. U slučaju ako imamo dostupnu dubinu, problem mapiranja se preformuilira sukladno jednadžbi 4.8

$$\min_{\boldsymbol{T}_{CW}^k \in \mathbf{SE}(3), \mathcal{G}, \forall k \in \mathcal{W}} \sum_{\forall k \in \mathcal{W}} \lambda_{pho} E_{pho}^k + (1 - \lambda_{pho}) E_{geo} + \lambda_{iso} E_{iso} \tag{4.13}$$

U konačnici je bitno napomenuti da se i za optimizaciju pozicija kamera i za optimizaciju parametara Gaussa koristi Adam optimizator [112].

Detaljno smo prikazali teorijsku pozadinu MonoGS sustava te rezultati dobiveni u eksperimentima koristeći taj sustav prikazani su u poglavlju 5.

5. Rezultati

Rezultate dobivene testirajući MonoGS sustav prikazat ćemo kvalitativno i kvantitativno kako bi prikazali efikasnost *Gaussian splatting*-a i fotorealističnost rekonstrukcije. Prikazat ćemo rad sustava na TUM RGB-D skupu podataka [113] u monokularnom i RGB-D načinu rada i na Replica skupu podataka [114] no samo sa RGB-D kamerom. Naposljetku, pokazat ćemo par kadrova iz rekonstrukcije kako bi kvalitativno prokomentirali rekonstrukciju te pokazali moguće probleme poput nagle promjene osvjetljenja ili dinamičnih objekata.

Sustav smo testirali na računalu sa AMD Ryzen Threadripper 3970x procesorom sa 32 jezgre i 64 dretve, 64 GB RAM memorije te NVIDIA RTX A5000 i NVIDIA Quadro GV100 grafičkim karticama sa Linux Ubuntu 20.04 operacijskim sustavom.

5.1. Kvantitativni rezultati

U okviru kvantitativnih rezultata prikazat ćemo srednji kvadartni gubitak odnosno RMSE (eng. *Root Mean Square Error*) u odnosu na apsolutnu pogrešku trajektorije (ATE, eng. *Absolute Trajectory Error*), te PSNR (eng. *Peak Signal-to-Noise Ratio*), SSIM (eng. *Structural Similarity Index Measure*) i LPIPS (eng. *Learned Perceptual Image Patch Similarity*) kao fotometrično mjerilo kvalitete scene. Ta mjerila za fotometričnost očitavamo prije i poslije optimizacije, kako bi se uvjerili da je precizna i efikasna. Sva mjerenja dobivena na TUM skupu podataka prikazana su tablicom 5.1. koja prikazuje rezultate korištenjem monokularne kamere i tablicom 5.2. koja prikazuje rezultate korištenjem RGB-D kamere. Vrijednosti RMSE su prikazane u metrima, dok su PSNR, SSIM i LPIPS prikazane svojim srednjim vrijednostima. Iz rezultata je moguće zaključit da je sustav efikasniji pri dostupnim informacija o dubini te se uvelike pospješuje lokalizacija i rekonstrukcija trajektorije s njima. Također, primjećujemo da je optimizacije efikasnija u sustavima sa

		Prije optimizacije			Poslije optimizacije		
	RMSE	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
TUM Desk	0.0464	17.1999	0.6476	0.3980	21.0019	0.7062	0.3572
TUM XYZ	0.0472	15.2674	0.6370	0.3742	22.2364	0.7211	0.2926
TUM Office	0.0396	19.0394	0.7219	0.3549	21.7356	0.7535	0.3665

Tablica 5.1. Kvantitativni rezultati MonoGS sustava na TUM skupu podataka u slučaju sa monokularnom kamerom.

		Prije optimizacije			Poslije optimizacije		
	RMSE	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
TUM Desk	0.0154	18.9951	0.7187	0.3130	23.5580	0.7860	0.2457
TUM XYZ	0.0120	15.8413	0.7076	0.3140	24.7487	0.7982	0.2155
TUM Office	0.0156	18.7292	0.7287	0.3464	24.7669	0.8340	0.2127

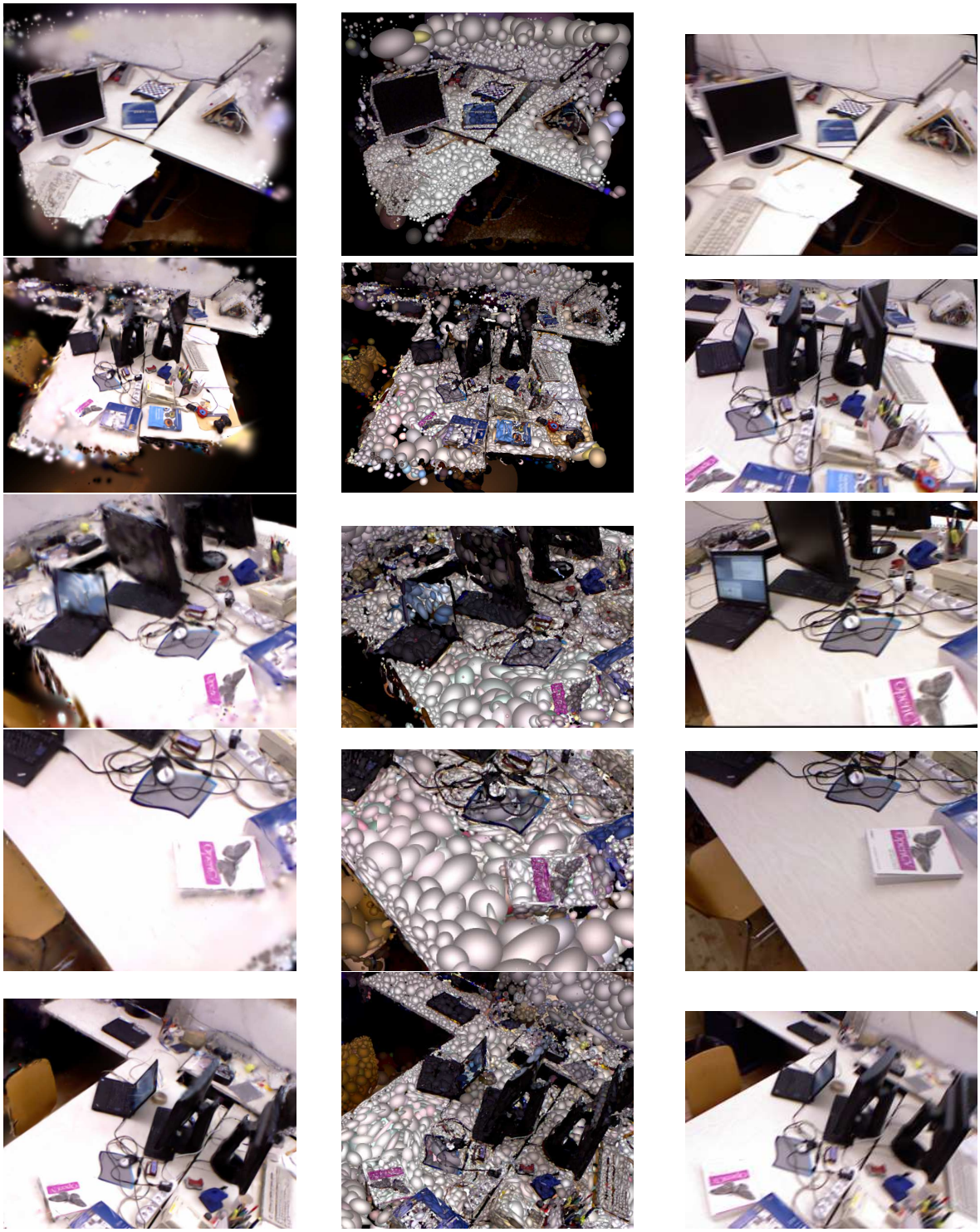
Tablica 5.2. Kvantitativni rezultati MonoGS sustava na TUM skupu podataka u slučaju sa RGB-D kamerom.

dostupnim dubinama, odnosno veća je razlika između fotometričnih metrika prije i poslije optimizacije prilikom korišetnja RGB-D kamere. Prilikom svih testiranja dobivali smo 1 do 2 FPS-a, neovisno o skupu podataka. Vidimo da je prijašnje opisan postupak optimizacije efikasan što se tiče kvantitavnih mjerila. Daljnja zapažanja i rezultate prikazat ćemo kvalitativno.

5.2. Kvalitativni rezultati

U okviru kvalitativnih rezultata prikazat ćemo kadrove unutar scene i odgovarajuće rekonstrukcije za skup podataka TUM Desk i tijekom rekonstrukcije u stvarnom vremenu sa Intel RealSense RGB-D kamerom. Na slici 5.1. možemo vidjeti rekonstrukciju sustava u lijevom stupcu, prikaz rekonstrukcije bez stapanja α, točnije prikaz samih Gaussovih elipsoida u sredini, te u desnom stupcu vidimo stvarne slike. Prikazani su isječci iz nekoliko dijelova scene, na početku, u sredini i na kraju. Iako teže uočljivo na ovim slikama zbog smanjene rezolucije, u početku smo mogli vidjeti zamućenja u rekonstrukciji, posebno kod dijelova kod kojih nije prošlo zgušnjavanje Gaussa ili u dijelovima visoke razine detalja. Nakon određenog vremena sustav izoštri rekonstrukciju i u kasnijim slikama možemo vidjeti prihvatljivu razinu detalja kako bi smo rekonstrukciju mogli smatrat fotorealističnom.

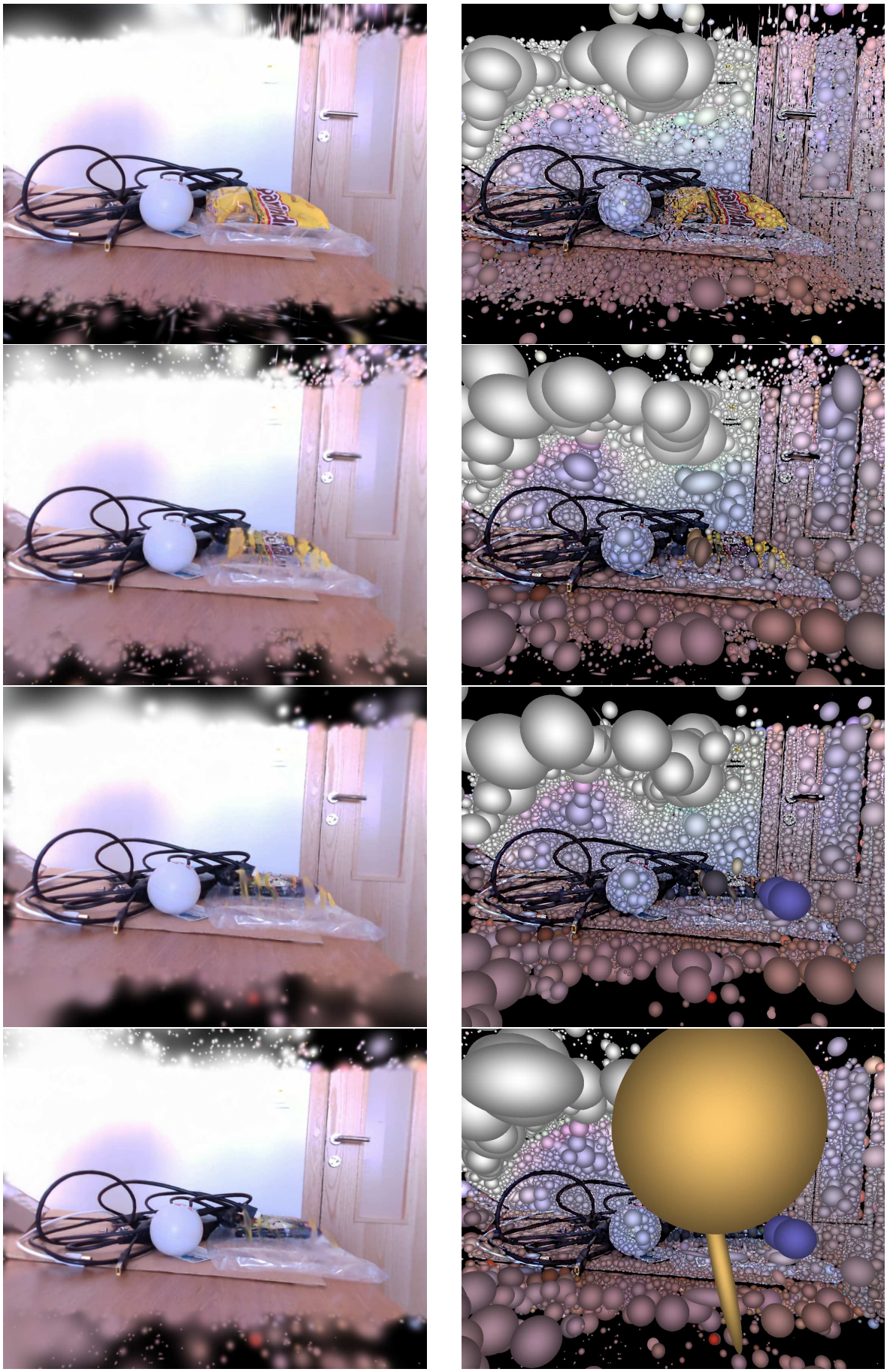
Naposljetku, prikazat ćemo rezultate dobivene u stvarnomv vremenu koristeći In-



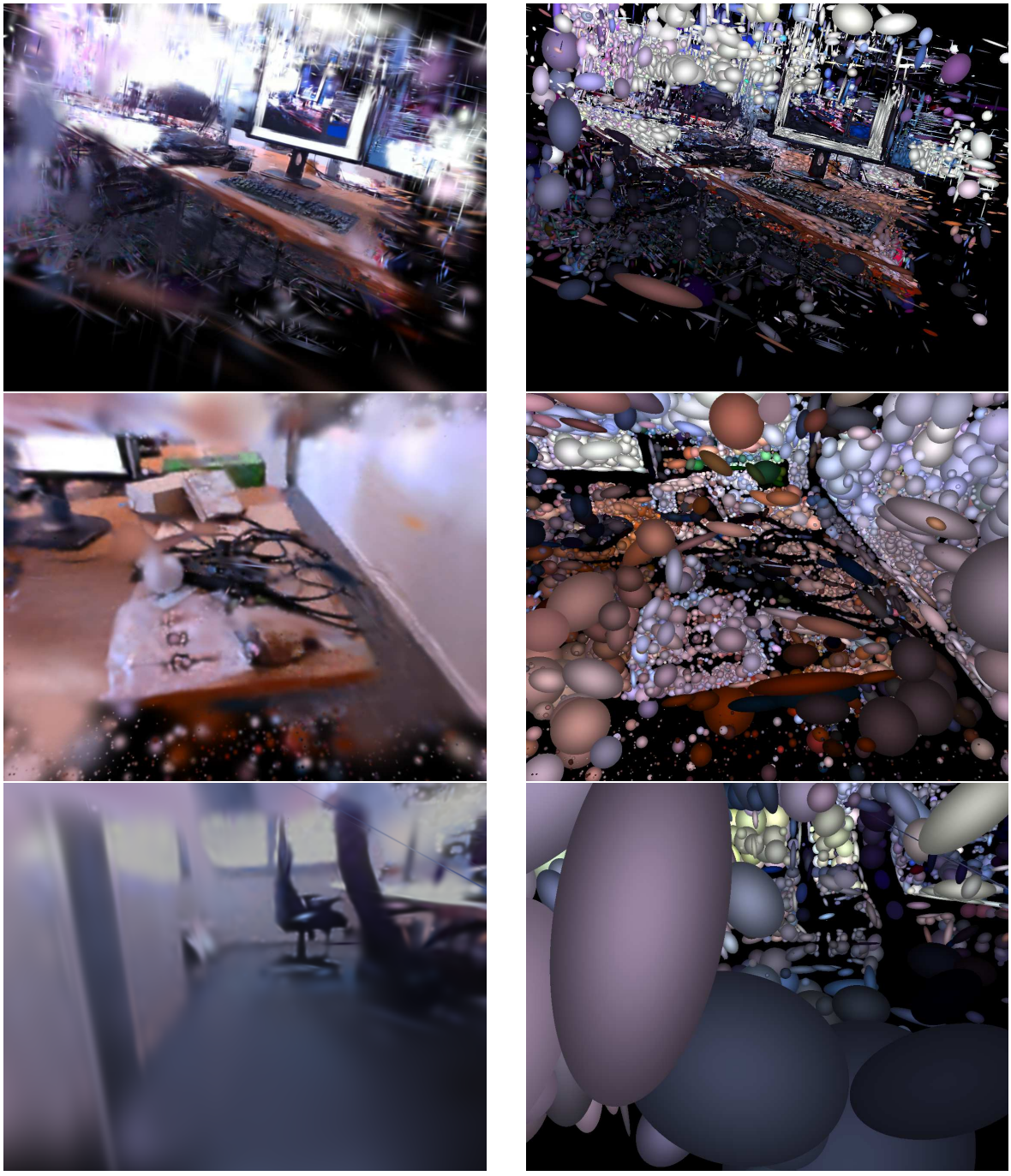
Slika 5.1. Kvalitativni rezultati MonoGS sustava na TUM Desk skupu podataka. Po stupcima idu redom: rekonstrukcija, prikaz Gaussovih elipsoida, stvarna slika.

tel RealSense D435i RGBD kameru. Na slici 5.2. možemo vidjeti kako MonoGS sustav ostvaruje fotorealističnu rekonstrukciju, no u stabilnim uvjetima. Na slikama se može uočiti da sustav ima problema sa direktnom svijetlinom. Naime, to je problem više do načina na koji sama kamera prima podatke, te se može ukloniti dodavanjem *auto exposure* algoritma naknadno. Za rješavanje tog problema u stvarnom vremenu trebala bi se uvesti drukčija provjera, jedan pristup bi mogao bit semantička segmentacija te onda Gausse koji predstavljaju takvo svijetlo u sceni prigušit. Nadalje, sustav ima problema i sa dinamičkim objektima u sceni. U slici 5.3. vidimo primjer što se dogodi ako se makne objekt iz scene koji je sustav inicijalno smatrao da je dio statične scene. Između prvog i drugog kadra uklonili smo žutu vrećicu. Sustav je i dalje nakon micanja objekta nastavio ga tamo prikazivat, u početku je objekt samo postao naizgled proziran no nakon par minuta ostali su samo artefakti koji se nikad nisu potpuno pročitstili. Konačno, sustav ima poteškoća prilikom loše inicijalizacije, odnosno dinamičnih pokreta kamere. Prilikom inicijalizacije, fotorealističnost sustava uvelike ovisi o statičnosti kamere. U slici 5.4. možemo vidjeti primjere dobivene dinamičnim pomicanjem kamere. Vidimo da sustav ne stigne efikasno optimizirat Gausse kako bi ostvarila fotorealističnost, te vidimo i velik broj nasumičnih Gaussa koji još nisu pročišćeni. Također, prilikom veće promjene orijetacije kamere, sustav se resetira i ispočetka započinje proces inicijalizacije što uvelike usporava izvođenje.

Slika 5.2. Rezultati MonoGS sustava dobiveni u stvarnom vremenu koristeći Intel RealSense D435i RGBD kameru. Po stupcima idu redom: rekonstrukcija, prikaz Gaussovih elipsoida, stvarna slika.



Slika 5.3. Primjer kad se iz scene makne objekt koji je inicijalno smatran kao dio statične scene. Između prvog i drugog retka, odnosno kadra, maknuta je žuta vrećica te vidimo da i nakon određenog vremena sustav i dalje čuva stare Gausse koji stvaraju artefakte.



Slika 5.4. Primjer rekonstrukcije prilikom dinamičnih pokreta kamere.

6. Zaključak

Kroz rad pokazali smo osnove SLAM sustava, pozadinu reprezentacija scene i SLAM sustav sa integriranim Gaussovim trodimenzionalnim reprezentacijama. Cilj rada je bio prikazati prednosti Gaussovih trodimenzionalnih reprezentacija naspram prethodnih reprezentacija, koristeći njihova svojsta kontinuranosti i diferencijabilnosti kako bi ostvarili fotorealističnu rekonstrukciju u stvarnom vremenu.

Proučavajući teorijsku pozadinu te onda eksprimente sa sustavom MonoGS, kao prvi SLAM sustav sa Gaussovim reprezentacijama, utvrdili smo da su Gaussove reprezentacije robustnije i efikasnije pri rekonstrukciji u stvarnom vremenu. Prethodni sustavi uvijek su morali raditi izbor između fotorealističnosti ili efikasnosti odnosno brzine izvođenja, no trodimenzionalne Gaussove reprezentacije predstavljaju korak prema fotorealističnoj rekonstrukciji u stavrnom vremenu.

Iako je takav pristup tek nedavno predložen, vidimo velik broj istraživača u tom području koji neprestano predlažu bolje i naprednije sustave. Široko područje primjene omogućava i pristup problemu sa više stajališta te otvara put za novim rješenjima. Zbog svih svojstava, možemo reći da su Gaussove trodimenzionalne reprezentacije značajan korak u području SLAM istraživanja i općenito u području računalnog vida i grafike.

Literatura

[1] K. Di, W. Wan, H. Zhao, Z. Liu, R. Wang, i F. Zhang, "Progress and applications of visual slam", *Journal of Geodesy and Geoinformation Science*, sv. 2, br. 2, str. 38, 2019.

[2] A. Tourani, H. Bavle, J. L. Sanchez-Lopez, i H. Voos, "Visual slam: What are the current trends and what to expect?" *Sensors*, sv. 22, br. 23, str. 92–97, studeni 2022. https://doi.org/10.3390/s22239297

[3] X. Gao, T. Zhang, Y. Liu, i Q. Yan, "14 lectures on visual slam: from theory to practice", *Publishing House of Electronics Industry*, str. 206–234, 2017.

[4] Sonya Kelley, https://www.allaboutvision.com/eye-care/eye-anatomy/optical-illusions-and-the-human-eye/, [mrežno; stranica posjećena: lipanj 2024.].

[5] C. Harris, M. Stephens *et al.*, "A combined corner and edge detector", u *Alvey vision conference*, sv. 15, br. 50. Citeseer, 1988., str. 10–5244.

[6] H. Bay, A. Ess, T. Tuytelaars, i L. Van Gool, "Speeded-up robust features (surf)", *Computer vision and image understanding*, sv. 110, br. 3, str. 346–359, 2008.

[7] E. Rublee, V. Rabaud, K. Konolige, i G. Bradski, "Orb: An efficient alternative to sift or surf", u *2011 International conference on computer vision*. Ieee, 2011., str. 2564–2571.

[8] N. Yang, R. Wang, i D. Cremers, "Feature-based or direct: An evaluation of monocular visual odometry", 05 2017.

[9] R. A. Newcombe, S. J. Lovegrove, i A. J. Davison, "Dtam: Dense tracking and mapping in real-time", u *2011 international conference on computer vision*. IEEE, 2011., str. 2320–2327.

[10] J. Engel, T. Schöps, i D. Cremers, "Lsd-slam: Large-scale direct monocular slam", u *European conference on computer vision*. Springer, 2014., str. 834–849.

[11] J. Engel, V. Koltun, i D. Cremers, "Direct sparse odometry", *IEEE transactions on pattern analysis and machine intelligence*, sv. 40, br. 3, str. 611–625, 2017.

[12] S. Park, T. Schoeps, i M. Pollefeys, "Illumination change robustness in direct visual slam", 05 2017., str. 4523–4530. https://doi.org/10.1109/ICRA.2017.7989525

[13] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems [j]", *Journal of basic Engineering*, sv. 82, br. 1, str. 35–45, 1960.

[14] R. E. Kalman i R. S. Bucy, "New results in linear filtering and prediction theory", *Journal of basic Engineering*, sv. 83, br. 1, str. 95–108, 1961.

[15] S. Xu, T. Wang, C. Lang, S. Feng, i Y. Jin, "Graph-based visual odometry for vslam", *Industrial Robot: An International Journal*, sv. 45, br. 5, str. 679–687, 2018.

[16] R. Duan, Y. Feng, i C.-Y. Wen, "Deep pose graph-matching-based loop closure detection for semantic visual slam", *Sustainability*, sv. 14, br. 19, str. 11864, 2022.

[17] N. Merrill i G. Huang, "Lightweight unsupervised deep loop closure", *arXiv preprint arXiv:1805.07703*, 2018.

[18] B. Chen, D. Yuan, C. Liu, i Q. Wu, "Loop closure detection based on multi-scale deep feature fusion", *Applied Sciences*, sv. 9, br. 6, str. 1120, 2019.

[19] K. A. Tsintotas, L. Bampis, i A. Gasteratos, "The revisiting problem in simultaneous localization and mapping: A survey on visual loop closure detection", *IEEE Transactions on Intelligent Transportation Systems*, sv. 23, br. 11, str. 19 929–19 953, 2022.

[20] S. Gupta, T. Guadagnino, B. Mersch, I. Vizzo, i C. Stachniss, "Effectively detecting loop closures using point cloud density maps", u *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[21] Y. Zhou, Y. Wang, F. Poiesi, Q. Qin, i Y. Wan, "Loop closure detection using local 3d deep descriptors", *IEEE Robotics and Automation Letters*, sv. 7, br. 3, str. 6335–6342, 2022.

[22] Nicollo Valigi, https://nicolovaligi.com/articles/bag-of-words-loop-closure-visual-slam/, [mrežno; stranica posjećena: lipanj 2024.].

[23] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, i W. Yu, "Structslam: Visual slam with building structure lines", *IEEE Transactions on Vehicular Technology*, sv. 64, br. 4, str. 1364–1375, 2015. https://doi.org/10.1109/TVT.2015.2388780

[24] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, i F. Moreno-Noguer, "Pl-slam: Real-time monocular visual slam with points and lines", 06 2017. https://doi.org/10.1109/ICRA.2017.7989522

[25] H. M. S. Bruno i E. L. Colombini, "Lift-slam: A deep-learning feature-based monocular visual slam method", *Neurocomputing*, sv. 455, str. 97–110, 2021. https://doi.org/https://doi.org/10.1016/j.neucom.2021.05.027

[26] R. Mur-Artal, J. M. M. Montiel, i J. D. Tardos, "Orb-slam: A versatile and accurate monocular slam system", *IEEE Transactions on Robotics*, sv. 31, br. 5, str. 1147–1163, listopad 2015. https://doi.org/10.1109/tro.2015.2463671

[27] Q. Peng, Z. Xiang, Y. Fan, T. Zhao, i X. Zhao, "Rwt-slam: Robust visual slam for highly weak-textured environments", 2022.

[28] J. Sun, Z. Shen, Y. Wang, H. Bao, i X. Zhou, "Loftr: Detector-free local feature matching with transformers", u *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021., str. 8922–8931.

[29] A. Steenbeek i F. Nex, "Cnn-based dense monocular visual slam for real-time uav exploration in emergency conditions", *Drones*, sv. 6, br. 3, 2022. https://doi.org/10.3390/drones6030079

[30] K. He, X. Zhang, S. Ren, i J. Sun, “Deep residual learning for image recognition”, u *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016., str. 770–778. https://doi.org/10.1109/CVPR.2016.90

[31] L. Xiao, J. Wang, X. Qiu, Z. Rong, i X. Zou, “Dynamic-slam: Semantic monocular visual localization and mapping based on deep learning in dynamic environment”, *Robotics and Autonomous Systems*, sv. 117, str. 1–16, 2019. https://doi.org/https://doi.org/10.1016/j.robot.2019.03.012

[32] S. Wang, R. Clark, H. Wen, i N. Trigoni, “Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks”, u *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, svibanj 2017. https://doi.org/10.1109/icra.2017.7989236

[33] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, i T. Brox, “Flownet: Learning optical flow with convolutional networks”, 2015.

[34] S. Milz, G. Arbeiter, C. Witt, B. Abdallah, i S. Yogamani, “Visual slam for automated driving: Exploring the applications of deep learning”, u *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[35] A. Geiger, P. Lenz, C. Stiller, i R. Urtasun, “The kitti vision benchmark suite”, *URL http://www. cvlibs. net/datasets/kitti*, sv. 2, br. 5, str. 1–13, 2015.

[36] Nepoznato, https://hr.geekbuying.com/item/DJI-Phantom-3-Professional-Version-with-4480mA-Battery-4K-Camera-GPS-GLONASS-White-Gold-343969.html, [mrežno; stranica posjećena: lipanj 2024.].

[37] G. Zhou, L. Fang, K. Tang, H. Zhang, K. Wang, i K. Yang, “Guidance: A visual sensing platform for robotic applications”, 06 2015., str. 9–14. https://doi.org/10.1109/CVPRW.2015.7301360

[38] R. Opromolla, G. Fasano, i D. Accardo, “A vision-based approach to uav detection and tracking in cooperative applications”, *Sensors*, sv. 18, br. 10, 2018.

https://doi.org/10.3390/s18103391

[39] C. Sifferman, “A review of scene representations for robot manipulators”, 2022.

[40] T. Kroger i F. Wahl, “Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events”, *Robotics, IEEE Transactions on*, sv. 26, str. 94 – 111, 03 2010. https://doi.org/10.1109/TRO.2009.2035744

[41] M. Gao, N. Ruan, J. Shi, i W. Zhou, “Deep neural network for 3d shape classification based on mesh feature”, *Sensors*, sv. 22, br. 18, 2022. https://doi.org/10.3390/s22187040

[42] D. Whitney, E. Rosen, E. Phillips, G. Konidaris, i S. Tellex, *Comparing Robot Grasping Teleoperation Across Desktop and Virtual Reality with ROS Reality*, 11 2019., str. 335–350. https://doi.org/10.1007/978-3-030-28619-4_28

[43] B. Omarali, B. Denoun, K. Althoefer, L. Jamone, M. Valle, i I. Farkhatdinov, “Virtual reality based telerobotics framework with depth cameras”, 08 2020., str. 1217–1222. https://doi.org/10.1109/RO-MAN47096.2020.9223445

[44] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, i R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis”, 2020.

[45] K. Gao, Y. Gao, H. He, D. Lu, L. Xu, i J. Li, “Nerf: Neural radiance field in 3d vision, a comprehensive review”, 2023.

[46] J. T. Kajiya i B. P. Von Herzen, “Ray tracing volume densities”, *SIGGRAPH Comput. Graph.*, sv. 18, br. 3, str. 165–174, jan 1984. https://doi.org/10.1145/964965.808594

[47] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. A. Hamprecht, Y. Bengio, i A. Courville, “On the spectral bias of neural networks”, 2019.

[48] G. Wang, L. Pan, S. Peng, S. Liu, C. Xu, Y. Miao, W. Zhan, M. Tomizuka, M. Pollefeys, i H. Wang, “Nerf in robotics: A survey”, 2024.

[49] E. Sucar, S. Liu, J. Ortiz, i A. J. Davison, “imap: Implicit mapping and positioning in real-time”, 2021.

[50] E. Kruzhkov, A. Savinykh, P. Karpyshev, M. Kurenkov, E. Yudin, A. Potapov, i D. Tsetserukou, "Meslam: Memory efficient slam based on neural fields", 2022.

[51] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, i D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections", 2021.

[52] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, i H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis", 2022.

[53] W. Yuan, Z. Lv, T. Schmidt, i S. Lovegrove, "Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering", *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, str. 13 139–13 147, 2020. [Mrežno]. Adresa: https://api.semanticscholar.org/CorpusID:230523891

[54] T. Li, M. Slavcheva, M. Zollhoefer, S. Green, C. Lassner, C. Kim, T. Schmidt, S. Lovegrove, M. Goesele, R. Newcombe, i Z. Lv, "Neural 3d video synthesis from multi-view video", 2022.

[55] A. Pumarola, E. Corona, G. Pons-Moll, i F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes", u *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021., str. 10 318–10 327.

[56] Y.-L. Liu, C. Gao, A. Meuleman, H.-Y. Tseng, A. Saraf, C. Kim, Y.-Y. Chuang, J. Kopf, i J.-B. Huang, "Robust dynamic radiance fields", 01 2023. https://doi.org/10.48550/arXiv.2301.02239

[57] Z. Li, S. Niklaus, N. Snavely, i O. Wang, "Neural scene flow fields for space-time view synthesis of dynamic scenes", 2021.

[58] C. Gao, A. Saraf, J. Kopf, i J.-B. Huang, "Dynamic view synthesis from dynamic monocular video", 2021.

[59] S. Liang, Y. Liu, S. Wu, Y.-W. Tai, i C.-K. Tang, "Onerf: Unsupervised 3d object segmentation from multiple views", 2022.

[60] S. Zhi, T. Laidlow, S. Leutenegger, i A. J. Davison, “In-place scene labelling and understanding with implicit scene representation”, 2021.

[61] S. Zhi, E. Sucar, A. Mouton, I. Haughton, T. Laidlow, i A. J. Davison, “Ilabel: Interactive neural scene labelling”, 2021.

[62] X. Fu, S. Zhang, T. Chen, Y. Lu, L. Zhu, X. Zhou, A. Geiger, i Y. Liao, “Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation”, 03 2022.

[63] Sumit Singh, https://www.labellerr.com/blog/semantic-vs-instance-vs-panoptic-which-image-segmentation-technique-to-choose/, [mrežno; stranica posjećena: lipanj 2024.].

[64] W. Jang i L. Agapito, “Codenerf: Disentangled neural radiance fields for object categories”, 09 2021.

[65] T. Xu i T. Harada, “Deforming radiance fields with cages”, 2022.

[66] J. Ost, F. Mannan, N. Thuerey, J. Knodt, i F. Heide, “Neural scene graphs for dynamic scenes”, 2021.

[67] Y. Li, Z.-H. Lin, D. Forsyth, J.-B. Huang, i S. Wang, “Climatenerf: Physically-based neural rendering for extreme climate synthesis”, 11 2022. https://doi.org/10.48550/arXiv.2211.13226

[68] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, i T.-Y. Lin, “Inerf: Inverting neural radiance fields for pose estimation”, 2021.

[69] S. Chen, Z. Wang, i V. Prisacariu, “Direct-posenet: Absolute pose regression with photometric consistency”, 12 2021., str. 1175–1185. https://doi.org/10.1109/3DV53792.2021.00125

[70] Z. Wang, S. Wu, W. Xie, M. Chen, i V. A. Prisacariu, “Nerf–: Neural radiance fields without known camera parameters”, 2022.

[71] Y. Shi, D. Rong, B. Ni, C. Chen, i W. Zhang, “Garf:geometry-aware generalized neural radiance field”, 2022.

[72] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, i M. Schwager, “Vision-only robot navigation in a neural radiance world”, 2022.

[73] T. Chen, P. Culbertson, i M. Schwager, “Catnips: Collision avoidance through neural implicit probabilistic scenes”, 2023.

[74] P. Marza, L. Matignon, O. Simonin, i C. Wolf, “Multi-object navigation with dynamically learned neural implicit representations”, 2023.

[75] F. Li, S. R. Vutukur, H. Yu, I. Shugurov, B. Busam, S. Yang, i S. Ilic, “Nerf-pose: A first-reconstruct-then-regress approach for weakly-supervised 6d object pose estimation”, u *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, October 2023., str. 2123–2133.

[76] Z. Hu, R. Tan, Y. Zhou, J. Woon, i C. Lv, “Template-based category-agnostic instance detection for robotic manipulation”, 2022. https://doi.org/10.1109/LRA.2022.3219021

[77] J. Ichnowski, Y. Avigal, J. Kerr, i K. Goldberg, “Dex-nerf: Using a neural radiance field to grasp transparent objects”, 2021.

[78] D. Driess, I. Schubert, P. Florence, Y. Li, i M. Toussaint, “Reinforcement learning with neural radiance fields”, 2022.

[79] B. Kerbl, G. Kopanas, T. Leimkühler, i G. Drettakis, “3d gaussian splatting for real-time radiance field rendering”, 2023.

[80] A. Yu, S. Fridovich-Keil, M. Tancik, Q. Chen, B. Recht, i A. Kanazawa, “Plenoxels: Radiance fields without neural networks”, 2021.

[81] A. Dalal, D. Hagen, K. G. Robbersmyr, i K. M. Knausgård, “Gaussian splatting: 3d reconstruction and novel view synthesis, a review”, 2024.

[82] Kate Yurkova, https://towardsdatascience.com/a-comprehensive-overview-of-gaussian-splatting-e7d570081362, [mrežno; stranica posjećena: lipanj 2024.].

[83] C. Sun, M. Sun, i H.-T. Chen, “Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction”, 2022.

[84] A. H. Baker, A. Pinard, i D. M. Hammerling, “Dssim: a structural similarity index for floating-point data”, 2023.

[85] C. Lassner i M. Zollhöfer, “Pulsar: Efficient sphere-based neural rendering”, 2020.

[86] D. Merrill i A. S. Grimshaw, “Revisiting sorting for gpgpu stream architectures”, *2010 19th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, str. 545–546, 2010. [Mrežno]. Adresa: https://api.semanticscholar.org/CorpusID:14902096

[87] J. Li, J. Zhang, X. Bai, J. Zheng, X. Ning, J. Zhou, i L. Gu, “Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization”, 2024.

[88] D. Charatan, S. Li, A. Tagliasacchi, i V. Sitzmann, “pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction”, 2024.

[89] Z. Yan, W. F. Low, Y. Chen, i G. H. Lee, “Multi-scale 3d gaussian splatting for anti-aliased rendering”, 2024.

[90] Z. Yu, A. Chen, B. Huang, T. Sattler, i A. Geiger, “Mip-splatting: Alias-free 3d gaussian splatting”, 2023.

[91] Y. Li, C. Lyu, Y. Di, G. Zhai, G. H. Lee, i F. Tombari, “Geogaussian: Geometry-aware gaussian splatting for scene rendering”, 2024.

[92] J. Zhang, F. Zhan, M. Xu, S. Lu, i E. Xing, “Fregs: 3d gaussian splatting with progressive frequency regularization”, 2024.

[93] S. Szymanowicz, C. Rupprecht, i A. Vedaldi, “Splatter image: Ultra-fast single-view 3d reconstruction”, 2024.

[94] Z. Fan, K. Wang, K. Wen, Z. Zhu, D. Xu, i Z. Wang, “Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps”, 2024.

[95] J. C. Lee, D. Rho, X. Sun, J. H. Ko, i E. Park, “Compact 3d gaussian representation for radiance field”, 2024.

[96] J.-C. Shi, M. Wang, H.-B. Duan, i S.-H. Guan, “Language embedded 3d gaussians for open-vocabulary scene understanding”, 2023.

[97] M. Qin, W. Li, J. Zhou, H. Wang, i H. Pfister, “Langsplat: 3d language gaussian splatting”, 2024.

[98] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, i X. Jin, “Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction”, 2023.

[99] G. Chen i W. Wang, “A survey on 3d gaussian splatting”, 2024.

[100] Y. Jiang, J. Tu, Y. Liu, X. Gao, X. Long, W. Wang, i Y. Ma, “Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces”, 2023.

[101] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, i B. Dai, “Scaffold-gs: Structured 3d gaussians for view-adaptive rendering”, 2023.

[102] J. Chung, J. Oh, i K. M. Lee, “Depth-regularized optimization for 3d gaussian splatting in few-shot images”, 2024.

[103] Y. Li, X. Fu, S. Zhao, R. Jin, i S. K. Zhou, “Sparse-view ct reconstruction with 3d gaussian volumetric representation”, 2023.

[104] Y. Cai, Y. Liang, J. Wang, A. Wang, Y. Zhang, X. Yang, Z. Zhou, i A. Yuille, “Radiative gaussian splatting for efficient x-ray novel view synthesis”, 2024.

[105] K. Wang, C. Yang, Y. Wang, S. Li, Y. Wang, Q. Dou, X. Yang, i W. Shen, “Endogslam: Real-time dense reconstruction and tracking in endoscopic surgeries using gaussian splatting”, 2024.

[106] Y. Liu, C. Li, C. Yang, i Y. Yuan, “Endogaussian: Real-time gaussian splatting for dynamic endoscopic scene reconstruction”, 2024.

[107] J. Tang, J. Ren, H. Zhou, Z. Liu, i G. Zeng, “Dreamgaussian: Generative gaussian splatting for efficient 3d content creation”, 2024.

[108] S. Qian, T. Kirschstein, L. Schoneveld, D. Davoli, S. Giebenhain, i M. Nießner, “Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians”, 2024.

[109] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, i J. Luiten, “Splatam: Splat, track & map 3d gaussians for dense rgb-d slam”, 2024.

[110] H. Matsuki, R. Murai, P. H. J. Kelly, i A. J. Davison, “Gaussian splatting slam”, 2024.

[111] C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, i X. Li, “Gs-slam: Dense visual slam with 3d gaussian splatting”, 2024.

[112] D. P. Kingma i J. Ba, “Adam: A method for stochastic optimization”, 2017.

[113] J. Sturm, N. Engelhard, F. Endres, W. Burgard, i D. Cremers, “A benchmark for the evaluation of rgb-d slam systems”, u *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[114] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, i R. Newcombe, “The replica dataset: A digital replica of indoor spaces”, 2019. [Mrežno]. Adresa: https://arxiv.org/abs/1906.05797

Sažetak

Gaussove trodimenzionalne reprezentacije za istovremenu lokalizaciju i mapiranje vizualnim senzorima

Ante Sladić

Kroz rad daje se uvid u SLAM sustave, reprezentacije scene uključujući NeRF i Gaussove trodimenzionalne reprezentacije, te naposljetku sustav MonoGS kao primjer SLAM sustava sa integriranim Gaussovim trodimenzionalnim reprezentacijama, popraćen sa rezultatima dobivenim u eksperimentima. Cilj je prikazati prednosti trodimenzionalnih Gaussovih reprezentacija kroz teorijsku pozadinu kako bi shvatili zašto su superioriniji naspram prethodnih reprezentacija i kroz eksperimente kako bi to potvrdili tu hipotezu.

Ključne riječi: neuronska radijalna polja; Gaussove trodimenzionalne reprezentacije; SLAM; MonoGS; računalni vid; monokularna kamera

Abstract

Gaussian three-dimensional representations for simultaneous localization and mapping with visual sensors

Ante Sladić

The paper gives insight into SLAM systems, scene representations including NeRF and Gaussian three-dimensional representations, and finally the MonoGS system as an example of a SLAM system with integrated Gaussian three-dimensional representations, accompanied by results obtained in experiments. The goal is to show the advantages of three-dimensional Gaussian representations through theoretical background to understand why they are superior to previous representations and through experiments to confirm this hypothesis.

Keywords: neural radiance fields; 3D Gaussian scene representations; SLAM; MonoGS; computer vision; monocular camera