Proširena stvarnost: simultano mapiranje i lokalizacija

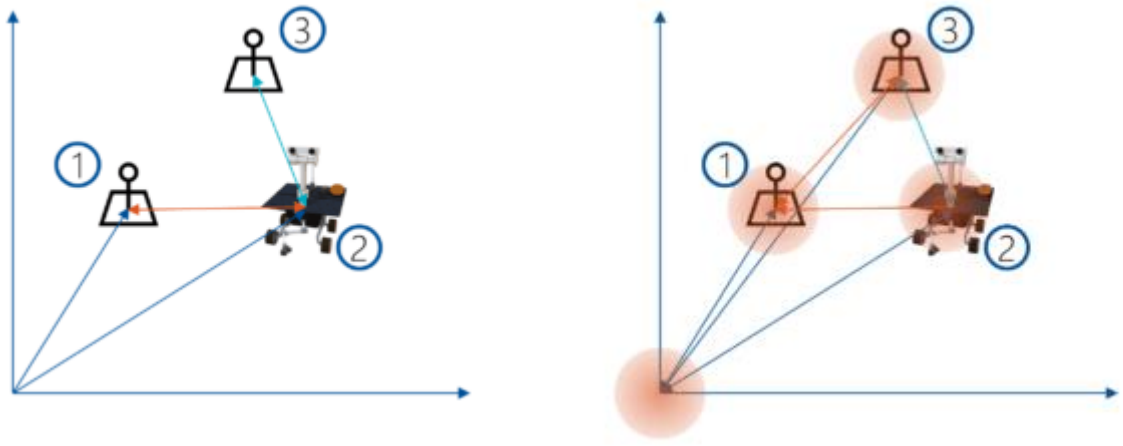
Bartol Rod

UVOD

Prilikom pokretanja aplikacije, u proširenoj stvarnosti potrebno je prepoznati okruženje. Sustav najviše informacija prima iz kamere, no za bolju preciznost koristi i brzinomjer, žiroskop, razne senzore itd. Od tako dobivenih podataka potrebno je stvoriti mapu prostora i locirati objekte. U ovom istraživanju predstavljaju se osnove simultanog mapiranja i lokalizacije. Opisani su izazovi koji nas susreću prilikom stvaranja SLAM aplikacije, anatomija takve aplikacije, postojeće rješenje i budućnost SLAM aplikaciji.

IZAZOVI

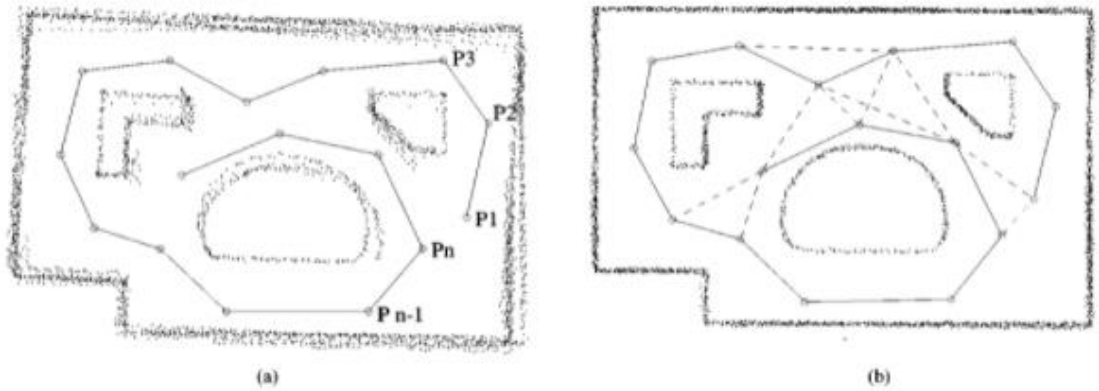
Prvi problem SLAM algoritma je lokalizacija. Algoritam mora uspješno locirati položaj promatrača i ostalih objekta u prostoru. Prilikom lociranja točki u proširenoj stvarnosti njihov položaj je neizvjestan. To jest, u nesavršenom, stvarnom snimljenom prostoru položaj točaka u svakom trenutku nije nam sa sigurnošću poznat. Svaka točka nalazi se u svojem „oblaku“ i može se sa određenom vjerojatnošću naći na određenoj udaljenosti od stvarnog položaja (slika 1.) [1]. Za rješenje ovog problema najčešće se koriste algoritmi poput *Maximum A Posteriori, MAP* algoritma ili *Bundle Adjustment*, BA algoritma. [1]



Slika 1. Prikaz „oblaka“ vjerojatnosti pozicija [1]

Drugi problem s kojim se susrećemo je mapiranje ili stvaranje mape svijeta. Svaki novi izmjereni položaj potrebno je što točnije poravnati sa znanim informacijama o do sad poznatom svijetu (slika 2.a). To je ostvarivo poravnanjem prijašnje izmjerenih položaja (slika 2.b). Pitanje je koliko informacija o prijašnjim mjerenjima je dovoljno pamtiti jer s vremenom sustav

počinje gubiti memoriju i brzinu izvršavanja. Za ovaj problem često se koriste algoritmi koji koriste princip *survival of the fittest* kako bi se kasnije uklonila povijest koja nam nije bila korisna. [1]

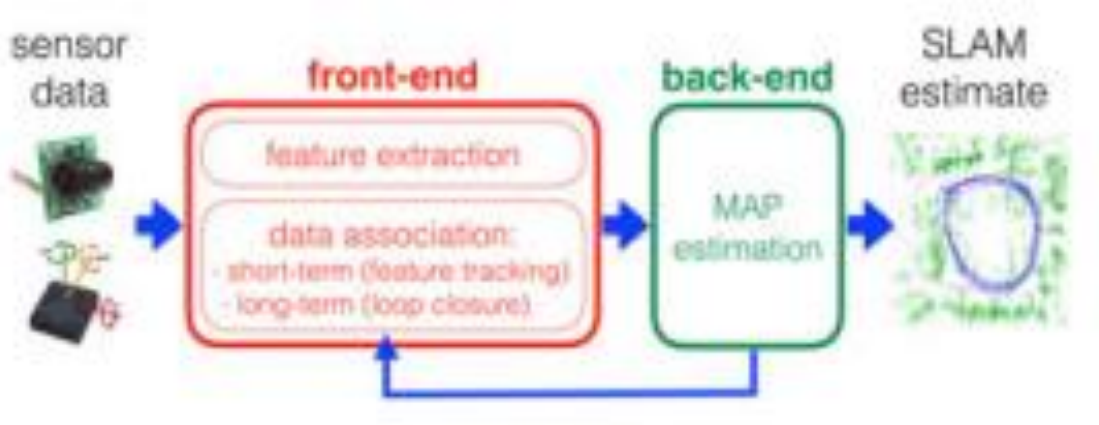


Slika 2. Nakupljanje pogreške [2]

Matematički opis ovih problema slijedi: za skup pokreta u_t i opažanja kamere o_t u stvarnom vremenu *t* potrebno je odrediti poziciju agenta x_t i mapu svijeta m_t. [7] Navedene veličine su probabilističke stoga je cilj izračunati $P(m_{t+1}, x_{t+1} | \, o_{1:t+1}, u_{1:t})$. [7]

ANATOMIJA SLAM APLIKACIJE

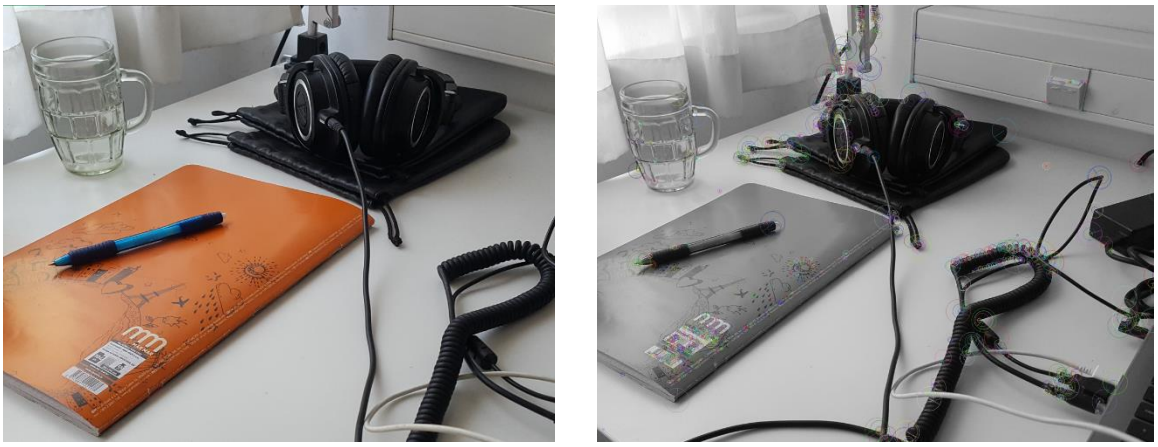
Navedene probleme moramo riješiti nekim redoslijedom, stoga se u ovom odlomku predstavlja anatomija VR: SLAM aplikacije u najosnovnijem obliku.



Slika 3. Anatomija SLAM aplikacije [3]

Kao i u svakoj aplikaciji, problematika se može podijeliti na *front-end* i *back-end*. Dakle, na *front-endu*, u podacima sa naših kamera, se izdvajaju značajke iz prostora. Tu nam mogu pomoći razni algoritmi, na primjer *Binary Robust Invariant Scalable Keypoints,* BRISK.

[4] Algoritam u osnovici radi na način da gleda okruženje od šesnaest piksela nekog promatranog piksela i uspoređuje njihove nijanse sive boje. Ako je barem devet slijednih piksela tamnije od središnjeg, područje se može označiti kao kut. [4]



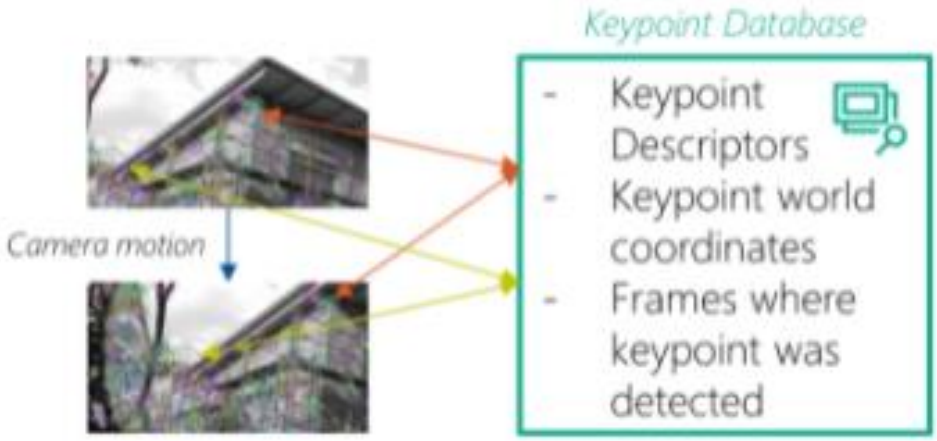
Slika 4. BRISK, primjer

Na *back-endu* iz tih podataka računaju se veze kutova, lokacija kamere te stvara se i rekonstruira cijela geometrija prostora proširene stvarnosti. Na taj način na izlazu dobivamo ključne značajke prostora, veze između njih i sve pozicije u prostoru.

U sljedećem odlomku predstavit će se konkretna implementacija navedenih mehanizama, algoritam ORB-SLAM.

ORB-SLAM ALGORITAM

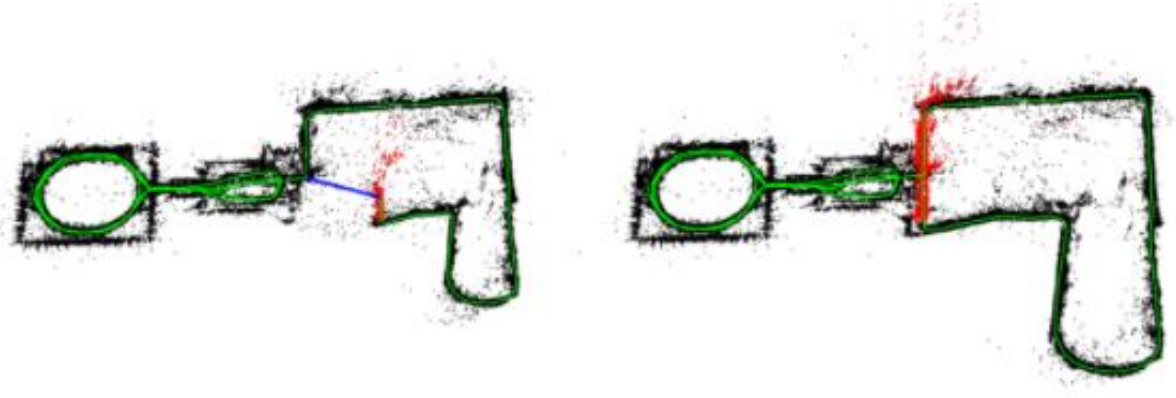
Jedna poznata implementacija SLAM problema je algoritam ORB-SLAM, *Mur-Atal, Montiel i Tardos*. [5] ORB-SLAM vizualan je algoritam što znači da informacije o prostoru uzima s kamere. [1] Ovaj algoritam za praćenje ključnih značajki koristi ORB algoritam. ORB je efektivnija inačica algoritama poput SIFT i SURF. [6] Koristi iste metode za pronalazak značajki kao i algoritam BRISK. Nakon što iz trenutnog *framea* očita sve značajke, sprema ih u mapu za daljnju usporedbu. [1]



Slika 5. Mapa značajki [1]

Iz dobivenih mjerenja novog *framea*, referencirajući se na prijašnji, algoritam pokušava podudariti ključne značajke i novu lokaciju kamere. [1] Nakon inicijalnog koraka tako dobivena pozicija kamere pokušava se unaprijediti na način da algoritam trenutni *frame* namješta u svoju mapu svijeta, ako je algoritam dovoljno siguran da je pozicija kamere dovoljno unaprijeđena pomoću triangulacije, translacije, rotacije osvježava svoju mapu svijeta. Ovo je kratak opis funkcije *LocalMapping::CreateNewMapPoints()* iz izvornog koda. [5] [1]

Zanimljiva funkcionalnost algoritma je i detekcija petlji, slika 5. Kako se kroz korištenje aplikacije izmjenjuju *frameovi* algoritam uspoređuje i koliko je slična pozicija ključnih značajki različitih lokacija te ako je sličnost prošla određenu granicu lokacije se spajaju i u mapi zatvaraju petlju.



Slika 5. Zatvaranje petlje u mapi [1]

Također, razvijen je ORB-SLAM2. To je unaprijeđena inačica ORB-SLAM algoritma te ima mogućnost korištenja monokularne, stereo i RGB-D kamere.

ZAKLJUČAK

U ovom istraživanju predstavljene su osnove simultanog mapiranja i lokalizacije u prostoru proširene stvarnosti. Predstavljeni su izazovi s kojima se susrećemo, neodređenost položaja i nakupljanje grešaka i neka rješenja tih problema te je ukratko opisan matematički model našeg problema. Nadalje, predstavljena je osnovna arhitektura aplikacije u kojoj je primijenjen SLAM, konkretna implementacija ORB-SLAM i neke funkcionalnosti tog algoritma.

Simultano mapiranje i lokalizacija ključan su dio pri ostvarivanju proširene stvarnosti, tehnologije budućnosti, trenutna tehnologija nije savršena i ima mnogo prostora za unaprjeđenje te ja stoga predmet proučavanja velikih tehnoloških kompanija poput Googlea.

LITERATURA

[1] *Basics of AR: SLAM – Simultaneous Localization and Mapping*
https://www.andreasjakl.com/basics-of-ar-slam-simultaneous-localization-and-mapping/

[2] *Lu, F., & Milios, E. (1997). Globally consistent range scan alignment for environment mapping. Autonomous robots, 4(4), 333-349*
https://link.springer.com/article/10.1023/A:1008854305733

[3] *Past, Present, and Future of Simultaneous Localization and Mapping: Towards the Robust-Perception Age (2016) by Cadena et. al.*
https://ieeexplore.ieee.org/abstract/document/7747236

[4] *Basics of AR: Amchors, Keypoints and Feature Detection*

https://www.andreasjakl.com/basics-of-ar-anchors-keypoints-feature-detection/

[5] *ORB-SLAM by Mur-Atal, Montiel and Tardós*

https://github.com/raulmur/ORB_SLAM/tree/ce199650a25653808f96b83557333bce3461d29f

[6] *ORB: an efficient alternative to SIFT or SURF by Ethan Rublee Vincent Rabaud Kurt Konolige Gary Bradski*

https://ieeexplore.ieee.org/abstract/document/6126544

[7] *Simultaneous localization and MapPoint, wikipedia*

https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping#Kinematics_modeling