

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Evolucijska robotika

Vizualna lokalizacija i estimacija pokreta

Rijeka, svibanj 2020.

Katarina Tolja

0011158843

Sadržaj

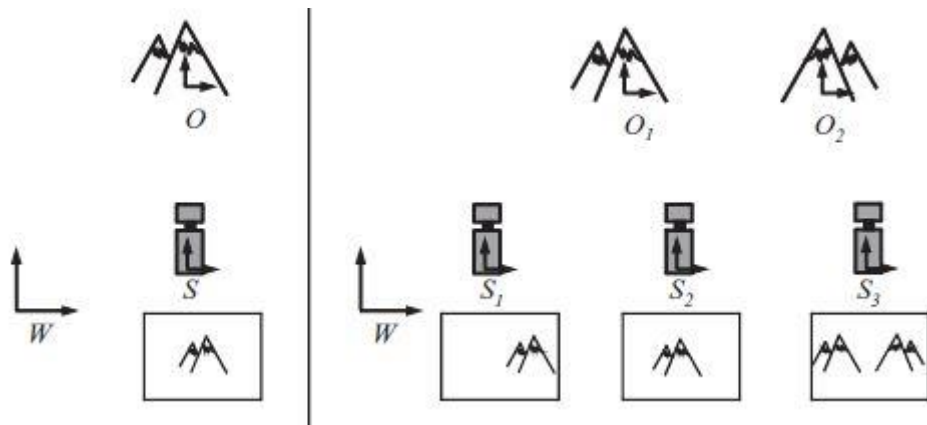
1. UVOD	1
2. OSNOVNI KONCEPTI	2
3. VIZUALNA LOKALIZACIJA.....	5
3.1. Formatiranje slika.....	5
3.2. Lokalizacija objekata.....	6
3.3. Osnovni pristup i problemi.....	7
3.4. Lokalizacija robota	8
3.5. Vizualna estimacija pokreta	9
3.6. Temeljni algoritmi	10
4. USKLAĐIVANJE SIGNALA LOKALIZACIJE I ESTIMACIJE POKRETA.....	11
4.1. Objektivne funkcije na temelju signala	11
4.2. Poravnavanje video signala u ravninu slike	12
4.3. Poravnavanje geometrije površine u raspon slika	13
5. USKLAĐIVANJE ZNAČAJKI LOKALIZACIJE I ESTIMACIJE POKRETA.....	15
5.1. Razvrstavanje i značajke	15
5.2. Objektivna funkcija	15
5.3. Tipične značajke i objektivne funkcije.....	16
5.4. RANSAC.....	17
5.5. Najbliže točke u rasponu podataka.....	19
6. OPTIMALNA POZICIJA	22
6.1. Određivanje pozicije.....	22
6.2. Ugladivanje pozicije.....	23
6.3. Praćenje pozicije.....	26
7. ZAKLJUČAK	31
8. LITERATURA.....	32
9. POPIS SLIKA	33

1. UVOD

Vizualna lokalizacija je u osnovi problem estimiranja kamere unutar prizora (scene) te ključna komponenta u vizualnim aplikacijama kao što su samoupravljivi automobili. U nastavku će se objasniti osnovni kocepti te problemi i moguća rješenja prilikom vizualne lokalizacije, usklađivanje signala i značajki lokalizacije i estimacije pokreta i optimalne pozicije. Svi će aspekti proučavanja navedenih tema potkrijepiti matematičkim opisom i primjerima. Primjeri će biti dani i sa škotskog instituta tehnologije, gdje se u iznimnoj mjeri proučavaju metode mapiranja, lokaliziranje te samog kretanja robota. Jedan od projekata odnosi se na kretanje drona u zatvorenom prostoru. Rad se prezentirao na Tehničkom fakultetu 2019. godine.

2. OSNOVNI KONCEPTI

Svi načini na koje senzor percepcije može eksplotirati karte i modele objekta za izradu spomenutih te za estimaciju pokreta senzora prikazani su na slici 2.1. Svi senzori percepcije su prezentirani kao kamera, a u ostatku rada bit će referirani kao slike.



Slika 2.1. Opći okvir [1]

Stvarno okruženje će biti referirano kao prizor kako bi se razlikovalo od njegovog prikaza u karti ili objektu modela. Također, pretpostavit će se da je pozicija senzora na robotu poznata tako da je lokalizacija senzora u bilo kojem okviru jednaka lokalizaciji robota.

Kanonski problemi:

➤ Mapiranje

Problem mapiranja je određivanje pozicije objekta u prizoru. Senzori ga ne mogu direktno mjeriti, ali mogu izmjeriti poziciju objekta u okviru senzora, ako je poznata pozicija robota. Princip je opisan s sljedećim izrazom:

$$\underline{\rho}_O^W = \underline{\rho}_S^W * \underline{\rho}_O^S, \quad (2.1)$$

gdje je:

$\underline{\rho}_O^W$ – pozicija objekta u prizoru,

$\underline{\rho}_S^W$ – pozicija senzora u prizoru,

$\underline{\rho}_O^S$ – pozicija objekta u okviru senzora.

➤ Lokalizacija

Problem pri lokalizaciji robota je određivanje $\underline{\rho}_S^W$. Ukoliko pozicija robota nije poznata, ali je $\underline{\rho}_O^W$ poznat, onda možemo dobiti kartu. Robota će biti moguće lokalizirati s sljedećim izrazom:

$$\underline{\rho}_S^W = \underline{\rho}_O^W * \underline{\rho}_S^W. \quad (2.2)$$

➤ Vizualna estimacija pokreta

U slučaju da nije poznata ni pozicija objekta $\underline{\rho}_O^W$ ni pozicija robota $\underline{\rho}_S^W$ onda se rubovi u pozicijskoj mreži ne povezuju s okvirom i ništa se ne može lokalizirati u odnosu na senzor. Međutim, još uvijek je moguće izmjeriti gibanje senzora u odnosu na predmete koji se promatraju. Ako se robot promatra u dvije slike odvojene gibanjem robota onda je to moguće opisati sljedećim izrazom:

$$\underline{\rho}_{S2}^{S1} = \underline{\rho}_{O1}^{S1} * \underline{\rho}_{S2}^{O1}, \quad (2.3)$$

gdje je:

$\underline{\rho}_{S2}^{S1}$ – pozicija drugog senzora u odnosu na okvir prvog,

$\underline{\rho}_{O1}^{S1}$ – pozicija jednog objekta u odnosu na okvir prvog senzora,

$\underline{\rho}_{S2}^{O1}$ – pozicija drugog senzora u odnosu na okvir prvog objekta.

➤ Simultana lokalizacija i mapiranje (SLAM)

Kada ni robot ni niti jedan drugi predmet nije poznat s obzirom na okvir, tada je moguće proizvoljno napraviti kartu. Na primjer, ako je drugi objekt prikazan u drugom senzoru pozicije, prvi objekt se može razmatrati kao original te se drugi može lokalizirati u odnosu na njega:

$$\underline{\rho}_{O2}^{O1} = \underline{\rho}_{S1}^{O1} * \underline{\rho}_{S2}^{S1} * \underline{\rho}_{O2}^{S2}, \quad (2.4)$$

gdje je:

$\underline{\rho}_{O2}^{O1}$ – pozicija drugog objekta u odnosu na prvi objekt,

$\underline{\rho}_{S1}^{O1}$ – pozicija prvog senzora u odnosu na prvi objekt,

$\underline{\rho}_{S2}^{S1}$ – pozicija drugog senzora u odnosu na prvi senzor,

$\underline{\rho}_{O2}^{S2}$ – pozicija drugog objekta u odnosu na drugi senzor.

Postupak će se za svaki naredni objekt ponavljati. Činjenica da su te dvije pozicije sastavljene kako bi lokalizirale drugi objekt sugerira da je to postupak koji nakuplja pogreške.

➤ Konzistentno mapiranje

Glavni problem konzistencije je što ako se robot vrati da pogleda neki predmet (npr. O_2) drugi put nakon što se značajna greška nakupila. Moguće je da neće prepoznati da je taj objekt (npr. O_{100}) isti kao predmet O_2 te će sigurno zaključiti sljedeće:

$$\underline{\rho}_{O_2}^{O_1} \neq \underline{\rho}_{100}^{O_1}. \quad (2.5)$$

Kako bi se ispravila navedena greška objekti se moraju prepoznati kao isti.

3. VIZUALNA LOKALIZACIJA

Proces vizualne lokalizacije temelji se na usporedbi onoga što robot vidi s onim što predviđa na temelju neke vrste mape ili modela. S obzirom na takvu definiciju, čak i GPS koristi proces vizualne lokalizacije gdje je senzor predstavljen višekanalskim radarom, a karta pozicijom satelita u orbiti. Vizualna lokalizacija nadalje će se detaljnije objasniti kroz sljedeća potpoglavlja: formatiranje slike, lokalizaciju objekata, osnovni pristupi i problemi, lokalizacija robota, vizualna estimacija pokreta te temeljni algoritmi [1] [3].

3.1. Formatiranje slika

Pretpostavlja se da je model objekta dostupan te se mogu prepoznati karakteristične točke u modelu. Točke modela u početku su poznate u koordinatama modela te ih je potrebno pretvoriti u koordinate senzora. U slučaju da je pozicija objekta u odnosu na senzor poznata, transformacija točaka iz okvira modela u okvir senzora glasi:

$$\underline{X}^S = T^S_0 \left(\underline{\rho}_O^S \right) \underline{X}^m, \quad (3.1)$$

gdje je:

T^S_0 – koordinate senzora,

$\underline{\rho}_O^S$ – pozicija objekta u odnosu na senzor,

\underline{X}^m – točke modela.

Također se može koristiti matrica projekcije kamere kako bi se vidjelo gdje točka pada na ravninu slike:

$$\underline{x}_i = P \underline{X}^S, \quad (3.2)$$

gdje je:

P – matrica projekcije kamere,

\underline{X}^S – točke.

Matrica projekcije kamere glasi:

$$\begin{bmatrix} \underline{x}_i \\ \underline{y}_i \\ \underline{z}_i \\ w_i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1/f & 0 & 1 \end{bmatrix} \begin{bmatrix} \underline{x}_S \\ \underline{y}_S \\ \underline{z}_S \\ 1 \end{bmatrix}. \quad (3.3)$$

3.2. Lokalizacija objekata

Lokalizacija objekta se razmatra s obzirom na senzor. Samo se relativna pozicija senzora i predmeta može mjeriti, pa je ekvivalentno tome senzor lokaliziran u odnosu na objekt. Kada slika objekta sadrži dovoljno podataka moguće je izračunati relativnu poziciju. Osnovni prikaz problema započinje s poznatim odnosom mjera i s još nekoliko argumenata:

$$\underline{z}(\underline{x}) = \underline{h}(\underline{x}, \underline{\rho}_o^s, \underline{Z}), \quad (3.4)$$

gdje je:

$\underline{z}(\underline{x})$ – slika signala s definiranom strukturom,

\underline{h} – proces formacije slike,

$\underline{\rho}_o^s$ – pozicija objekta u odnosu na senzor,

\underline{Z} – objekt modela.

Osnovne smjernice predviđanja slike su:

- objekt modela se može definirati kao signal $\underline{z}(\underline{x})$ preko skupa koordinata,
- koordinate slike i prizora su povezane s malim dimenzijskim transformacijama,
- jednom kad je transformacija poznata, cijela je slika predvidljiva iz modela.

Ovaj kratki opis opisuje računalnu grafiku.

Na primjer, \underline{x} može predstavljati koordinate reda i stupaca u slici u boji, a $\underline{z}(\underline{x})$ mogu biti vrijednosti piksela u prostoru crvene, zelene i plave boje. Ključna točka je da se objektni model može prikazati kao signal preko skupa koordinata. Ako \underline{x} predstavlja točke u 3D prostoru, onda $\underline{z}(\underline{x})$ može predstavljati boju za svaku točku na prizoru (barem svaku točku na vidljivoj površini). Sva lokalizacija na percepciji slike se temelji na činjenici da su slike i koordinate scene povezane transformacijom malih dimenzija. Kada je ta transformacija poznata, cjeloukupna slika postaje predvidljiva iz modela. Na primjer, može se razmotriti senzor kamere u boji i pretpostaviti da je transformacija iz koordinata prizora u koordinate slike matrica T koja ovisi isključivo i relativnoj poziciji $\underline{\rho}_o^s$:

$$\underline{x} = T(\underline{\rho}_o^s) \underline{X}. \quad (3.5)$$

Uvrštavanjem u izraz trdobiva se sljedeća relacija:

$$\underline{z}(\underline{x}) = \underline{z} \left[T \left(\underline{\rho}_O^S \right) \underline{X} \right] = \underline{Z}(\underline{X}), \quad (3.6)$$

gdje je:

$\underline{Z}(\underline{x})$ – objekt modela,

$T \left(\underline{\rho}_O^S \right)$ – transformacija koordinata.

Drugim riječima, proces stvaranje slike prenosi signal iz točke u prizoru u odgovarajuću točku slike. Ovaj proces može biti predviđen ako se zna model slikanja koji transformira koordinate i sadržaj modela.

3.3. Osnovni pristup i problemi

Dva su osnovna pristupa vizualnoj lokalizaciji. Prvi je osnovan na usklađivanju signala. Traži se pozicija koja objašnjava predviđenu sliku danu modelu (ili karti):

$$\underline{z}_{pred}(\underline{x}) = \underline{h} \left(\underline{x}, \underline{\rho}_O^S, \underline{Z} \right), \quad (3.7)$$

gdje je:

$\underline{z}_{pred}(\underline{x})$ – predviđena slika,

$\underline{\rho}_O^S$ – pozicija objekta u odnosu na senzor.

Drugi pristup je osnovan na potražnji pozicije koja usklađuje koordinate (ili koordinacijkim funkcijama, koje će nadalje biti referirane kao značajke):

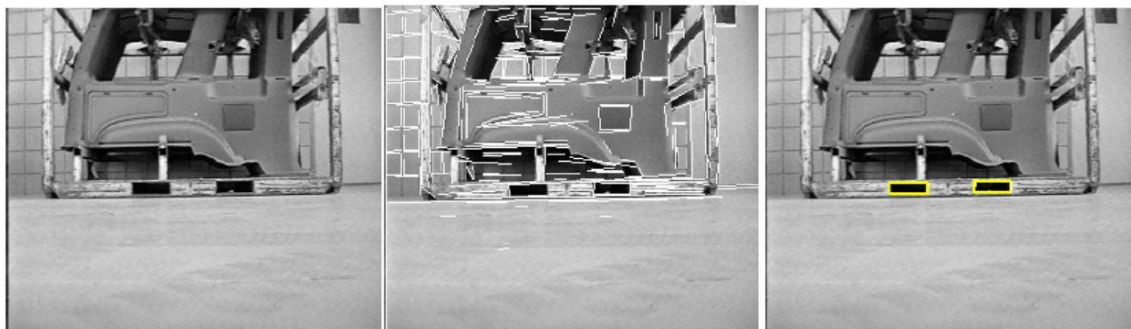
$$\underline{x}_{pred} = T \left(\underline{\rho}_O^S \right) \underline{X} \quad (3.8)$$

Uz dovoljan broj parova točaka prizora i slike $(\underline{x}_k, \underline{X}_k)$, može se izračunati pozicija ili čitava transformacija. Prilikom rješavanja ovog problem javlja se određeni broj problema. Jedan dio se odnosi na povezivanja podataka. Time se postavlja pitanje: ako bi dijelovi slike trebali biti usklađeni s dijelovima prizora, kako ćemo znati koje svojstvo slike uskladiti s kojim svojstvom prizora? Ovaj problem se naziva problem podudaranja. Drugi skup pitanja odnosi se na prirodu jednadžba koje je potrebno riješiti. Neka od takvih pitanja su :

- Postoji li jedinstvena pozicija koja opisuje sliku?
- Je li dostupna početna procjena pozicija?
- Kakva je kvaliteta podataka senzora?

- Koliko vremena i računanja je na raspolaganju?

Navedeni problemi su riješeni s Kalmanovim filtrima.



Slika 3.1. Detekcija rubova za vilica-rupa pronalazak (lijevo- početna slika s paletama, sredina- detektirani svi rubovi, desno- pronađene rupe za vilicu) [1]

Jedan takav primjer se odnosi na pronalazak paleta i prikazan je na slici 3.1. Prvo je potrebno reducirati sliku do rubova inteziteta. Potom uskladiti rubove s modelom (vilične rupe) te pronaći pozicije.

3.4. Lokalizacija robota

Lokalizacija objekata u odnosu na senzor je matematička indentična kao lokalizacija robota u odnosu na kartu. Predviđene mjere slike se mogu izraziti sljedećim izrazom:

$$\underline{z}_{pred}(\underline{x}) = \underline{h}(\underline{x}, \underline{\rho}_s^w, \underline{Z}), \quad (3.9)$$

gdje je:

$\underline{z}_{pred}(\underline{x})$ – slika,

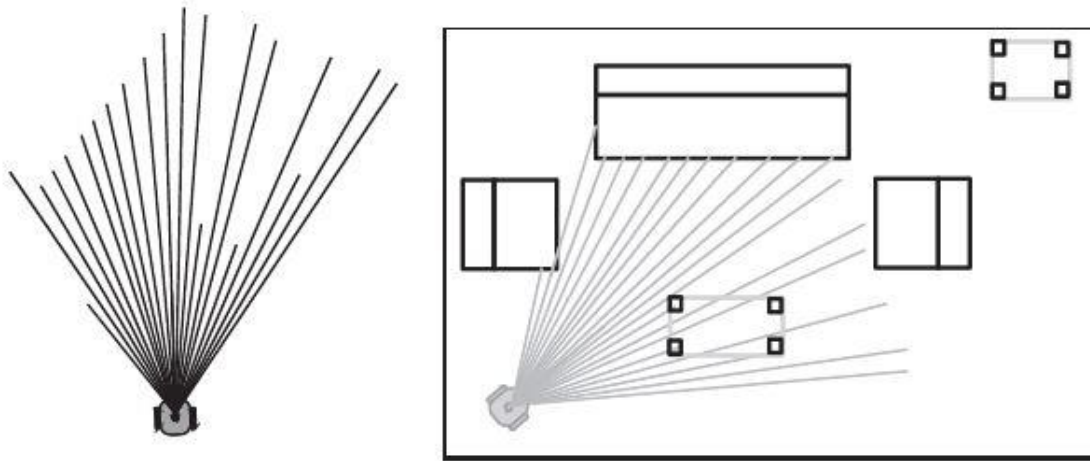
\underline{h} – formatiranje slike,

\underline{x} – koordinate slike,

$\underline{\rho}_s^w$ – pozicija senzora u odnosu na okružje,

\underline{Z} – karta.

Kao primjer se može uzeti lokaliziranje robota korištenjem lidar mreže. Lidar je detekcijski sustav koji radi na principu radar, ali koristi svjetlost lasera.



Slika 2.2. Lokalizacija bazirana na percepciji[1]

Slika 3.2. prikazuje slučaj u kojem se određuje pozicija robota na osnovi sadržaja lidar raspona slike. Pozicija vakuumskog robota za čišćenje je konzistentna s rasponom skeniranja.

3.5. Vizualna estimacija pokreta

U slučaju kada nema karte, još uvijek se može estimirati pokret tako da se upare prošle slike sa sadašnjim. Ako se objekt pojavljuje na obje slike znači da je mapiranje s jedne na drugu sliku moguće. Tada se to može opisati sljedećim izrazom:

$$\underline{x}_2 = T(\underline{\rho}_o^{s2}) \underline{X} = T(\underline{\rho}_o^{s2}) T^{-1}(\underline{\rho}_o^{s1}) \underline{x}_1 = T(\underline{\rho}_o^{s1}, \underline{\rho}_o^{s2}) \underline{x}_1, \quad (3.10)$$

gdje je:

$T(\underline{\rho}_o^{s1}, \underline{\rho}_o^{s2})$ – usklađivanje koordinata pozicija objekta s strane prvog i drugog senzora.

Međutim, ovo je jednostavniji problem od vizualne lokalizacije jer se signal ne treba predvidjeti s karte. Ponekad, se navedeni izraz može zapisati u odnosu na relativnu poziciju senzora:

$$\underline{x}_2 = T(\underline{\rho}_o^{s1}, \underline{\rho}_o^{s2}) \underline{x}_1 = T(\underline{\rho}_{s1}^{s2}) \underline{x}_1. \quad (3.11)$$

gdje je:

$T(\underline{\rho}_{S1}^{S2})$ – koordinate relativne pozicije prvog senzora s obzirom na drugi.

S obzirom na dovoljno značajnih podudaranja, može se utvrditi relativna pozicija. Navedena transformacija se može zapisati i kao transformacije slika na sliku:

$$\underline{x}_2 = T_S(\underline{\rho}_{S1}^{S2}) \underline{x}_1 = T_I(\underline{\rho}_{I1}^{I2}) \underline{x}_1. \quad (3.12)$$

3.6. Temeljni algoritmi

Do sada su upoznata tri osnovna algoritma koja se mogu koristiti za lokalizaciju i estimaciju pokreta:

- poravnavanje dva signala u dvije slike (jedan signal je mjereno, a drugi se može mjeriti ili predvidjeti),
- uspoređivanja značajki za stvaranje korespondencija (koristi se za oporavak nepoznate pozicije ili transformaciju),
- izračunavanje relativne pozicije u prizoru iz relativne pozicije na slici.

4. USKLAĐIVANJE SIGNALA LOKALIZACIJE I ESTIMACIJE POKRETA

U ovom odjeljku razmotrit će se aspekti usklađivanja raspona slike i videa. Poravnavanje signala je osnovno sredstvo vizualne lokalizacije. Transformacija koja je potrebna da bi se dva signala uskladila koristi se kao dokaz prostornog odnosa interesa. Usklađivanje signala obično se koristi kada je pozicijska nesigurnost dovoljno niska da je potraga za transformacijom poravnavanja izvediva. Također se koristi kada postoji relativno malo podataka u signalu [1].

4.1. Objektivne funkcije na temelju signala

Prvo je potrebno definirati signal predikcije:

$$\underline{z}_{pred}(\underline{x}, \underline{\rho}, \underline{Z}) = \underline{h}(\underline{x}, \underline{\rho}, \underline{Z}). \quad (4.1)$$

Srž problema je pronaći poziciju koja usklađuje promatrane i predviđene signale. Razmatra se rezidualna razlika između promatrane slike i one koja bi se mogla predvidjeti na temelju estimacije pozicije:

$$\underline{r}(\underline{x}, \underline{\rho}, \underline{Z}) = \underline{z}_{obs}(\underline{x}) - \underline{z}_{pred}(\underline{x}, \underline{\rho}, \underline{Z}), \quad (4.2)$$

gdje je:

$\underline{r}(\underline{x}, \underline{\rho}, \underline{Z})$ – rezidualna razlika,

$\underline{z}_{obs}(\underline{x})$ – promatrana slika.

Skalarni smjer može se formirati korištenjem zbroja kvadratnih rezidua signala za sve ili neke podskupine domene signala, a problem određivanja pozicije zatim se može formulirati kao nelinearni najmanji kvadrat koji se izračunava preko nekog prozora W u koordinatama slike:

$$\underline{\rho}^* = \underset{\underline{\rho}}{\operatorname{argmin}} \left[f(\underline{\rho}) = \frac{1}{2} \sum_{\underline{x} \in W} \underline{r}^T(\underline{x}, \underline{\rho}, \underline{Z}) \underline{r}(\underline{x}, \underline{\rho}, \underline{Z}) \right]. \quad (4.3)$$

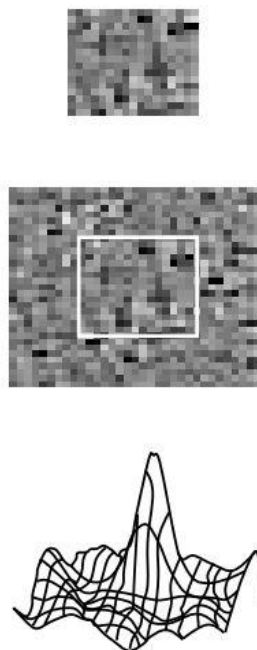
Potom je potrebno poredati sve elemente u rezidualnoj bazi na \underline{x} te se onda \underline{x} argumenti mogu otkloniti. Pod ovom transformacijom problem nelinearnih najmanjih kvadrata je:

$$\underline{\rho} = \underset{\underline{\rho}}{\operatorname{argmin}} \left[f(\underline{\rho}) = \frac{1}{2} \sum_{\underline{x} \in W} \underline{r}^T(\underline{\rho}, \underline{Z}) \underline{r}(\underline{\rho}, \underline{Z}) \right]. \quad (4.4)$$

Preporučljivo je normalizirati video slike prije računanja reziduumu.

4.2. Poravnavanje video signala u ravninu slike

Slika 4.1. prikazuje jednostavan slučaj u kojem se monokromatska slika pretražuje horizontalno i vertikalno u malom području kako bi se pronašao najbolji položaj promatrane slike. Izvodi se pretraga kroz prozor za pretraživanje. Potom se provjeravaju korespondencije u ograničenim regijama mogućih premještanja te razlikuju pročišćene pozicije i povezani podaci. Koristi se filter korelacije kako bi se smanjio okolni šum.



Slika 3.1. Korelacija[1]

Transformacija značajnih pozicija glasi:

$$\underline{y}(\underline{x}, \underline{p}) = \underline{T}(\underline{x}, \underline{p}) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix}. \quad (4.5)$$

Reziduumi piksela su definirani na sljedeći način:

$$(\underline{x}, \underline{p}, \underline{Z}) = \underline{z}_{obs}(\underline{x}) - \underline{z}_{pred}(\underline{x}, \underline{p}, \underline{Z}) = \underline{z}_{obs}(\underline{x}) - \underline{Z}[\underline{y}(\underline{x}, \underline{p})], \quad (4.6)$$

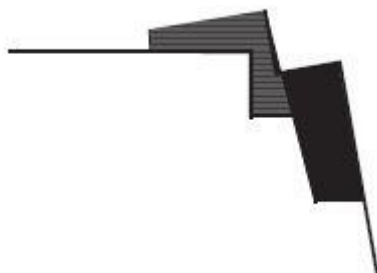
gdje je \underline{Z} druga slika. Kao što je slučaj kad god se uspoređuju uzorkovani signali, možda će biti potrebno napraviti interpolaciju kako bi se između izmjerenih uzoraka stvorila vrijednost signala. Vektorizacija definiranih signala *vec* operatorom dovodi do sljedećeg:

$$\underline{\rho} = \underset{\rho}{\operatorname{argmax}} \left[f(\underline{\rho}) = \frac{1}{2} \underline{r}^T(\underline{\rho}, \underline{Z}) \underline{r}(\underline{\rho}, \underline{Z}) \right]. \quad (4.7)$$

Ovo je nelinearni problem najmanjih kvadrata.

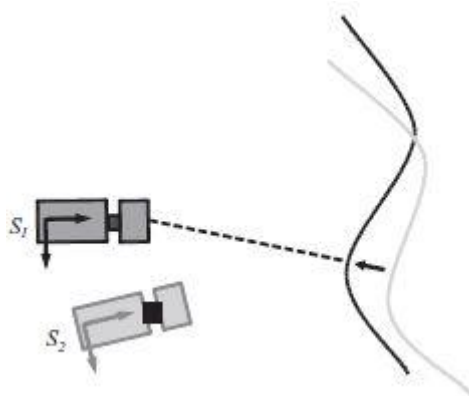
4.3. Poravnavanje geometrije površine u raspon slika

Ekvivalent poravnavanja videa je poravnavanja krivulje ili površine. Poravnavanje se može provesti u koordinatama slike ili prizora. Temelji se na pretpostavci da su dvije površine uveliko iskrivljene, ali može se i izvršiti pretraživanje izobličenja. Područje između površina jedna je odgovarajuća mjera neusklađivanja među njima koja je slična zaostacima inteziteta u videu. Za procjenu ovog područja može se upotrijebiti nekoliko metoda. Savršena registracija događa se kada je područje nula, ali algoritmi koji djeluju na ovom području konvergiraju samo do najbližeg minimuma pa s time početna pretpostavka mora biti dovoljno blizu. Princip je prikazan na slici 4.2.



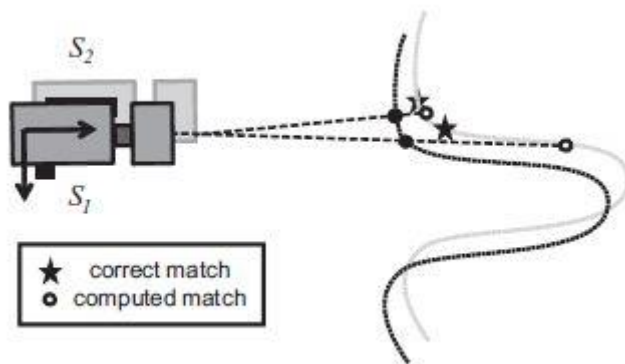
Slika 4.2. Rezidum dva lidar skena[1]

Opisani postupak može se izvesti u koordinatama prizora, ali se može koristiti i izravno prilagođavanje signala raspona. Time se procjenjuje područje pogreške registracije između krivulja u polarnim koordinatama pomoću zbroja niza duljina linija. Osnovni princip prikazan je na slici 4.3. Svaka točka na svijetloj krivulji povezana je s projektiranom ekvivalentnom točkom na krivulji tamne boje.



Slika 4.3. Usklađivanje raspona podataka u koordinatama slike[1]

Svaka točka drugog skeniranja pretvara se u koordinate ishodišta prvog skeniranja. Točke koje padaju izvan vidnog polja prvog skeniranja se zanemaruju. Točke koje su preostale upotrebljavaju se za formaciju raspona reziduum s odgovarajućim rasponima u prvom skeniranju. Interpolacijom prve slike dobit će se brže konvergiranje i preciznije usklađivanje. Takva vrsta projekcije prikazana je na slici 4.4.



Slika 4.4. Projekcijsko pridruživanje[1]

Učinkovito pridruživanje može se izvesti prikazivanjem druge slike iz perspektive prve. Točnost je visoka kada je neusklađenost skeniranja mala i kut upadanja zraka do površine mal. Usklađivanje površina je prikladno kada su površine koje se slikaju glatke. Poboljšanje u računarskoj učinkovitosti i robusnosti može se postići ako se samo parovi s velikom ili sličnom lokalnom zakrivljenošću upare.

5. USKLAĐIVANJE ZNAČAJKI LOKALIZACIJE I ESTIMACIJE POKRETA

Usklađivanje signala izbjegava problem povezivanja podataka jer se pretpostavlja da se u oba signala podaci o istom položaju podudaraju prilikom svake iteracije. Prilikom usklađivanja izmjerene s predviđenom slikom uspoređuje se element po element signala. Kada je pogreška pozicije mala ili traženje podudaranja moguće navedeni pristup je izvediv. Međutim, usavršavanjem podataka može se dostići poboljšanje računanja, radijusa i stope konvergencije te same točnosti. Nakon što se signali svedu na značajke dolazi do problema povezivanja podataka. Kao primjer može se dati vakuumski robot. Potrebno je spremati položaje svih kuteva u prostorijskoj na novu kartu značajki. Kako bi se koristila karta potrebno je prvo izvući kutove iz slike. Zatim se problem rješava na način da se pretraži pozicija u kojoj se najpreciznije usklade izmjerene i predviđene značajke [1].

5.1. Razvrstavanje i značajke

Na slici je mnogo više informacija nego što je potrebno za ograničavanje 6 stupnjeva slobode prilikom pozicioniranja robota. Zbog toga je potrebno reducirati sliku na korisne podskupove pa čak provesti i još jednu redukciju kako bi se izvele nabitnije značajke.

Razvrstavanje slika na određene značajke ima svoje prednosti:

- pojačavanje sadržaja signala,
- funkcije minimalnih vrijednosti su izoštrije,

a značajke mogu biti različite:

- rubovi regija u videu,
- točke velike zakrivljenosti u određenom rasponu podataka.

5.2. Objektivna funkcija

Slika sadrži mnogo više informacija nego što je amplituda signala jer se pojedine amplitude pojavljuju negdje na slici. Često se korisna informacija svede na to gdje se amplituda dogodila. Na primjer za robotski vakuumski usisivač bilo je moguće pregledati svaki kut prostorije u svjetskim koordinatama te bi time mjerni model mogao jednostavno stvoriti poziciju istog kuta u okviru robota:

$$\underline{x}_k = \underline{h}(\underline{\rho}, \underline{X}_k), \quad (5.1)$$

gdje je:

\underline{x}_k – pozicija kuta u okviru robota,

\underline{X}_k – pozicija kuta u okviru svijeta.

5.3. Tipične značajke i objektivne funkcije

Iz predviđene i promatrane lokacije značajki može se izvesti reziduum za svaku značajku k :

$$r_k(\underline{\rho}, \underline{X}_k) = \underline{x}_k - \underline{h}(\underline{\rho}, \underline{X}_k). \quad (5.2)$$

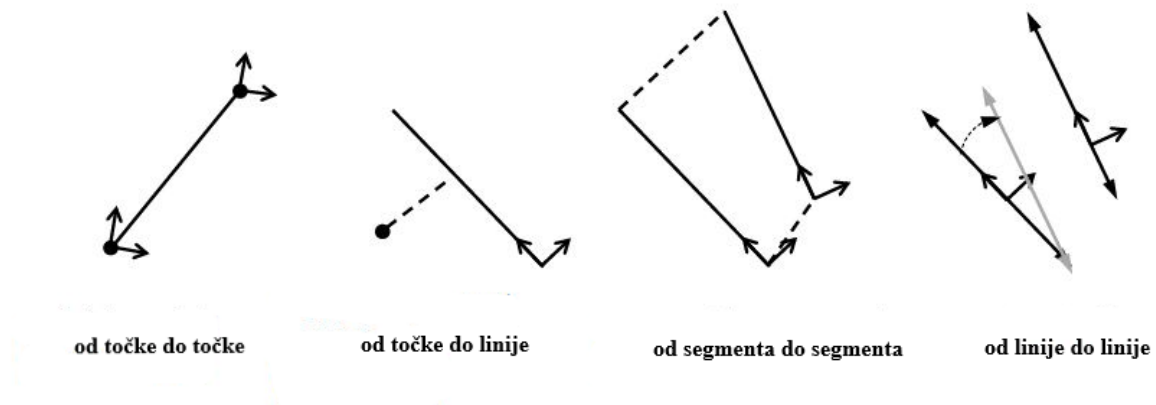
Potom ih je potrebno postaviti u jedan vektor (time se izostavlja k):

$$r(\underline{\rho}, \underline{X}) = \underline{x} - \underline{h}(\underline{\rho}, \underline{X}), \quad (5.3)$$

te pronaći lokaciju koja odgovara ostatku norme:

$$\underline{\rho} = \underset{\rho}{\operatorname{argmin}} \left[f(\underline{\rho}) = \frac{1}{2} \underline{r}^T(\underline{\rho}, \underline{Z}) \underline{r}(\underline{\rho}, \underline{Z}) \right]. \quad (5.4)$$

Kad su značajke krajnje točke lidarskih zraka ili točaka na slici u boji, izvedivo je jednostavno izračunati udaljenost između odgovarajućih točaka u bilo kojim pogodnim koordinatima. To je samo jedna opcija. Primjeri drugih planarnih korespondencija prikazani su na slici 5.1.



Slika 5.1. Planarna korespondencija značajki[1]

Različita uparivanja stvaraju rezidume različitih dimenzija. Ako su okviri modela povezani s značajkama (kako je prikazano na slici 5.1.), opažanja se mogu prikladno formulirati u koordinatama značajki.

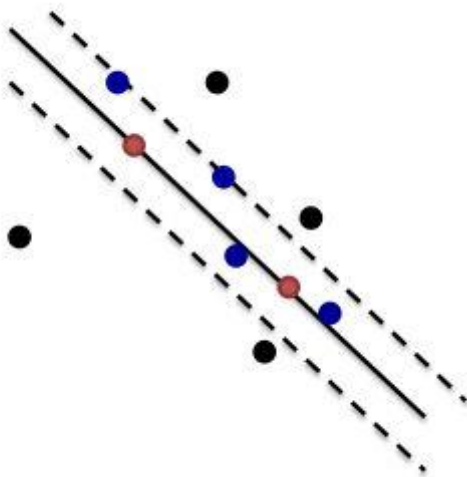
5.4. RANSAC

RANSAC je (*engl. RANdom Sample Consensus*) važan algoritam prilikom usklađivanja podataka. Općeniti je estimator parametara dizajniran kako bi se nosio s velikim razmjerima ulaznih podataka. RANSAC je tehnika uzorkovanja koja generira rješenja koristeći minimalni broj točaka podataka potrebnih za procjenu osnovnih parametara modela. S obzirom na pretpostavljena podudaranja, RANSAC određuje skup ispravnih [4].

Ideja za podešavanje linije svodi se na:

- slučajni odabir dvije točke,
- definiranje linije,
- odabir točaka unutar određene udaljenosti,
- ponavljanje postupka određeni broj puta,
- odabir linije s najtočnijom estimacijom.

Primjer navedene osnovne tehnike RANSAC-a prikazan je na slici 5.2.



Slika 5.2. Osnovni princip RANSAC-a [5]

Cilj je robusno uklapanje modela u skup podataka S koji sadrži izdatke.

Algoritam se izvodi u sljedećim koracima:

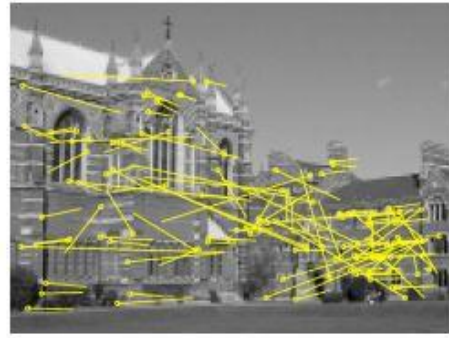
1. nasumičan odabir uzorak podatkovnih točaka (s) iz S i instancirajte model iz te podskupine,
2. određivanje skupa podatkovnih točaka S_i koji su unutar udaljenosti praga t od modela (skup S_i je kosenzusni skup uzoraka i definira izdatke S),
3. ako je podskup S_i veći od nekog praga T_i , potrebno je ponovno procijeniti model koristeći sve točke S_i ,
4. ako je veličina S_i manja od T_i , odabire se novi skup podataka te se postupak ponavlja,
5. nakon N puta ponavljanja procesa odabire se najveći koncenzus skupa S_i , a model se ponovno procjenjuje koristeći sve točke podskupine S_i .

Primjer primjene RANSAC-a dan je na slici 5.3. Potrebno je automatski izračunati interesne točke na svakoj slici, detektirati rubove te imati na umu da sve točke interesa nemaju korespondenciju na drugoj slici [5].



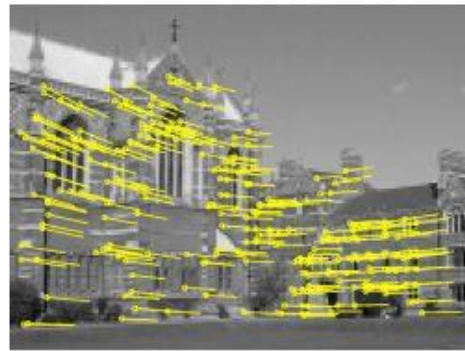
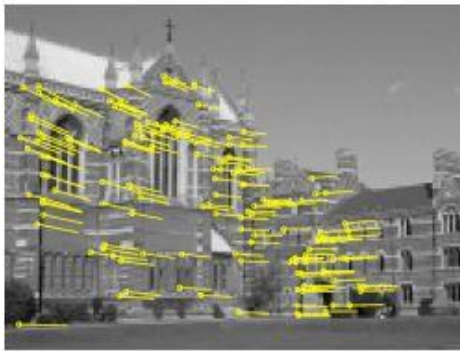
Slika 5.3. Interesne točke na slikama [5]

Nakon toga se računa skup podudaranja točaka ne temelju neke mjere sličnosti (kavadratna suma razlika inteziteta, normalizirana korelacija). Mnoge korespondencije nisu prikladne za primjenu. Na slici 5.4. su prikazani rezultati korespondencije za dan slučaj.



Slika 5.4. Korespondencije [5]

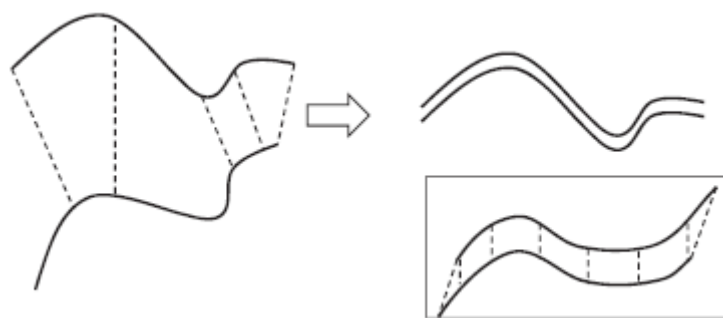
Potom slijedi optimalna estimacija i podudaranje. Rezultat je prikazan na slici 5.5.



Slika 5.5. Iteracija postupka [5]

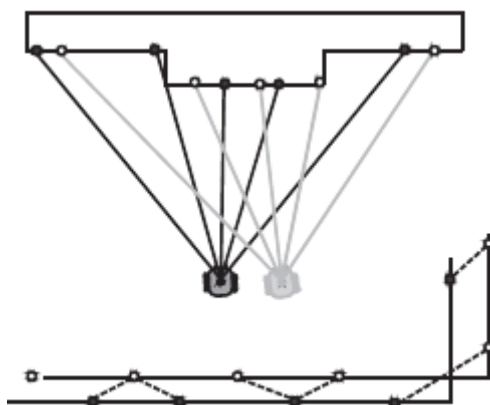
5.5. Najbliže točke u rasponu podataka

Algoritam iterativne najbliže točke koristi najbližu susjednu asocijaciju za usklađivanje podataka raspona s drugim podacima raspona ili površina. Algoritam povezuje svaku točku slike raspona s najbližom točkom, kartezijanskim koordinatama, slikom raspona ili površinom koja se poravnava. Princip je prikazan na slici 5.5..



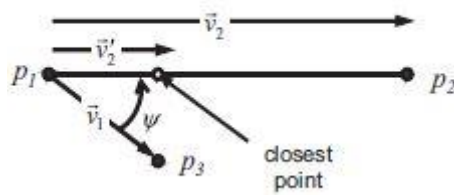
Slika 5.5. Najbliža susjedna točka[1]

S lijeve strane slike 5.5. točke se na donjoj krivulji podudaraju s najbližom točkom na gornjoj krivulji. Interpolacija podataka iznimno je važna pri usklađivanju raspona podataka. Nedostatak interpolacije unosi pogrešne komponente u funkciju što dovodi do lokalnog minimuma koji se ne pojavi na ispravnom poravnavanju. Glavni razlog za to je činjenica da su točke pogrešno i nedosljedno povezane, pa algoritam završava korištenjem pogrešnih točaka. Suprotno tome, povezivanje s najbližom točkom mjeri udaljenost normale do površine, dopuštajući tangencijalno gibanje.



Slika 5.6. Značaj interpolacije[1]

Rasponi točaka prilikom dva skena se virtualno ne poklapaju. Međusobno diskretnim rasponom uzoraka stvaraju se troškovi za tangencijalno gibanje. Slici 5.6. prikazuje izravan algoritam za pronalaženje nablize točke na linijskom segmentu do druge točke. Ako je točkasti produkt vektora pozitivan, onda točka na liniji od p_1 do p_2 je bliža p_3 .



Slika 5.7. Interpolacija[1]

Uvjeti za računanje bliže točke:

$$\vec{v}_1 \cdot \vec{v}_2 > 0. \quad (5.5)$$

Ako je navedeno stanje istinito, onda vektor \vec{v}_2' pokazuje novu točku. Dano je krajnjom točkom projekcije \vec{v}_1 na \vec{v}_2 . Formula za takvu projekciju glasi:

$$\vec{v}_2' = |\vec{v}_1| \cos \Psi \frac{\vec{v}_2}{|\vec{v}_2|} = |\vec{v}_1| \frac{|\vec{v}_2|}{|\vec{v}_2|} \cos \Psi \frac{\vec{v}_2}{|\vec{v}_2|} = \frac{(\vec{v}_1 \cdot \vec{v}_2)}{(\vec{v}_2 \cdot \vec{v}_2)} \vec{v}_2. \quad (5.6)$$

Ovaj algoritam bi se pozivao na oba susjeda od p_1 (lijeva i desna). Moguće je da ako je p_1 unutarnji kut, oba njegova susjeda mogu vratiti bližu točku od pridružene krajnje točke. Na slici 5.7. koncept pronalaženja najbliže točke.

6. OPTIMALNA POZICIJA

Odjeljak se odnosi na problem pronalaska pozicija koje najbolje objašnjavaju sliku. U slučaju da je slaba početna estimacija pozicije ili je uopće nema otežano je pronalaženje optimalne pozicije, odnosno svaki kut bi mogao izgledati isto u lidar podacima. Najteži problem je određivanje pozicije. Kada je dostupna dovoljno precizna početna estimacija pozicije mogu se koristiti gradijentni podaci za traženje optimalne pozicije. Ovaj problem se zove usavršavanje pozicije. Praćenje pozicije je problem u slučaju kad se robot kreće i kad su dostupne mjere ili druge sekundarne estimacije kretanja[1].

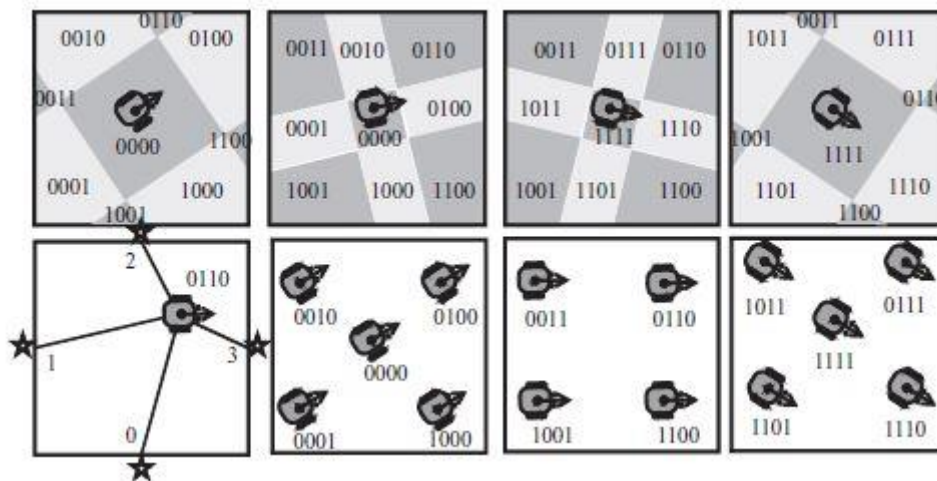
6.1. Određivanje pozicije

Određivanje pozicije nailazi na problem umetanja jer je potrebno pozicionirati robota na ispravnu poziciju na karti. Dio problema se odnosi na prepoznavanje pozicije. Prilikom pomicanja senzora, pronalažanje prve pozicije može biti teže od pronalaženja sljedećih pozicija jer se koriste zadnje pozicije kao inicijalne pozicije. Prilikom izračuna globalnog minimuma dostupne su sve numeričke tehnike. U slučaju da informacije o gradijentu nisu dovoljne za rješavanje problema potrebno je uzorkovanje. Uzorkovanjem se uzima veći broj inicijalnih estimacija te se računa najbliži lokalni minimum za svaki uzorak. Budući da je usavršavanje pozicija povećati problem u pronalaženju korijena, ostali problemi bi trebali biti mali. Pod uvjetom da je uzorak dovoljno gust i da je šum dovoljno mala, naći će se globalni minimum. Međutim, ako postoji karta ili objektni model, to može značiti da su podaci senzora koji se koriste za izradu modela dostupni, pa čak i ako nisu, bit će moguće predvidjeti mjerenja s obzirom na model mjerenja. U oba slučaja, mapiranje iz pozicija u slike može se proizvesti i izvanmrežno manipulirati kako bi se izračunalo rješenje problema prepoznavanja mjesta. Kada su karta i područje pozicije nesigurnosti dovoljno mali tablice za pretraživanje mogu biti učinkovite[2].

Za primjer određivanja pozicije uzet će se prepoznavanja mjesta. Razmatrat će se slučaj automatiziranog laserski vođenog vozila koji koristi laserski senzor za skeniranje kako bi se lokalizirao. Vozilo djeluje u području s četiri retroreflektivna fiduciala raspoređena u dijamanntu konfiguraciju. Problem umetanja vozila je obrađivanje jednog skeniranja od 4 ležaja, a zatim grubo određivanje gdje se robot nalazi kako bi se pokrenuli algoritmi vođenja. Simetrija rasporeda znači da postoje 4 potencijalna rješenja za bilo koje skeniranje, ali će se

pretpostaviti da je pravac unutar 45° u smjeru istoka. Primjer je prikazan na slici 6.1. Rješenje smanjivanje svakog skena za binarni broj od 4 znamenke odnosno:

- 0 ako je ležaj $< 45^\circ$,
- 1 ako je ležaj $> 45^\circ$.



Slika 6.1. AVG tablica pretraživanja[1]

Izvanmrežno (offline) je napisan program za uzorkovanje položaja i orijentaciju robota u rezoluciji od 1 cm i orijentaciji 1° . Za svaku poziciju, simulirane mjere su pretvorene u binarni broj gdje najznačajniji bit dolazi iz prvog povratka čiji ležaj premašuje (-45°) u odnosu na prednji smjer robota. Bitovi su povezani s ostalim ležajevima u smjeru suprotnom od kazaljke na satu. Ovi postupci smanjuju svako skeniranje za 4-bitni binarni broj koji će se koristiti u pretraživačkoj tablici. Postoji nominalno 16 mogućih tipki, ali broj 5 (0101) i 10 (1010) se ne pojavljuju. Izvanmrežni program izračunava središnju vrijednost svih pozicija povezanih s 16 mogućih ključeva. Ovaj proces stvara listu od 16 mogućih pozicija. Lista je jednostavan preglednik u svrhu brzog prepoznavanja pozicije. Kako bi opisani proces bio uspješan potrebno je imati dovoljno pozicija u tablici kako bi se osigurao početni pogodak za svaki sken unutar radijusa konvergencije točne pozicije za skeniranje.

6.2. Ugladivanje pozicije

Ako je dostupna početna estimacija moguće je koristiti gradijent informacija kako bi se pronašao lokalni minimum. Tada se mjere modela mogu linerizirati i problem se rješava kao iterativni optimizacijski problem. Prvi korak je lineariziranje. U slučaju usklađivanja signala

potrebna je funkcija zakrivljenja. Nepoznata pozicija i zakrivljenost su definirani setom podataka \underline{p} . Rezidium će tada glasiti:

$$\underline{r}(\underline{x}, \underline{p}, \underline{Z}) = \underline{z}_{obs}(\underline{x}) - \underline{z}_{pred}(\underline{x}, \underline{p}, \underline{Z}) = \underline{z}_{obs}(\underline{x}) - \underline{Z}[y(\underline{x}, \underline{p})]. \quad (6.1)$$

Potrebno se podsjetiti Newtonovog koraka:

$$\Delta \underline{p} = -[\underline{r}_p^T \underline{r}_p]^{-1} \underline{r}_p^T \underline{r}(\underline{p}), \quad (6.2)$$

gdje je:

\underline{r}_p – gradijent rezidiuma,

\underline{p} – parametri.

Tada jednačba poprima oblik:

$$\underline{r}_p = \underline{r}_p(\underline{p}, \underline{Z}) = -\underline{h}_p(\underline{p}, \underline{Z}) = -\underline{Z}_p[y(\underline{p})]. \quad (6.3)$$

Lančanim pravilom nadalje slijedi:

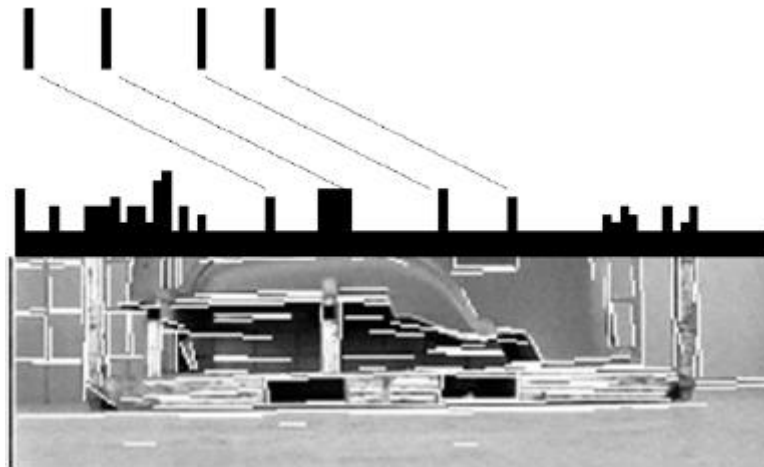
$$\underline{Z}_p = \frac{\partial}{\partial \underline{p}} \{ \underline{Z}(y(\underline{p})) \} = \left(\frac{\partial \underline{Z}}{\partial y} \right) \left(\frac{\partial y}{\partial \underline{p}} \right), \quad (6.4)$$

gdje je:

$\left(\frac{\partial \underline{Z}}{\partial y} \right)$ – gradijent slike procijenjen na $y(\underline{p})$,

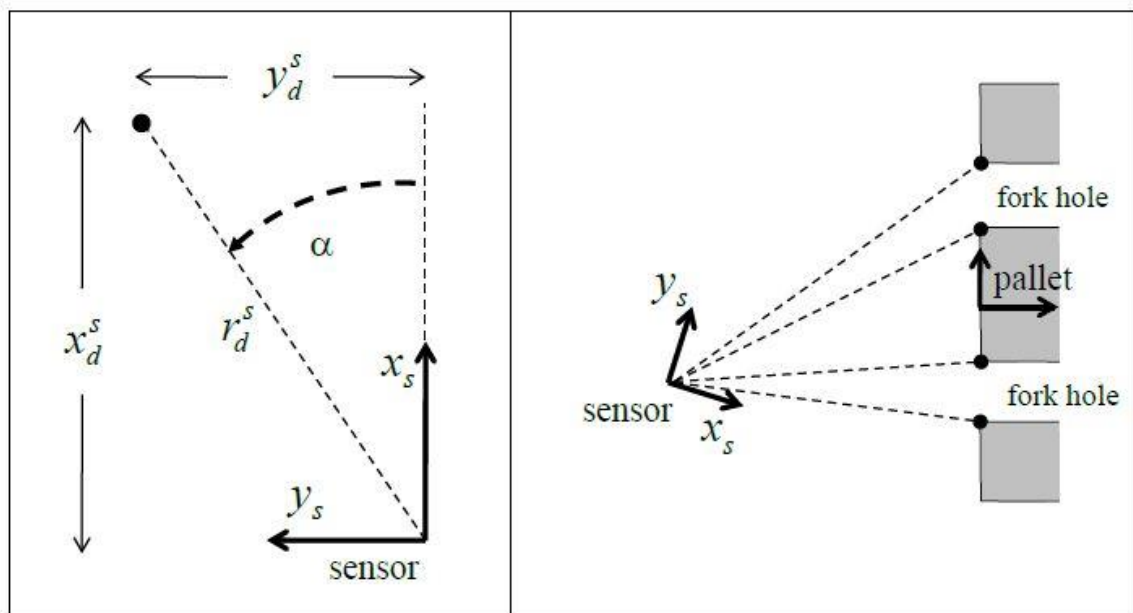
$\left(\frac{\partial y}{\partial \underline{p}} \right)$ – parametar Jacobijeve transformacije procijenjen na \underline{p} .

Kao primjer za uglađivanja pozicije dano je lokaliziranje palete. Kutovi vertikalnih rupa za vilice moraju orijentirani na karti te se pojaviti u parovima zatamnjeno/zasvijetljeno. Različiti predlošci se mogu povezati kako bi se indetificirao tip palete kako je prikazano na slici 6.2. Modeli kodiraju (ovisno o skali) omjer širine i razdvajanja od rupe.



Slika 6.2. Predložak rubova[1]

Pretpostavit će se da su pronađena 4 vertikalna kuta rupa palete. Potom je potrebno lokalizirati paletu. Lokalizacija je prikazana na slici 6.3.



Slika 6.3. Geometrija kamere i odnos prizora[1]

Senzor kamere mjeri tangentu od ležaja do točaka na prizoru. Četri vertikalna kuta ograničavaju poziciju palete u okviru kamere. Lokalizacija senzora prikazana je sljedećim izrazima:

➤ model mjerenja glasi:

$$➤ y_d^i = (f y_d^s) / x_d^s, \quad (6.5)$$

gdje je:

s – senzor,

y_d^i – proporcionalan kutu ležaja α ,

m – model okvira,

➤ vektor koordinata prizora glasi:

$$\text{➤ } r_d^s = [x_d^s y_d^s]^T, \quad (6.6)$$

➤ Jacobijeva transformacija glasi:

$$\text{➤ } H_{sd}^{id} = \frac{\partial y_d^t}{\partial r_d^s} = \left[-\frac{f y_d^s}{(x_d^s)^2} \frac{f}{x_d^s} \right]. \quad (6.7)$$

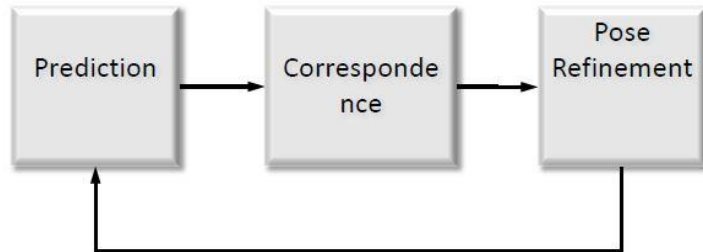
Potom se veže okvir za značajku svake od četiri točke. Tada Jacobijeva matrica poprima oblik:

$$H_{sm}^{id} = \frac{\partial \rho_d^s}{\partial \rho_m^s} = \begin{bmatrix} 1 & 0 & -(s\psi x_d^m + c\psi y_d^m) \\ 0 & 1 & (c\psi x_d^m - s\psi y_d^m) \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -(y_d^s - y_m^s) \\ 0 & 1 & (x_d^s - x_m^s) \\ 0 & 0 & 1 \end{bmatrix}. \quad (6.8)$$

Koristit će se samo dva prva retka. Postupak se potom ponavlja do konvergencije.

6.3. Praćenje pozicije

Prilikom praćenja pozicije predviđaju se značajke lokacija temeljenih na sekundarnoj procjeni kretanja te prenose na algoritam ugađivanja pozicije. Osnovni princip je prikazan na slici 6.4.



Slika 6.4. Osnovni princip praćenja pozicije[2]

Ako su potrebne brzine onda sekundarna estimacija nije potrebna. Brzina kamere se može izračunati iz brzina značajki. Prvo je potrebno linearizirati mjere modela wrt senzora pokreta:

$$\Delta \underline{x}_k = H(\underline{p}, \underline{x}_k) \Delta \underline{\rho}. \quad (6.9)$$

Potom podijeliti sve sa Δt :

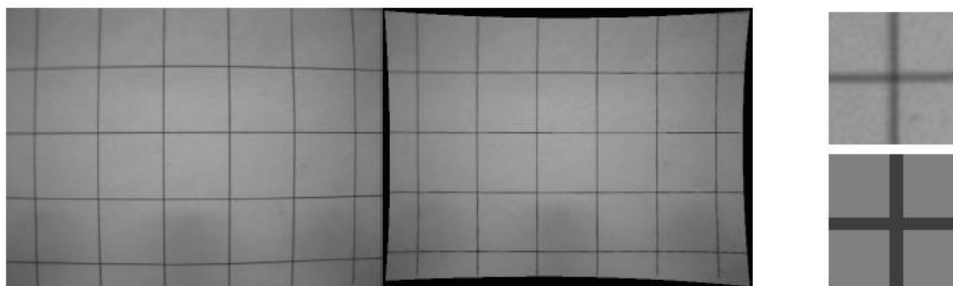
$$\dot{\underline{x}}_k = H(\underline{x}, \underline{p}, \underline{Z}) \dot{\underline{\rho}}. \quad (6.10)$$

Za primjer objasniti će izrada i praćenje podnih mozaika koji je prikazan na slici 6.5. Svi sustavi koji koriste mapiranje za lokalizaciju su vizualni pratitelji. Stoga je vizualno praćenje jednoj od najvažnijih algoritama u mobilnoj robotici. Model sustava Kalmanovog filtra iznosi procjenu invertriranog gibanja. Model mjerenja je mehanizam predviđanja. Podni mozaici su korišteni kao karta. Značajke u slici su korelirani s predikcijama baziranim na karti. Moguća je postići preciznost manju od milimetra i brzinu od 97 km/h [2].



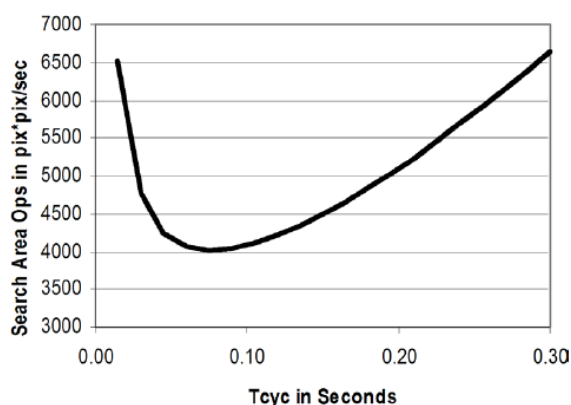
Slika 6.5. Podni mozaici [2]

Potreban je široki FOV objektiv jer je kamera iznimno blizu poda te veliki otisak slike. Postupak se svodi na slikanje rešetke i izračuna funkcija izobličenja objektiva. Potom se invertiraju distorzije kako bi se popravila slika (slika 6.6).



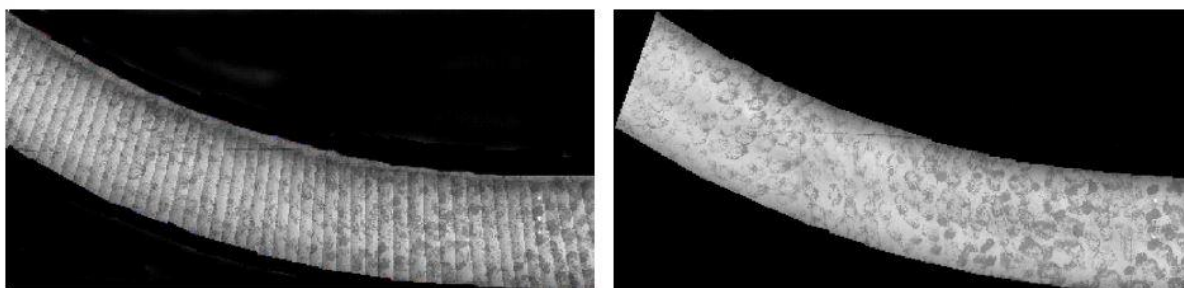
Slika 6.6. Postupak poboljšavanja karte [2]

Može se evaluirati brzina praćenja ažuriranja. Na slici 6.7. se može primjetiti da se slučajna greška ažuriranjem nadograđuje sporo, a sustavna greška brzo.



Slika 6.7. Brzina praćenja ažuriranja [2]

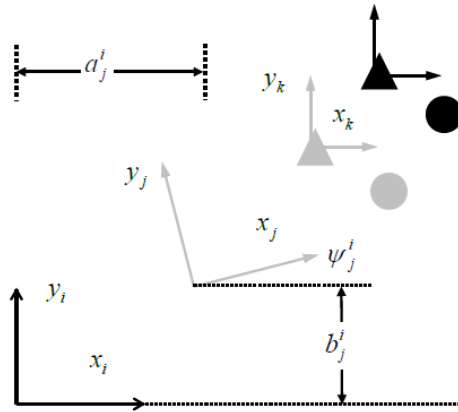
Mozaik se potom izvodi vrlo jednostavno zbog toga jer su rasponi poznati, okolina je ravina (nema distorzija) te je gibanja otprilike poznato (odometrija). Svijetla središta spajaju se u jednostavnu dugu tanku sliku (slika 6.8.). [2]



Slika 6.8. 1D mozaik slika [2]

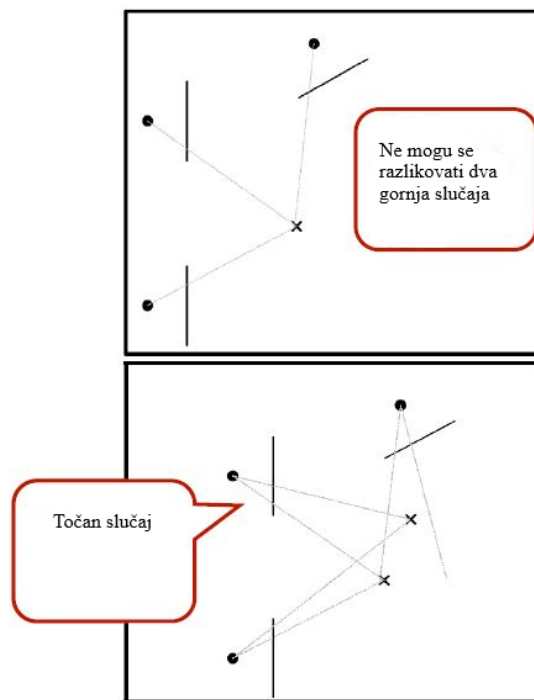
Nije potrebno riješiti rotacije svake značajke. Usklađivanje slike izvedeno je lijevo posloženom pozicijom u Jacobijevoj matrici:

$$\begin{bmatrix} \Delta a_k^i \\ \Delta b_k^i \end{bmatrix} = \begin{bmatrix} 1 & 0 & -(b_k^i - b_j^i) \\ 0 & 1 & (a_k^i - a_j^i) \end{bmatrix} \begin{bmatrix} \Delta a_j^i \\ \Delta b_j^i \end{bmatrix}. \quad (6.11)$$



Slika 6.9. Registracija značajki u 2D-u [2]

Vizualna odometrija rješava problem praćenja, ali se pretpostavlja da je protok uzrokovan gibanjem fotoaparata. Cilj je prenaći putanju kamere. Osnovna matematika jednaka je kao i u uglađivanju pozicije. Vektor stanja predstavlja diferencijalno gibanje. Temelj vizualne odometrije je stvaranje korespondencija između niza očitavanja senzora kako vozilo (kamera) mijenja položaj. Zatim sustav rješava gibanje (slika 6.9.) koje objašnjava korespondenciju. Ako je potrebno, proces sekundarne integracije izračunava apsolutnu poziciju iz niza estimacija relativnih pozicija. Značajke na dvostrukoj dubini su u skladu s dvostrukom translacijom. Ni na koji način se ne može reći koji je od prva dva slučaja točan. Međutim, promjena orijentacije može biti izmjerena bez poznavanja dubine, ako se zna da je gibanja rotacija (slika 6.10.). Razlikovanje rotacije od translacije je već drugi problem. Neki se koriste SFM kako bi dobili dubinu. Vizodo je poseban slučaj gdje se ignorira oblik izlaza. Stereo je druga alternativa. Potrebne su dvije značajke za određivanje 2D gibanja. FOV širina je dovoljna da bi se vidjele dobro razdvojene značajke koje pomažu u problemu razlikovanja rotacije od translacije [2].



Slika 6.10. Određivanje dubine [2]

7. ZAKLJUČAK

Pozicioniranje na temelju percepcije, umjesto da je ezoterično, je temeljna sposobnost mobilnih robota. Obradene tehnologije: ugađivanje pozicije, registracija, vizualno praćenje i vizualna odometrija, slične su u materiji, ali različite u primjeni. Sve ovise o rješenjima predikcije, korespondencije i registriranja. Ugađivanje pozicije i registracija ovise o reziduumu između stvarnih i predviđenih značajki, dok vizualno praćenje i odometrija ovise o reziduumu između dva seta realnih značajki lokacija. Postojanje prethodne karte ili modela je od ključne važnosti za daljnu obradu. Prethodne karte omogućuju iteraciju estimacije pokreta.

8. LITERATURA

- [1] Alonzo Kelly, "Mobile Robotics - Mathematics, Models, and Methods", *Visual Localization and Motion Estimation*, New York, NY, USA, 2013, pp. 593-623. ISBN: 978-1-107-03115-9
- [2] S interneta: <http://www.cs.cmu.edu/~alonzo/books/map3.pdf>, pristupljeno 29.04.2020.
- [3] Sjö K., Paul C., Jensfelt P., Object Localization using Bearing Only Visual Detection, a Royal Institute of Technology (KTH), Centre for Autonomous Systems, SE-100 44 Stockholm, Sweden
- [4] S interneta: http://www.cse.yorku.ca/~kosta/CompVis_Notes/ransac.pdf, pristupljeno 30.04.2020.
- [5] S interneta: http://www.csc.kth.se/~madry/courses/mvg10/Attachments/03_Presentation02_Ch4_Pravin.pdf, pristupljeno 30.04.2020.
- [6] Luo J., Pronobis A., Caputo B., Jensfelt P., The KTH-IDOL2 Database, KTH, SE-100 44 Stockholm, Sweden, 13 listopada. 2006.

9. POPIS SLIKA

Slika 2.1. Opći okvir	2
Slika 3.1. Detekcija rubova za vilica-rupa pronalazak (lijevo- početna slika s paletama, sredina- detektirani svi rubovi, desno- pronađene rupe za vilicu)	8
Slika 3.2. Lokalizacija bazirana na percepciji	9
Slika 4.1. Korelacija	12
Slika 4.2. Rezidum dva lidar skena	13
Slika 4.3. Usklađivanje raspona podataka u koordinatama slike	14
Slika 4.4. Projekcijsko pridruživanje	14
Slika 5.1. Planarna korespondencija značajki	16
Slika 5.2. Osnovni princip RANSAC-a.....	17
Slika 5.3. Interesne točke na slikama	18
Slika 5.4. Korespondencije.....	19
Slika 5.5. Iteracija postupka	19
Slika 5.5. Najbliža susjedna točka.....	20
Slika 5.6. Značaj interpolacije	20
Slika 5.7. Interpolacija.....	21
Slika 6.1. AVG tablica pretraživanja.....	23
Slika 6.2. Predložak rubova.....	25
Slika 6.3. Geometrija kamere i odnos prizora	25
Slika 6.4. Osnovni princip praćenja pozicije.....	26
Slika 6.5. Podni mozaici.....	27
Slika 6.6. Postupak poboljšavanja karte	28
Slika 6.7. Brzina praćenja ažuriranja.....	28
Slika 6.8. 1D mozaik slika.....	28
Slika 6.9. Registracija značajki u 2D-u	29
Slika 6.10. Određivanje dubine	30