

Univerza v Ljubljani
Fakulteta za elektrotehniko

Simon Tomažič

**LOKALIZACIJA V NOTRANJEM OKOLJU
Z UPORABO PAMETNEGA TELEFONA**

DOKTORSKA DISERTACIJA

Ljubljana, 2016

Univerza v Ljubljani

Fakulteta za elektrotehniko

Simon Tomažič

**LOKALIZACIJA V NOTRANJEM OKOLJU
Z UPORABO PAMETNEGA TELEFONA**

DOKTORSKA DISERTACIJA

Mentor: prof. dr. Igor Škrjanc, univ. dipl. inž. el.

Ljubljana, 2016



IZJAVA

Spodaj podpisani/-a, Simon Tomažič, z vpisno številko 64060450 s svojim podpisom izjavljam, da sem avtor/-ica zaključnega dela z naslovom:

LOKALIZACIJA V NOTRANJEM OKOLJU Z UPORABO PAMETNEGA TELEFONA
INDOOR LOCALIZATION WITH SMARTPHONE

S svojim podpisom potrjujem:

- da je predloženo zaključno delo rezultat mojega samostojnega raziskovalnega dela in da so vsa dela in mnenja drugih avtorjev skladno s fakultetnimi navodili citirana in navedena v seznamu virov, ki je sestavni del predloženega zaključnega dela,
- da je elektronska oblika zaključnega dela identična predloženi tiskani obliki istega dela,
- da na Univerzo v Ljubljani neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravici shranitve avtorskega dela v elektronski obliki in reproduciranja ter pravico omogočanja javnega dostopa do avtorskega dela na svetovnem spletu preko Repozitorija Univerze v Ljubljani (RUL).

V Ljubljani, 7. 11. 2016

Podpis avtorja/-ice:

Zahvala

Zahvaljujem se mentorju prof. dr. Igorju Škrjancu za vse strokovne nasvete in pomoč tekom doktorskega študija. Zahvaljujem se tudi vsem sodelavcem iz Laboratorija za avtonomne mobilne sisteme in Laboratorija za modeliranje, simulacijo in vodenje za prijetno delovno vzdušje in koristne nasvete. Posebna zahvala gre moji družini in ženi Tonkici, ki so mi vedno stali ob strani in me spodbujali pri mojem delu.

Hvala vam!

Povzetek

V doktorski disertaciji obravnavamo lokalizacijo v notranjem okolju z uporabo pametnega telefona. Ta je postal skoraj nepogrešljiv pripomoček sodobnega človeka, tudi zaradi vse večje vloge na področju osebnih navigacijskih sistemov. Lokalizacija v notranjem okolju je trenutno predmet številnih raziskav in projektov, saj se še nobeden izmed obstoječih lokalizacijskih sistemov ni uveljavil v taki meri, kot so se globalni navigacijski satelitski sistemi pri lokalizaciji in navigaciji v zunanjem okolju. Razlogi za to se skrivajo v visoki ceni preciznih lokalizacijskih sistemov oz. v slabi natančnosti cenovno dostopnih sistemov. Da bi nek nov lokalizacijski sistem prišel v širšo uporabo, pa mora biti z njim dosežen kompromis, ki predvideva zadovoljivo natančnost lokalizacije za relativno nizko ceno. Sodobni pametni telefoni imajo vgrajene številne senzorje (to so inercialni senzorji, kamera, barometer) in komunikacijske module (WiFi, Bluetooth, NFC), ki omogočajo implementacijo različnih lokalizacijskih algoritmov. Da bi v čim večji meri izkoristili potencial sodobnih mobilnih naprav, smo razvili tri različne lokalizacijske sisteme, in sicer vizualno lokalizacijo, inercialni navigacijski sistem in pa radijsko lokalizacijo. V sklopu vizualne lokalizacije smo na pametnem telefonu implementirali algoritem vizualne odometrije, ki se izvaja v realnem času na dveh ločenih nitih. Pri inercialnem navigacijskem sistemu smo za namen uporabe navigacije na slepo implementirali števec korakov (pedometer) in digitalni kompas, ki temeljita na uporabi pospeškometra, žiroskopa in magnetometra. Ker sta tako vizualna odometrija kot tudi inercialni navigacijski sistem občutljiva na zunanje motnje, smo oba lokalizacijska sistema združili z uporabo razširjenega Kalmanovega filtra. Na ta način smo dobili natančen in robusten relativni lokalizacijski sistem, ki omogoča določanje položaja v prostoru glede na poznano začetno točko. Da bi lahko določali tudi globalne položaje

v prostoru, smo se lotili razvoja radijske lokalizacije, v sklopu katere smo preizkusili več različnih metod, in sicer trilateracijo, metodo prstnih odtisov in optimizacijo z rojem delcev, ki omogočajo določanje položaja v prostoru glede na meritve jakosti Bluetooth signalov. Za potrebe delovanja omenjenih metod smo zgradili nelinearne ter tudi mehke modele, ki opisujejo spreminjanje jakosti signalov v odvisnosti od razdalje do oddajnikov. Pri zajemu meritev, na podlagi katerih so bili zgrajeni modeli, je imel ključno vlogo razviti relativni lokalizacijski sistem, saj smo z njegovo uporabo na enostaven način zbrali meritve jakosti signalov v odvisnosti od razdalje do oddajnikov. Najboljši rezultati lokalizacije so bili dobljeni z uporabo mehkih modelov in metode prstnih odtisov. V tem primeru smo za 53 % ocenjenih položajev dobili napako manjšo od 0,5 m oz. za 99 % ocenjenih položajev manjšo od 1 m. Z namenom, da bi rezultate globalne lokalizacije še nekoliko izboljšali, smo implementirali fuzijo Bluetooth lokalizacijskega sistema, ki temelji na metodi prstnih odtisov in relativnega lokalizacijskega sistema, ki ga sestavljata vizualna odometrija in inercialni navigacijski sistem. Z združitvijo obeh sistemov so se napake globalnih položajev zmanjšale, saj je bila v tem primeru napaka manjša od 0,5 m pri več kot 80 % ocenjenih položajev. Z izvrstnimi rezultati se razvitemu lokalizacijskemu sistemu odpirajo številna vrata, saj se lahko uporablja pri vodenju avtonomnih mobilnih sistemov ali ljudi znotraj zgradb.

Abstract

The doctoral thesis deals with localization in indoor environment by using a smartphone. The latter has in a way become an indispensable tool of modern man, due to, among other things, the ever-increasing role in the field of personal navigation systems. Indoor localization is currently the subject of numerous studies and projects since none of the existing positioning systems have been established in such a way as the global navigation satellite systems for localization and navigation in the outdoor environment. The reasons for this lie in the high cost of precision positioning systems and in the poor performance of affordable systems. A new indoor positioning system may come into wider use if a compromise is reached, i.e. when sufficient accuracy and precision of localization are provided for a relatively low price. Modern smartphones are equipped with numerous sensors (inertial sensors, camera, barometer) and communications modules (WiFi, Bluetooth, NFC), which enable the implementation of various localization algorithms. To exploit the potential of modern mobile devices as much as possible, we have developed three different localization systems, namely visual localization, inertial navigation system and radio localization. Within the visual localization a visual odometry algorithm was implemented on a smartphone, where it can run in real time in two separate threads. On inertial navigation system a step counter (pedometer) and a digital compass, which are based on the use of accelerometer, gyroscope and magnetometer, were implemented for the purpose of dead reckoning. Since visual odometry as well as the inertial navigation system are sensitive to external disturbances, both localization systems were combined with the extended Kalman filter. In this way, an accurate and robust relative localization system which enables determining the position relative to a known starting point was obtained. In order to determine global positions in indoor environment, we undertook the

development of radio localization, under which we tested several different methods, namely trilateration, fingerprint-based method and particle swarm optimization, which can operate with analysis of measurement of Bluetooth signals' strengths. For the needs of the operation of aforementioned methods, we built a non-linear, as well as fuzzy models, which describe the change in signals' strengths as a function of the distance to the transmitters. The developed relative localization system played a key role at measurement acquisition on which the models were built, since signals' strengths as a function of the distance to the transmitters can simply be obtained by its use. The best localization results were obtained by using the fuzzy models and the fingerprint-based method. In this case 53 % of the position errors were smaller than 0.5 m and 99 % of the position errors were smaller than 1 m. In order to get even better results of the global localization, a fusion of the global positioning system based on Bluetooth fingerprints and relative localization system, which consists of visual odometry and inertial navigation system, was implemented. By combining these two systems, the errors of the global positions were reduced, since in this case more than 80 % of the errors were smaller than 0.5 m. With excellent results many doors open for the developed localization system, as it can be used in guiding autonomous mobile systems or people inside buildings.

Vsebina

Seznam slik	xvii
Seznam tabel	xxvii
1 Uvod.....	1
2 Pregled obstoječih metod za lokalizacijo v notranjem okolju	7
2.1 Lokalizacija z uporabo kamere.....	8
2.2 Lokalizacija z uporabo inercialnih senzorjev	21
2.3 Lokalizacija z uporabo radijskih signalov	26
2.4 Fuzija različnih metod lokalizacije.....	34
3 Vizualna odometrija	39
3.1 Uvod v monokularno vizualno odometrijo	39
3.2 Kalibracija kamere.....	40
3.2.1 Model kamere luknjičarke.....	40
3.2.2 Uvod v kalibracijo.....	42
3.2.3 Ocena homografije med kalibracijskim objektom in njegovo sliko	44
3.2.4 Ocena intrinzičnih in ekstrinzičnih parametrov	45
3.2.5 Določanje rotacijske matrike.....	46
3.2.6 Ocena maksimalne verjetnosti	47
3.2.7 Kompenzacija radialne distorzije.....	47
3.2.8 Rezultati kalibracije kamere.....	50
3.3 Optični tok.....	56
3.3.1 Kanade-Lucas-Tomasi sledilnik značilnih točk	57
3.3.2 Piramidni KLT algoritem	58
3.3.3 Standardni KLT algoritem	61
3.3.4 Iskanje značilnih točk.....	63
3.3.5 Sobel operator	68
3.4 RANSAC.....	70
3.5 Togi premik	73

3.5.1	Izračun togega premika	75
3.6	Opis monokularne vizualne odometrije.....	78
3.7	Rezultati preizkusa vizualne odometrije.....	84
4	Inercialni navigacijski sistem	91
4.1	Digitalni kompas.....	92
4.2	Pedometer	95
4.2.1	Ocenjevanje dolžine koraka	97
4.3	Rezultati inercialnega navigacijskega sistema	98
5	Fuzija vizualne odometrije in inercialnega navigacijskega sistema.....	103
5.1	Rezultati fuzije vizualne odometrije in inercialnega navigacijskega sistema	107
6	Lokalizacija z uporabo radijskih signalov	113
6.1	Bluetooth	115
6.1.1	Bluetooth Low Energy	116
6.1.2	BLE oddajniki	123
6.2	Izgradnja modelov jakosti signalov	128
6.2.1	Meritve jakosti signalov z uporabo pametnega telefona	128
6.2.2	Prileganje modela na meritve jakosti signalov	130
6.2.3	Mehki identifikacijski modeli	135
6.2.4	Rezultati gradnje mehkih modelov	146
6.3	Določanje trenutnega položaja s trilateracijo	148
6.3.1	Določanje položaja z uporabo lineariziranega modela	150
6.3.2	Določanje položaja z uporabo nelinearne optimizacije.....	154
6.4	Rezultati lokalizacije na osnovi trilateracije.....	162
6.4.1	Lokalizacija z uporabo trilateracije in metode linearnih najmanjših kvadratov	163
6.4.2	Lokalizacija z uporabo trilateracije in omejene nelinearne optimizacije	165
6.5	Lokalizacija z uporabo optimizacije z rojem delcev	169
6.5.1	Optimizacija z rojem delcev	169
6.5.2	Rezultati lokalizacije na osnovi optimizacije z rojem delcev	173
6.5.3	Vpliv parametrov metode PSO na natančnost lokalizacije	182
6.5.4	Vpliv šuma na natančnost lokalizacije	185
6.6	Lokalizacija z uporabo zemljevida prstnih odtisov	197
6.6.1	Izgradnja zemljevida prstnih odtisov	197
6.6.2	Metode iskanja najbližjih sosedov	200

6.6.3	Rezultati lokalizacije na osnovi prstnih odtisov.....	209
6.7	Diskusija.....	223
7	Fuzija vizualne odometrije, inercialnega navigacijskega sistema in Bluetooth lokalizacije	227
7.1	Kalmanov filter.....	227
7.1.1	Model konstantne hitrosti.....	228
7.2	Postopek fuzije lokalizacijskih metod	230
7.3	Rezultati fuzije lokalizacijskih metod	235
8	Zaključek.....	243
9	Doprinos k znanosti	247
A	Singularni razcep.....	251
B	Kovariančna matrika šuma procesa.....	253
	Literatura	255

Seznam slik

Slika 2.1: Komponente sodobnih pametnih telefonov	7
Slika 2.2: Google zemljevid muzeja »Smithsonian National Air and Space Museum« v Washingtonu na Android telefonu [24].....	8
Slika 2.3: Shema lokalizacijskega sistema, ki so ga razvili na TUM [47].....	11
Slika 2.4: PTAM (levo: sledenje značilnih točk, desno: oblak značilnih točk) [50] .	14
Slika 2.5: Rekonstrukcija točk v oblaku	16
Slika 2.6: 3D zemljevid točk v oblaku	17
Slika 2.7: Voziček za gradnjo 3D zemljevida točk	18
Slika 2.8: Segmentacija oblaka točk na ravnine, ki so osnova za projektivne transformacije.....	19
Slika 2.9: Lokalizacijski sistem, ki sta ga predlagala Nguyen in Lee [72]	20
Slika 2.10: Žiroskop meri relativne zasuke mobilne naprave po vseh treh oseh	22
Slika 2.11: Magnetno polje Zemlje [81]	23
Slika 2.12: Detekcija korakov in gibanja [79]	25
Slika 2.13: Inercialni lokalizacijski sistem [12]	26
Slika 2.14: Shematski prikaz pristopa s prstnimi odtisi [113]	30
Slika 2.15: Minimalno razpeto drevo 3D prstnih odtisov [128]	33
Slika 2.16: Različne možnosti izvedbe WiFi/Bluetooth lokalizacije [124]	33
Slika 2.17: Fuzija različnih metod za natančno lokalizacijo v zaprtem prostoru [136]	35
Slika 2.18: Shematski prikaz sistema lokalizacije, ki so ga predlagali Kannan in drugi [134].....	36
Slika 3.1: Geometrija modela kamere luknjičarke	40
Slika 3.2: Model kamere luknjičarke (ang. pinhole camera)	42

Slika 3.3: Levo je prikazano popačenje slikovnih točk pri negativni radialni distorziji in desno pri pozitivni radialni distorziji	47
Slika 3.4: V primeru, da leča in senzor nista vzporedna, se pojavi tangencialna distorzija	49
Slika 3.5: Detekcija vogalov pri kalibraciji kamere	51
Slika 3.6: Pri kalibraciji smo šahovnico slikali pod različnimi koti	51
Slika 3.7: Absolutna razlika med popačeno in nepopačeno sliko	52
Slika 3.8: Srednja vrednost pogreška pri re-projekciji	53
Slika 3.9: Pogrešek pri re-projekciji v slikovnih točkah (za vseh 7 slik)	53
Slika 3.10: Primer slike, za katero smo izračunali ekstrinzične parametre	54
Slika 3.11: Vizualizacija ekstrinzičnih parametrov	55
Slika 3.12: Transformacije med koordinatnimi sistemi podlage (P), šahovnice (T) in kamere (C)	55
Slika 3.13: Sledenje značilnim točkam z uporabo algoritma KLT skozi sekvenco slik	58
Slika 3.14: Ocena napake pri iskanju premika med slikama I in J	58
Slika 3.15: Primer piramide s štirimi nivoji ($L_m = 3$)	59
Slika 3.16: Optični tok, ki ga dobimo z uporabo piramidnega KLT algoritma na pametnem telefonu	60
Slika 3.17: (a) »ravno« področje – v okolici ni sprememb intenzitete, (b) rob – ni sprememb intenzitete v smeri roba, (c) vogal – velike spremembe intenzitete v vseh smereh	65
Slika 3.18: Področja, ki povedo, ali smo detektirali vogal (zeleno področje), rob (sivo in modro področje) ali nič (roza področje)	67
Slika 3.19: Detektirani vogali z uporabo algoritma Shi-Tomasi pri velikosti slike 1280×720 slikovnih točk	68
Slika 3.20: Detektirani vogali z uporabo algoritma Shi-Tomasi pri velikosti slike 320×240 slikovnih točk	68
Slika 3.21: Velikost gradienta slike, ki ga dobimo z uporabo operatorja Sobel	69
Slika 3.22: Primerjava metode najmanjših kvadratov (a) in metode RANSAC (b) ..	71
Slika 3.23: Togi premik	74

Slika 3.24: Togi premik med svetovnim koordinatnim sistemom in koordinatnim sistemom kamere.....	74
Slika 3.25: Togi premik prve množice točk (modri križci) za kot $\vartheta = 60^\circ$ in translacijo $[2 \ 2]^T$ v drugo množico točk (rdeči krogci).....	77
Slika 3.26: Prekrivanje točk iz prve množice in točk iz druge množice, ki so premaknjene glede na izračunani rotacijski in translacijski matriki	78
Slika 3.27: Diagram poteka izračuna togega premika v 2D.....	79
Slika 3.28: Togi premik v 2D iz trenutnega K.S. v svetovni K.S. dobimo tako, da upoštevamo sprotne izračune za togi premik iz ključnega K.S. v trenutni K.S.....	81
Slika 3.29: Preverjanje ustreznosti togega premika z algoritmom Ransac	82
Slika 3.30: Togi premik gWC iz svetovnega K.S. v K.S. kamere.....	83
Slika 3.31: Določanje relativnega položaja pešca z vizualno odometrijo.....	84
Slika 3.32: Težave vizualne odometrije pri določanju zasuka.....	85
Slika 3.33: Spreminjanje zasuka (določenega z VO) pri hoji po poti, ki je prikazana na sliki 3.32 (eksperiment je bil petkrat ponovljen)	86
Slika 3.34: Srednja kvadratna napaka pri določanju zasuka z vizualno odometrijo..	87
Slika 3.35: Razdalja do začetne točke pri hoji po ravni črti dolžine 10,2 m (eksperiment je bil petkrat ponovljen)	88
Slika 3.36: Relativna napaka pri izračunu prehojene razdalje z vizualno odometrijo	88
Slika 4.1: Svetovni koordinatni sistem, ki ga uporablja rotacijski vektor [169].....	92
Slika 4.2: Koordinatni sistem pametnega telefona [169]	93
Slika 4.3: Fuzija podatkov iz Rotacijskega vektorja in kalibriranega žiroskopa	94
Slika 4.4: Shema pedometra.....	95
Slika 4.5: Zgoraj je prikazan neobdelan vertikalni pospešek in spodaj filtriran vertikalni pospešek.....	97
Slika 4.6: Rezultat določanja položaja pešca z uporabo inercialnega navigacijskega sistema.....	99
Slika 4.7: Spreminjanje zasuka (določenega z INS) pri hoji po poti, ki je prikazana na sliki 4.6 (eksperiment je bil petkrat ponovljen)	100
Slika 4.8: Srednja kvadratna napaka pri določanju zasuka z digitalnim kompasom	101
Slika 4.9: Razdalja do začetne točke pri hoji po ravni črti dolžine 10,2 m (eksperiment je bil petkrat ponovljen)	101

Slika 4.10: Relativna napaka pri izračunu prehojene razdalje s pedometrom	102
Slika 5.1: Shema fuzije vizualne odometrije in inercialnega navigacijskega sistema z razširjenim Kalmanovim filtrom.....	105
Slika 5.2: Določanje trajektorije gibanja pešca s fuzijo vizualne odometrije in inercialnega navigacijskega sistema.....	108
Slika 5.3: Spreminjanje zasuka pešca glede na njegovo začetno orientacijo med hojo po referenčni poti, ki je prikazana na sliki 5.2	109
Slika 5.4: Srednja kvadratna napaka pri določanju zasuka z vizualno odometrijo, inercialnim navigacijskim sistemom in fuzijo slednjih dveh algoritmov	110
Slika 5.5: Spreminjanje razdalje do začetne točke med hojo po ravni črti dolžine 10,2 m.....	111
Slika 5.6: Relativna napaka pri izračunu prehojene razdalje z vizualno odometrijo, inercialnim navigacijskim sistemom in fuzijo slednjih dveh algoritmov	111
Slika 5.7: Določanje položajev in zasukov pešca pri hoji po sklenjeni poti s fuzijo vizualne odometrije in inercialnega navigacijskega sistema.....	112
Slika 6.1: BLE frekvenčni kanali	117
Slika 6.2: BLE sklad.....	120
Slika 6.3: Hierarhija GATT profila	121
Slika 6.4: Pasivno (zgoraj) in aktivno (spodaj) skeniranje.....	122
Slika 6.5: BLE oddajniki	124
Slika 6.6: Podatki BLE oglaševalskega paketa [189]	124
Slika 6.7: Podatki oglaševalskega paketa pri »iBeacon« oddajnikih	126
Slika 6.8: Podatki treh oglaševalskih paketkov pri »Eddystone« oddajnikih	127
Slika 6.9: Položaj Bluetooth oddajnikov v prostoru	129
Slika 6.10: Vsi položaji, v katerih so bile zajete meritve jakosti signalov	130
Slika 6.11: Meritve jakosti signalov (iz oddajnika z MAC naslovom E3:10:72:E5:75:09) po poteh, ki so označene na sliki 6.9	131
Slika 6.12: Prileganje matematičnega modela (6.2) na meritve jakosti Bluetooth signalov oddajnika z MAC naslovom E3:10:72:E5:75:09.....	132
Slika 6.13: Širjenje Bluetooth signala (oddajnika z MAC naslovom E3:10:72:E5:75:09) po prostoru.....	133

Slika 6.14: Meritve jakosti signalov (iz oddajnika z MAC naslovom D9:50:3B:F6:AA:46) po poteh, ki so označene na sliki 6.9	134
Slika 6.15: Prileganje matematičnega modela (6.2) na meritve jakosti Bluetooth signalov oddajnika z MAC naslovom D9:50:3B:F6:AA:46.....	134
Slika 6.16: Razpon napak pri posameznih modelih padanja jakosti signalov	135
Slika 6.17: Mehki model sestavljen iz več lokalnih linearnih modelov	137
Slika 6.18: Primer mehkega modela (a), ki je sestavljen iz treh linearnih modelov. Njihov prispevek določajo utežne funkcije Φ_k (b).....	137
Slika 6.19: Primer rojenja s tremi roji.....	138
Slika 6.20: Prileganja mehkega modela (s petimi linearnimi pod modeli) na meritve jakosti Bluetooth signalov oddajnika z MAC naslovom E3:10:72:E5:75:09	147
Slika 6.21: Prileganja mehkega modela (z desetimi linearnimi pod modeli) na meritve jakosti Bluetooth signalov oddajnika z MAC naslovom D9:50:3B:F6:AA:46	147
Slika 6.22: Razpon napak za vse identificirane mehke modele	148
Slika 6.23: Trilateracija.....	149
Slika 6.24: Realen problem trilateracije, kjer se krožnice ne sekajo v eni točki.....	150
Slika 6.25: Zmanjševanje območja zaupanja pri optimizaciji z metodo omejenega koraka.....	157
Slika 6.26: Položaj Bluetooth oddajnikov v prostoru	163
Slika 6.27: Določanje položaja s trilateracijo in linearno metodo najmanjših kvadratov	164
Slika 6.28: Napaka pri določanju položaja s trilateracijo in linearno metodo najmanjših kvadratov	165
Slika 6.29: Natančnost lokalizacije podana s kumulativno porazdelitvijo verjetnosti napake	166
Slika 6.30: Določanje položaja z uporabo trilateracije in omejene nelinearne optimizacije.....	167
Slika 6.31: Napaka pri določanju položaja s trilateracijo in omejeno nelinearno optimizacijo.....	167
Slika 6.32: Natančnost lokalizacije, podana s kumulativno porazdelitvijo verjetnosti napake	168

Slika 6.33: Določanje položaja z uporabo trilateracije in (neomejene) nelinearne optimizacije	168
Slika 6.34: Napaka pri določanju položaja s trilateracijo in (neomejeno) nelinearno optimizacijo	169
Slika 6.35: Model gibanja delcev	171
Slika 6.36: Položaj Bluetooth oddajnikov v prostoru	174
Slika 6.37: Dejanska pot, ki je bila določena z vizualno odometrijo in inercialnim navigacijskim sistemom	174
Slika 6.38: Določanje položaja z uporabo nelinearnih modelov in optimizacijske metode PSO	176
Slika 6.39: Napake pri določanju položaja z metodo PSO (nelinearni modeli)	176
Slika 6.40: Določanje položaja z uporabo mehkih modelov in optimizacijske metode PSO (velikost iskalnega območja: $0,1 \times 0,1$ m)	177
Slika 6.41: Napake pri določanju položaja z metodo PSO (mehki modeli)	178
Slika 6.42: Natančnost lokalizacije, podana s kumulativno porazdelitvijo verjetnosti napake pri uporabi nelinearnih oz. mehkih modelov	178
Slika 6.43: Določanje položaja z uporabo mehkih modelov in optimizacijske metode PSO (velikost iskalnega območja: $0,8 \times 0,8$ m)	179
Slika 6.44: Glajenje poti (ki je sestavljena iz položajev, dobljenih z uporabo mehkih modelov in metode PSO) z uporabo nizkopasovnega filtra	180
Slika 6.45: Napake pri določanju položaja z metodo PSO (mehki modeli) v primeru, ko je za glajenje poti uporabljen nizkopasovni filter	181
Slika 6.46: Natančnost lokalizacije, podana s kumulativno porazdelitvijo verjetnosti napake pri uporabi nelinearnih oz. mehkih modelov in metode PSO. V kombinaciji z mehki modeli je bil uporabljen nizkopasovni filter za glajenje poti	181
Slika 6.47: Vpliv maksimalnega števila iteracij in parametra ω (metode PSO) na natančnost lokalizacije	182
Slika 6.48: Vpliv parametrov φp in φg (metode PSO) na natančnost lokalizacije	183
Slika 6.49: Vpliv velikosti iskalnega območja in števila delcev (metode PSO) na natančnost lokalizacije	184

Slika 6.50: Normalizirana vsota napak (pri določanju položaja) v odvisnosti od števila upoštevanih oddajnikov pri treh različnih velikostih iskalnega območja metode PSO	185
Slika 6.51: Možni položaji pri upoštevanju vseh kombinacij iz tabele 6.4.....	187
Slika 6.52: Napake pri različnih kombinacijah oddajnikov in odmikov jakosti signalov	188
Slika 6.53: Razpon napake pri različnih kombinacijah oddajnikov v točki $T1$	189
Slika 6.54: Označevanje področij znotraj grafa - »škafca z brki«	189
Slika 6.55: Vpliv vsote razdalj med izbranimi oddajniki in položajem sprejemnika na mediano napak	190
Slika 6.56: Kumulativna porazdelitev verjetnosti napake pri različnih kombinacijah oddajnikov in odmikov jakosti signalov v točki $T1$	191
Slika 6.57: Možni položaji pri upoštevanju vseh kombinacij iz tabele 6.5.....	192
Slika 6.58: Razpon napake pri različnih kombinacijah oddajnikov v točki $T1$	193
Slika 6.59: Kumulativna porazdelitev verjetnosti napake pri različnih kombinacijah oddajnikov in odmikov jakosti signalov v točki $T1$	193
Slika 6.60: Možni položaji pri upoštevanju vseh kombinacij iz tabele 6.6.....	194
Slika 6.61: Razpon napake pri različnih kombinacijah oddajnikov v točki $T2$	195
Slika 6.62: Vpliv vsote razdalj med izbranimi oddajniki in položajem sprejemnika na mediano napak	196
Slika 6.63: Kumulativna porazdelitev verjetnosti napake pri različnih kombinacijah oddajnikov in odmikov jakosti signalov v točki $T2$	196
Slika 6.64: Zemljevid prstnih odtisov	198
Slika 6.65: Podatki, ki jih vsebuje posamezni prstni odtis.....	198
Slika 6.66: Prikaz širjenja Bluetooth signalov po prostoru na podlagi modelov	199
Slika 6.67: Prikaz širjenja bluetooth signalov po prostoru na podlagi modelov.....	200
Slika 6.68: Ocena položaja z upoštevanjem najbližjega sosedu	201
Slika 6.69: Ocena položaja z upoštevanjem K najbližjih sosedov	201
Slika 6.70: Ocena položaja z upoštevanjem uteženih K najbližjih sosedov	202
Slika 6.71: Koncept vp-drevesa (v seznamu elementov so z zeleno označene odločitvene točke in s sivo vsi ostali elementi).....	204

Slika 6.72: Območje, ki ga pokriva vozlišče (z zeleno označena odločitvena točka). Z modro označena vozlišča pripadajo levemu podrejenemu vozlišču, z oranžno označena vozlišča pa pripadajo desnemu podrejenemu vozlišču.	205
Slika 6.73: Področje okoli poizvedbenega elementa q , znotraj katerega se morajo nahajati najbližji sosede	206
Slika 6.74: Scenarij, pri katerem je poizvedbena točka q oddaljena od odločitvene točke v za več kot $(\mu + \tau)$	207
Slika 6.75: Scenarij, pri katerem je poizvedbena točka q oddaljena od odločitvene točke v za manj kot $(\mu - \tau)$	207
Slika 6.76: Scenarij, pri katerem je poizvedbena točka q oddaljena od odločitvene točke v za več kot $(\mu - \tau)$ in manj kot $(\mu + \tau)$	208
Slika 6.77: Položaji (določeni z uporabo vizualne odometrije in inercialnega navigacijskega sistema), v katerih so bile zajete meritve jakosti signalov RSSIM	210
Slika 6.78: Določanje položaja v prostoru pri uporabi meritev jakosti signalov iz treh različnih oddajnikov (dolžina vektorja RSSIM je 3) in upoštevanju 16 najbližjih sosedov ($K=16$)	210
Slika 6.79: Napake pri določanju položaja v primeru, ko so upoštevane meritve jakosti signalov iz treh različnih oddajnikov in je $K=16$	211
Slika 6.80: Določanje položaja v prostoru pri uporabi meritev jakosti signalov iz petih različnih oddajnikov (dolžina vektorja RSSIM je 5) in upoštevanju enega najbližjega sosedu	212
Slika 6.81: Napake (razdalje med izračunanim položajem in dejanskim položajem) pri določanju položaja v primeru, ko so upoštevane meritve jakosti signalov iz petih različnih oddajnikov in je $K=1$	212
Slika 6.82: Vpliv parametra K (število najbližjih sosedov) na natančnost lokalizacije	213
Slika 6.83: Določanje položaja v prostoru pri uporabi meritev jakosti signalov iz petih različnih oddajnikov in upoštevanju optimalnega števila najbližjih sosedov ($K=23$)	213
Slika 6.84: Napake (razdalje med izračunanim položajem in dejanskim položajem) pri določanju položaja v primeru, ko so upoštevane meritve jakosti signalov iz petih različnih oddajnikov in je $K=23$	214

Slika 6.85: Določanje položaja v prostoru pri uporabi meritev jakosti signalov iz petih različnih oddajnikov, upoštevanju optimalnega števila najbližjih sosedov ($K=23$) in uporabi nizkopasovnega filtra za glajenje poti.....	215
Slika 6.86: Napake pri določanju položaja v primeru, ko so upoštevane meritve jakosti signalov iz petih različnih oddajnikov, $K=23$ in je uporabljen nizkopasovni filter.	215
Slika 6.87: Kumulativna porazdelitev napake v primeru, ko je uporabljen nizkopasovni filter za glajenje poti (92 % časa je napaka manjša od 1 m).....	216
Slika 6.88: Prikaz širjenja Bluetooth signalov po prostoru na podlagi mehkih modelov	217
Slika 6.89: Prikaz širjenja Bluetooth signalov po prostoru na podlagi mehkih modelov	218
Slika 6.90: Določanje položaja v prostoru pri uporabi prstnih odtisov, ki so bili generirani na podlagi mehkih modelov (dolžina vektorjev meritev RSSIM je 5 in $K=23$)	219
Slika 6.91: Napake pri določanju položaja z uporabo prstnih odtisov, ki so bili generirani na podlagi mehkih modelov (dolžina vektorjev meritev RSSIM je 5 in $K=23$)	219
Slika 6.92: Določanje položaja v prostoru pri uporabi prstnih odtisov, ki so bili generirani na podlagi mehkih modelov (dolžina vektorjev meritev RSSIM je 5 in $K=23$) in upoštevanju nizkopasovnega filtra za glajenje poti.....	220
Slika 6.93: Napake pri določanju položaja z uporabo prstnih odtisov, ki so bili generirani na podlagi mehkih modelov (dolžina vektorjev meritev RSSIM je 5 in $K=23$). Za glajenje poti je bil uporabljen nizkopasovni filter.....	220
Slika 6.94: Primerjava kumulativnih porazdelitev verjetnosti napak pri določanju položaja z uporabo prstnih odtisov, ki so bili generirani z nelinearnimi modeli oz. mehki modeli	221
Slika 6.95: Primerjava kumulativnih porazdelitev verjetnosti napak pri določanju položaja z uporabo metode PSO oz. metode prstnih odtisov (v obeh primerih so bili uporabljeni mehki modeli in dodatno filtriranje z nizkopasovnim filtrom).....	222
Slika 6.96: Primerjava kumulativnih porazdelitev verjetnosti napak pri določanju položaja z uporabo metode prstnih odtisov. V prvem primeru (vijolična krivulja) je	

bila v postopku iskanja najbližjih sosedov uporabljena evklidska razdalja, v drugem primeru (zelena krivulja) pa utežena evklidska razdalja.	224
Slika 7.1: Fuzija vizualne odometrije, inercialnega navigacijskega sistema in Bluetooth lokalizacije.....	231
Slika 7.2: Izpopolnjena fuzija vizualne odometrije, inercialnega navigacijskega sistema in Bluetooth lokalizacije.....	234
Slika 7.3: Rezultati lokalizacije pri uporabi posameznih metod (BLE, VO in INS) ter fuzije vseh metod. Pešec je prehodil pot od štarta to cilja v smeri rumenih puščic, pri čemer je osrednji pravokotnik obhodil dvakrat.	236
Slika 7.4: Orientacija pešca med hojo po referenčni poti, ki je prikazana na sliki 7.3	237
Slika 7.5: Razdalja med položajem, pridobljenim z Bluetooth lokalizacijo in izhodom iz Kalmanovega filtra, ter stopnja zakasnitve in stanje dodatne predikcije	238
Slika 7.6: Položaji določeni z Bluetooth lokalizacijo (rdeče pike) in popravljeni položaji pri upoštevanju dodatni predikciji (modra krivulja).....	238
Slika 7.7: Fuzija po osi X.....	239
Slika 7.8: Fuzija po osi Y	240
Slika 7.9: Položaji, določeni z Bluetooth lokalizacijo ter fuzijo vizualne odometrije, inercialnega navigacijskega sistema ter Bluetooth lokalizacije	240

Seznam tabel

Tabela 6.1: Bluetooth razredi	116
Tabela 6.2: Možne izhodne moči BLE oddajnika in doseg signalov [184]	118
Tabela 6.3: Položaj Bluetooth oddajnikov v prostoru.....	129
Tabela 6.4: Odstopajoče jakosti signalov v točki $T1(0, 2)$	187
Tabela 6.5: Odstopajoče jakosti signalov v točki $T1(0, 2)$	191
Tabela 6.6: Odstopajoče jakosti signalov v točki $T2(0, 8)$	195

1 Uvod

Sposobnost določanja lastnega položaja v poljubnem okolju se je do neke mere razvila pri vseh živih bitjih na Zemlji. Z razvojem novih metod in algoritmov lokalizacije pa bi to sposobnost radi omogočili tudi robotom, avtonomnim vozilom in predvsem ljudem, pri katerih je ta sposobnost oslabljena, npr. zaradi slepote. Hiter razvoj mobilnih in komunikacijskih tehnologij je v zadnjih letih spodbudil številne raziskave na področju lokalizacije in navigacije v notranjem okolju. Pametni telefoni postajajo nepogrešljivi pripomočki sodobnega človeka tudi zaradi možnosti njihove uporabe za namen osebnega navigacijskega sistema. Medtem ko je lokalizacija v zunanjem okolju v večji meri rešen problem, pa za notranje okolje še vedno ni znano, kateri pristop in tehnologija bosta prevladala. Možnost določanja položaja v notranjem okolju je namreč še toliko bolj pomembna, saj človek tudi do 90 % svojega življenja preživi v zaprtih prostorih. Ker v notranjem okolju signali globalnih navigacijskih satelitskih sistemov (GNSS) niso na voljo, je potrebno izbrati alternativne senzorje in komunikacijske module pametnega telefona, ki omogočajo dovolj natančno lokalizacijo. Poleg izbranih senzorjev in komunikacijskih modulov je za natančnost delovanja lokalizacijskega sistema pomemben algoritem, ki na podlagi meritev s senzorjev in komunikacijskih modulov preračuna trenutni položaj v prostoru. Natančen lokalizacijski sistem, ki lahko deluje tudi v notranjem okolju, ima veliko uporabno vrednost, saj se lahko vgradi v osebni navigacijski sistem (ang. personal navigation system – PNS) za vodenje po nakupovalnih središčih, muzejih, letališčih, javnih ustanovah itd. Tak sistem je še posebej koristen za slepega človeka. Natančen lokalizacijski sistem pa je nepogrešljiv tudi pri avtonomnih vozilih, ki se gibljejo po zaprtem prostoru in pri tem izvajajo določene naloge. S tem razlogom je bilo izbrano znanstveno-raziskovalno področje, ki obsega lokalizacijo oseb ali avtonomnih mobilnih sistemov v notranjem okolju z uporabo sodobnega pametnega telefona

Android. Ta ima vgrajene številne senzorje: pospeškometer, žiroskop, magnetometer, barometer, kamero, ter komunikacijske module: WiFi, Bluetooth, NFC, LTE, ki posredno omogočajo lokalizacijo v notranjem okolju.

Med razvitimi lokalizacijskimi algoritmi, ki so že bili preizkušeni v praksi [1-5], večji potencial kažejo tisti, ki temeljijo na uporabi kamere, inercialnih senzorjev ali WiFi/Bluetooth modula. Ker je pri večini obstoječih lokalizacijskih algoritmov hkrati uporabljen le eden izmed pristopov (npr. samo pristop s kamero ali z inercialnimi senzorji), je natančnost določanja položaja slabša, kot bi lahko bila, če bi več senzorjev in komunikacijskih modulov združili v en algoritem. Zato smo naše raziskovalno delo usmerili v razvoj lokalizacijskega sistema, ki združuje različne pristope k lokalizaciji, in sicer z uporabo kamere, inercialnih senzorjev ter radijskih signalov.

Pri razvoju lokalizacijskih algoritmov se moramo zavedati zahteve, da se vsi algoritmi izvajajo v realnem času, saj so v nasprotnem primeru bolj ali manj neuporabni. Pri zahtevi, da se vsi algoritmi izvajajo na pametnem telefonu, pa je le te potrebno še dodatno prilagoditi in optimirati, ter s tem prilagoditi njihovo računsko zahtevnost strojni opremi, na kateri se izvajajo.

Pretekle študije [1-5] so pokazale, da je kamera tisti senzor pametnega telefona, ki v kombinaciji z ustreznim algoritmom omogoča največjo natančnost lokalizacije. Vendar se v praksi vizualna lokalizacija na pametnem telefonu še ni uveljavila predvsem zaradi računske zahtevnosti algoritmov za obdelavo slike. Večina preteklih študij vizualne lokalizacije je bilo povezanih z avtonomnimi vozili ter mobilnimi roboti, kateri so bili opremljeni z zmogljivimi procesorskimi sistemi. Prednost lokalizacije z uporabo kamere je ta, da ne potrebuje nobene dodatne infrastrukture znotraj zgradb in da je vsa potrebna strojna oprema relativno poceni. Pri uporabi kamere v splošnem obstaja več različnih pristopov, metod in algoritmov, s katerimi lahko določimo položaj ter prepotovano pot agenta (vozila, človeka, robota), na katerega je kamera pritrjena. Med uveljavljene metode sodijo struktura iz gibanja (SFM) [6, 7], Simultaneous Localization and Mapping (SLAM) [8] in vizualna odometrija (VO) [9]. V povezavi s pametnimi telefoni pa se uporablja tudi pristop, pri katerem se trenutno zajeta slika primerja z znanim 3D zemljevidom oz. oblakom točk (ang. image-to-map matching) [1, 3, 7, 10]. Za metode, ki temeljijo na pristopu SFM in SLAM velja, da so računsko zahtevne, hkrati pa porabijo tudi veliko pomnilnika,

saj se poleg ocene gibanja sproti gradi tudi 3D zemljevid oz. struktura okolja. Ker vizualna odometrija ocenjuje le gibanje kamere, lahko posledično deluje v realnem času tudi na računsko manj zmogljivi strojni opremi. Pojem vizualne odometrije je uveljavil Nister [9], ki je v svojem delu predstavil idejo o sledenju inkrementalnim premikom kamere s sprotnim izračunavanjem trigoriščnega tenzorja (ta predstavlja posplošitev fundamentalne matrike za tri poglede). Vizualna odometrija pogosto temelji na metodi optičnega toka [11], saj je na ta način algoritem računsko manj potraten, kot bi bil v primeru, če bi značilne točke detektirali na vsaki sliki, nato te značilne točke opisali ter iskali njihove pare na zaporednih slikah.

Drugi pristop k lokalizaciji z uporabo pametnega telefona je inercialni navigacijski sistem (ang. Inertial Navigation System – INS), ki temelji na meritvah inercialne merilne enote (IMU – ang. Inertial Measurement Unit). Ta je sestavljena iz pospeškometra, žiroskopa in magnetometra. INS preko tehnike navigacije na slepo (ang. dead reckoning) omogoča sledenje položaja pametnega telefona glede na poznano začetno točko z uporabo digitalnega števca korakov (pedometra) ter digitalnega kompasa [12]. Inercialni navigacijski sistem je mogoče uporabiti tudi na kolesnem robotu, pri čemer se namesto pedometra uporabi dvojna integracija pospeška za določanje prepotovane razdalje. Ta pristop je nekoliko bolj izpostavljen akumulaciji napake, vendar to lahko zmanjšamo z upoštevanjem druge lokalizacijske tehnike, kot je npr. vizualna ali kolesna odometrija.

Trenutno najbolj razširjeni pristop k notranji lokalizaciji s pametnim telefonom temelji na radijski lokalizaciji z uporabo WiFi ali Bluetooth omrežja [13]. Lokalizacija z uporabo radijskih signalov zahteva sprotno spremljanje različnih parametrov radijskih signalov (npr. RSS – ang. Received Signal Strength, ID oddajnika), ki potujejo med mobilno napravo in skupino baznih postaj [4]. Ker sodobni mobilni aparati omogočajo sprejem različnih radijskih signalov, so se temu primerno uveljavili različni pristopi k lokalizaciji, in sicer z uporabo mobilnega LTE omrežja, WiFi omrežja, Bluetooth tehnologije, FM signalov, NFC povezave itd. V zadnjem času se vse bolj uveljavlja Bluetooth LE (Low Energy) tehnologija [14], saj pri uporabi te tehnologije za namen lokalizacije v notranjem okolju pametni telefoni porabijo zelo malo energije ob relativno dobri natančnosti lokalizacije (študije [13] so pokazale, da je mogoče doseči natančnost do 1 m). Pri uporabi WiFi ali Bluetooth lokalizacijskega

pristopa pametni telefon meri jakosti signalov vseh dostopnih točk oz. Bluetooth oddajnikov, ki imajo različne MAC naslove. Na podlagi izmerjenih jakosti signalov je položaj v prostoru mogoče določiti na dva načina, in sicer z metodo, ki temelji na principu »prstnih odtisov« (ang. fingerprinting-based method), ali z upoštevanjem modelov jakosti signalov ter uporabo trilateracije [15]. Pri slednjem načinu je položaj uporabnika določen z modelom [16], ki pove, kakšna je jakost signala glede na oddaljenost od oddajnika (upoštevata se informacija o padanju jakosti signala z oddaljevanjem od oddajnika).

Z namenom, da bi odpravili pomanjkljivosti posameznih metod lokalizacije, so se številni avtorji [17-20] lotili fuzije (združevanja) različnih metod in algoritmov. Namreč, visoko natančnost in robustnost lokalizacije je možno doseči le s fuzijo različnih lokalizacijskih metod, ki se med seboj ustrezno dopolnjujejo [17]. Zato se tudi inercialni navigacijski sistem redkokdaj uporablja samostojno za namen določanja položaja v prostoru. Pogosto se uporablja v kombinaciji z vizualno lokalizacijo ali lokalizacijo na podlagi radijskih signalov [21].

V nadaljevanju (drugo poglavje) je najprej predstavljen pregled področja lokalizacije v notranjem okolju. Na tem mestu so opisani obstoječi lokalizacijski algoritmi in metode, ki temeljijo na uporabi meritev, zajetih s senzorji in komunikacijskimi moduli sodobnih pametnih telefonov. Nato so v ločenih poglavjih predstavljeni trije različni pristopi k lokalizaciji s pametnim telefonom, za katere so bili razviti novi algoritmi in rešitve.

V tretjem poglavju disertacije obravnavamo prvi pristop k lokalizaciji, in sicer vizualno lokalizacijo, ki temelji na uporabi kamere pametnega telefona. V tem poglavju je opisan algoritem monokularne vizualne odometrije, ki omogoča sledenje premikov kamere in posledično določanje relativnega položaja človeka ali avtonomnega vozila, na katerega je pritrjena kamera oz. pametni telefon. Vizualno odometrijo sestavljajo štirje algoritmi, in sicer algoritem za kalibracijo kamere, sledilnik značilnih točk – KLT (Kanade-Lucas-Tomasi), algoritem za določanje togega premika ter algoritem RANSAC.

V četrtem poglavju obravnavamo drugi pristop k lokalizaciji, in sicer inercialni navigacijski sistem, ki temelji na uporabi vgrajenih inercialnih senzorjev (pospeškometra, žiroskopa in magnetometra). Inercialni navigacijski sistem omogoča

določanje trenutnega položaja uporabnika pametnega telefona glede na začetno točko preko tehnike PDR (ang. Pedestrian Dead Reckoning). Za namen štetja korakov smo razvili števec korakov oz. pedometer, ki temelji na merjenju vertikalnega pospeška. Za določanje absolutne orientacije pametnega telefona smo implementirali digitalni kompas, ki za svoje delovanje uporablja vse tri inercialne senzorje (pospeškometer, žiroskop in magnetometer).

V petem poglavju disertacije je opisana fuzija vizualne odometrije in inercialnega navigacijskega sistema, katere namen je odpraviti pomanjkljivosti posameznih metod in doseči natančen in robusten relativni lokalizacijski sistem.

V šestem poglavju disertacije obravnavamo tretji pristop k lokalizaciji, in sicer radijsko lokalizacijo, ki temelji na merjenju jakosti Bluetooth signalov. V tem poglavju je opisanih več različnih metod, ki se lahko uporabljajo pri radijski lokalizaciji (z WiFi ali Bluetooth signali). Za vsako metodo so bili izvedeni eksperimenti, katerih rezultati kažejo prednosti in slabosti posameznih metod.

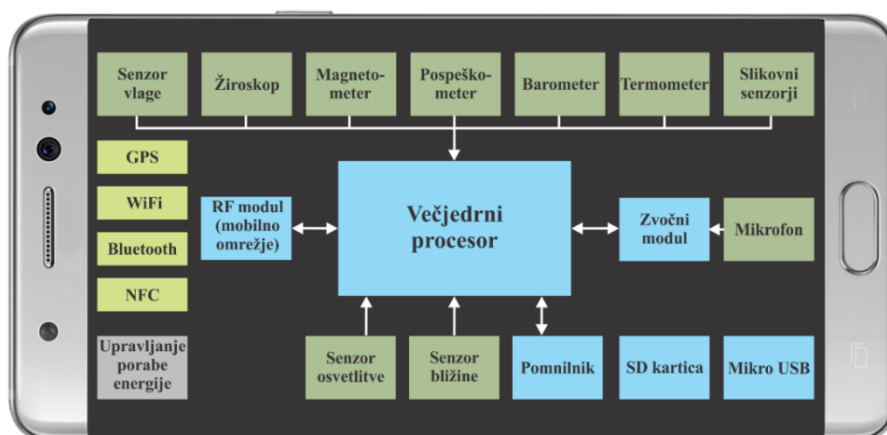
V sedmem poglavju disertacije je predstavljena fuzija vseh treh lokalizacijskih pristopov, in sicer vizualne odometrije, inercialnega navigacijskega sistema in radijske lokalizacije, ki temelji na uporabi Bluetooth signalov. Za namen fuzije je bil uporabljen Kalmanov filter, pri katerem sta bila v predikcijski del vključena vizualna lokalizacija in inercialni navigacijski sistem, v korekcijski del pa globalna lokalizacija na podlagi radijskih signalov. Inercialni navigacijski sistem je predvsem pomemben za odpravljanje napak, do katerih lahko pride pri vizualni odometriji (npr. napačno določen zasuk zaradi zameglitve slike). Vizualna odometrija je pomembna pri določanju natančnih translacij med inkrementalnimi premiki, Bluetooth lokalizacija pa ima ključno vlogo pri določanju globalnega položaja v prostoru in omejevanju napake, ki se s časom povečuje pri vizualni odometriji in inercialnemu navigacijskemu sistemu.

V zaključku, ki predstavlja zadnje poglavje disertacije, so podane sklepne misli in omenjene možnosti za nadaljnji razvoj in izboljšave lokalizacijskega sistema.

2 Pregled obstoječih metod za lokalizacijo v notranjem okolju

Ko se nahajamo znotraj velikih zgradb, ki so nam še nepoznane, se zna zgoditi, da bomo izgubili veliko časa za iskanje željenega cilja (osebe, trgovine, pisarne itd.), ali da bomo celo povsem izgubili orientacijo. Globalni navigacijski satelitski sistemi – GNSS (GPS, GLONASS, BeiDou-2, Galileo) [22], ki se uporabljajo za navigacijo v zunanjem okolju, so blizu zgradb in znotraj njih nezanesljivi (ter pogosto tudi nedostopni) zaradi odbojev in absorpcije signalov, kar pomeni, da moramo poiskati alternativni pristop k lokalizaciji znotraj zgradb. Hitra rast števila uporabnikov pametnih telefonov [23] vpliva tudi na širjenje in razvoj osebnih lokalizacijskih sistemov, ki imajo velik pomen, saj večino časa preživimo znotraj zgradb.

Sodobni pametni telefoni so nadvse primerni za namen lokalizacije, saj so opremljeni s številnimi senzorji (pospeškometer, žiroskop, magnetometer, barometer), dvema kamerama, WiFi modulom, Bluetooth modulom, NFC tehnologijo itd., kot je prikazano na sliki 2.1, ter tudi njihova računska moč, se neprestano povečuje.



Slika 2.1: Komponente sodobnih pametnih telefonov

V pametne telefone se vgrajujejo že tudi deset jedrni procesorji, kot je npr. MediaTek MT6797. Zato so sodobne mobilne naprave vse bolj primerne tudi za poganjanje računsko zahtevnih algoritmov.

Velika podjetja, kot so Google [24] in Microsoft [25], ki se ukvarjajo z gradnjo digitalnih zemljevidov, so v večjih mestih že dodali tlorise zgradb (slika 2.2) v svoje zemljevide. Google od leta 2012 omogoča (brezplačno), da uporabniki lahko tudi sami naložijo tlorise zgradb, ki jih pri Googlu nato obdelajo in spremenijo v standardno obliko ter vgradijo v njihov zemljevid. Na področju gradnje zemljevidov za notranje okolje se je uveljavilo tudi mlado »start-up« podjetje Micello [26], ki pa svojih storitev ne ponuja brezplačno. Druga podjetja, kot so Apple, Nokia [27], Meridian [28] in IndoorAtlas [29], prav tako razvijajo svoje aplikacije za mobilne naprave (z operacijskim sistemom Android in iOS), ki omogočajo navigacijo in druge servise povezane z lokalizacijo v notranjem okolju. Primarni namen teh mobilnih aplikacij je vodenje ljudi po npr. trgovskih centrih, muzejih, letališčih, fakultetah, parkirnih hišah itd., vendar pa se informacije o položaju uporabnika vse pogosteje uporabljajo tudi za ciljno (»targetirano«) oglaševanje. Lokalizacijski sistem na pametnem telefonu je lahko v veliko pomoč tudi slepim, slabovidnim, ljudem z Alzheimerjevo boleznijo in gasilcem.



Slika 2.2: Google zemljevid muzeja »Smithsonian National Air and Space Museum« v Washingtonu na Android telefonu [24]

2.1 Lokalizacija z uporabo kamere

Vizualna lokalizacija v notranjem okolju, ki temelji na uporabi ene ali več kamer, postaja predmet številnih študij, diskusij in resnih projektov, predvsem zaradi velikega

potenciala, saj trenutno omogoča enega izmed najbolj natančnih načinov lokalizacije v zaprtem prostoru. Le natančen in učinkovit način sledenja gibanja kamere pa omogoča implementacijo osebne navigacije po zaprtem prostoru [30] in drugih aplikacij, ki za svoje delovanje zahtevajo poznavanje trenutne lokacije [31]. Vizualna lokalizacija se pogosto uporablja tudi pri vodenju avtonomnih mobilnih sistemov (raziskovalni roboti, kvadrokopterji) [32, 33, 34], tako v zunanjem kot v notranjem okolju. Prednost vizualne lokalizacije je v tem, da za svoje delovanje ne potrebuje nobene dodatne infrastrukture pri uporabi naravnih značilk, kot so SIFT (ang. Scale Invariant Feature Transform) [35], SURF (ang. Speeded-Up Feature Transform) [36], BRIEF (ang. Binary Robust Independent Elementary Features) [37] itd. Binarni opisniki značilnih točk so začeli pridobivati na pomembnosti, ko je bil prvič predstavljen BRIEF opisnik značilnih točk [38]. Avtor tega dela Calonder je dokazal, da ta opisnik vrača za majhne zasuke in razlike v velikosti slik enako dobre rezultate, kot dobro uveljavljeni opisnik SURF [39]. Velika prednost BRIEF-a v primerjavi s SURF-om je ta, da BRIEF kar 40 krat hitreje najde značilne točke in hkrati porabi veliko manj spomina. Poleg tega je računanje razdalje med BRIEF značilnimi točkami veliko hitrejše, saj je uporabljena Hammingova razdalja, namesto evklidske razdalje. Ti dobri rezultati so spodbudili nadaljnji razvoj binarnih značilnih točk. Tako je nastal dopolnjeni binarni deskriptor ORB (Oriented Binary Robust Independent Elementary Features), ki invarianten na rotacije. Rotacija je v tem primeru ocenjena z uporabo centroidne metode [40]. Drugi deskriptor, ki je prav tako nastal na podlagi BRIEF deskriptorja je BRISK (ang. Binary Robust Invariant Scalable Keypoints) deskriptor [41]. Ta se v primerjavi z deskriptorjema BRIEF in ORB razlikuje po tem, da položaji vzorčenja slike niso več naključno določeni. Poleg tega, da upošteva orientacijo značilnih točk, BRISK vključuje tudi teorijo skalirnega prostora, ki služi za prilagajanje vzorca na maksimalno vrednost v skalirnem prostoru. Zato je BRISK invarianten na rotacije in skaliranje. Zadnji izmed binarnih opisnikov, ki so nastali na podlagi BRIEF metode je FREAK [42]. Ta deskriptor ima izvedeno primerjavo intenzitete vzorcev na tak način, da je podobna razporeditvi na človeški mrežnici. Ta deskriptor je invarianten na rotacije, skaliranje in spremembo svetlosti. Glede na ovrednotenje v [43] FREAK v vseh pogledih prekaša opisnike, kot so BRIEF, ORB, BRISK, SURF in SIFT. Vsi zgoraj omenjeni opisniki značilnih točk potrebujejo

detektor značilnih točk, ki pove algoritmu katero točko mora opisati. Ker je pri binarnih značilnih točkah glavni poudarek na hitrosti, so ti deskriptorji pogosto uporabljeni v kombinaciji s FAST (ang. Accelerated Segment Test) [44] detektorjem značilnih točk. Kot pove že njegovo ime, deluje zelo hitro pri tem, ko išče vogale na sliki. Deskriptorja BRISK in FREAK sta pogosto kombinirana z izboljšano verzijo detektorja FAST, ki se imenuje AGAST. Ta detektor prav tako kot FAST na sliki išče vogale.

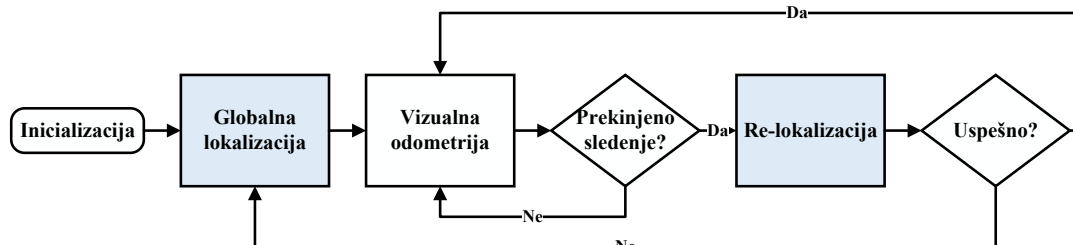
Da bi ugotovili, kako zelo sta si podobni dve značilni točki, moramo med njima izračunati razdaljo. Med binarnimi značilnimi točkami lahko razdaljo izračunamo s Hammingovo razdaljo. Slednjo izračunamo tako, da pogledamo razlike v bitih med dvema nizoma (uporabimo XOR), ter nato še preštejemo število enic v rezultatu.

Pri lokalizaciji z uporabo kamere je iskanje značilnih točk skoraj v vseh primerih prisotno. Detekcija in opis značilnih točk v realnem času pogosto predstavlja težavo, še posebej na procesorsko manj zmogljivih napravah. Avtorji Saipullah in drugi [45] so v svojem delu opisali rezultate testiranja več različni detektorjev značilnih točk na mobilni napravi z OS Android. Pri detekciji objekta v realnem času so bili preizkušeni naslednji algoritmi: SIFT, SURF, CenSurE (ang. Center Surrounded External), GFTT (ang. Good Features To Track), MSER (ang. Maximally-Stable External Region Extractor), ORB in FAST. Rezultati so pokazali, da algoritem FAST vrača najboljše rezultate pri detektiranju objektov, glede na hitrost delovanja. Testirali so hitrost na posamezen slikovni okvir, število detektiranih značilnih točk, ponovljivost in stopnjo napake pri različnih osvetlitvah in zornih kotih.

Ker so vgradni sistemi, kamor sodijo tudi mobilni telefoni, procesorsko vse zmogljivejši, še največjo težavo povzročajo baterije, ki se pri uporabi kamere hitro izpraznijo. Zato je pri uporabi vizualne lokalizacije na pametnem telefonu pomembna enostavna oz. optimizirana implementacija algoritma, ki lahko teče (kot servis) v ozadju mobilne aplikacije. Tako ostane tudi več procesorske moči za drugi del aplikacije, ki vključuje uporabniški vmesnik (npr. za obogateno resničnost, ang. augmented reality [31]).

V znanstvenih študijah se pojavlja več različnih pristopov k vizualni lokalizaciji, pri čemer je najbolj pogost ta, ki vključuje predhodno vizualno kartiranje notranjosti

zgradb. Tak sistem so razvili tudi na TUM Munich v okviru projekta Navvis, ki ga je opisal Straub [1, 46] in je shematsko prikazan na sliki 2.3.



Slika 2.3: Shema lokalizacijskega sistema, ki so ga razvili na TUM [47]

Blokec z imenom vizualna odometrija (ang. Visual Odometry), ki je del sheme na sliki 2.3, se izvaja na mobilni napravi in inkrementalno ocenjuje pozicijo naprave. Odometrija je inicializirana preko globalnega lokalizacijskega algoritma (razloženo v nadaljevanju), ki teče na serverju. Sistem vključuje tudi postopek ponovne lokalizacije (re-lokalizacije) v primeru, da pride do napake pri sledenju. Re-lokalizacija (do katere pride v primeru hitrih premikov ali v primeru, ko se pojavi ovira, ali so slabi svetlobni pogoji) se izvede z uporabo trenutno zgrajenega 3D zemljevida (oblaka točk) z metodo SLAM (PTAM [2, 5]) na mobilni napravi. V primeru, da je tudi re-lokalizacija neuspešna, potem se celoten sistem ponovno inicializira glede na globalno lokalizacijo, ki se izvede na serverju (primerjava zajete slike z množico slik v bazi po metodi »Content-based image retrieval« - CBIR, ki obsega iskanje na podlagi vsebine slike [48]).

Pojem vizualne odometrije je uveljavil Nister [9], ki je v svojem delu predstavil idejo o sledenju inkrementalnim premikom kamere s sprotnim izračunavanjem trigoriščnega tenzorja. Uporabnost monokularne vizualne odometrije se pokaže pri določanju inkrementalnih premikov robota, brezpilotnega letalnika [49] ali druge mobilne naprave opremljene z eno kamero. Zato pride v poštev tudi pri določanju pozicije in orientacije pametnih mobilnih naprav [2, 50]. Pri vizualni odometriji se običajno uporabljata metodi optičnega toka [11] in strukture iz gibanja [6]. Prav tako lahko z metodo SLAM (ang. Simultaneous localization and mapping) implicitno ocenjujemo gibanje kamere [2, 5]. SLAM lahko hkrati ocenjuje in optimizira trajektorijo, ter gradi 3D strukturo okolja skozi čas. Pri večini metod vizualne

odometrije je detekcija in opis značilnih točk [35, 36, 37] ključen korak. Obstajajo pa tudi metode, ki tega koraka ne zahtevajo, saj je ravninska rotacija in translacija med zaporednima slikovnim okvirjema lahko ocenjena tudi s fazno korelacijo.

Lokalizacijski sistemi, ki temeljijo na vizualni odometriji, omogočajo zelo natančno pozicioniranje, kot tudi oceno orientacije [2, 18, 46, 51]. Vendar ker vizualna odometrija sodi med relativne lokalizacijske sisteme, je pogosto kombinirana z drugim globalnim lokalizacijskim sistemom [48], ki omogoča inicializacijo vizualne odometrije, tj. ocenjevanje začetne lokacije (mobilne) naprave v obsežnem notranjem okolju [46]. Globalni vizualni lokalizacijski sistemi, ki temeljijo npr. na metodi CBIR [48] so računsko potratni in se zato izvajajo na serverju. Ker gre pri tem za prenos podatkov (o detektiranih značilnih točkah) na server, je pomembna hitrost prenosa in velikost podatkov (v tem primeru imajo prednost binarne značilne točke [37]).

Za oceno 3D pozicije z monokularno vizualno odometrijo sta potrebna vsaj dva zajema slik pri različnih kotih. Algoritem mora torej iskati ujemanja med dvema slikovnim okvirjema, ki prikazujeta več istih značilnih točk v 3D prostoru. Za določanje 3D položajev posameznih značilnih točk v prostoru je pomembna metoda triangulacije (ali rekonstrukcije), za zmanjšanje re-projekcijskih pogreškov, ki nastanejo pri tem, pa metoda »bundle adjustment« [52]. Slednja pogosto temelji na uporabi robustnega Levenberg-Marquardt algoritma, ki omogoča nelinearno optimizacijo po metodi najmanjših kvadratov.

Hilsenbeck in drugi [53] so v svojem delu predstavili sistem vizualne odometrije s poudarkom na konsistentnosti in robustnosti za večje časovne intervale. Pri implementaciji tega lokalizacijskega sistema na mobilni napravi je jedro algoritma predstavljala metoda PTAM [2, 5]. V študiji so se osredotočili na iskanje skalirnega faktorja tako pri zgrajenem zemljevidu, kot pri trajektoriji (določeni z metodo PTAM), saj le tega ne poznamo pri uporabi ene same kamere. Problema so se lotili z uporabo inercialnih senzorjev in razširjenega Kalmanovega filtra (ang. extended Kalman filter (EKF)). Iskali so lokalne geometrijske lastnosti zgrajenega zemljevida z naključnimi projekcijami. Iz širine hodnika so ocenili spremembo skale zaradi napake pri sledenju in tako določili popravek z EKF. Časovno potratnost algoritma so skrajšali s paralelno implementacijo *ponovne vzpostavitve sledenja* in *re-inicializacije*. Tako sledenje teče nemoteno. V delu Strasdat in drugi [54] pa so se avtorji osredotočili na omejevanje

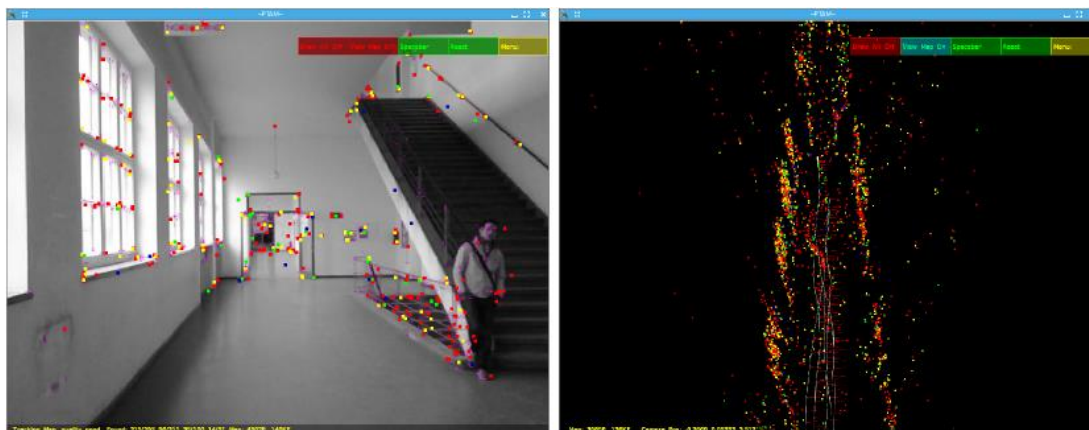
lezenja skalirnega faktorja pri uporabi vizualne odometrije skozi daljše časovno obdobje. Lezenje skalirnega faktorja so uspešno omejili z metodo zapiranja zanke (poti). Pri uporabi monokularne vizualne odometrije je lahko skalirni faktor določen tudi glede na znane mere nekega objekta, ki ga sistem prepozna.

V delu avtorjev Davison in drugi [55, 56] je bilo prvič pokazano, da je monokularni vizualni SLAM (ang. Simultaneous Localization and Mapping) prav tako uporaben, kot stereo vizualni SLAM za namen lokalizacije v neznanem okolju. Davison in drugi so predlagali algoritem SLAM na osnovi razširjenega Kalmanovega filtra za oceno vektorja stanj, ki vključuje položaj kamere in pozicijo značilnih točk v 3D prostoru. Za uspešno sledenje premikom kamere, EKF pri vsakem novem okvirju s ponovnim izračunom popravi položaj. Ker so EKF posodobitve računsko potratne, je bil ta sistem omejen na okoli 100 značilnih točk znotraj 3D zemljevida.

Sim in drugi so v svojem delu [57] uporabili filter delcev za reševanje vizualnega SLAM-a. Njihov sistem temelji na veliki podatkovni bazi SIFT opisnikov, z uporabo katere so ocenili položaj robota na 2D zemljevidu. Vendar njihov sistem ni bil primeren za delovanje v realnem času, saj je potreboval kar 11,9 sekunde na slikovni okvir. Teh omejitev sta se znebila avtorja Eade in Drummond [8] z uporabo novega načina iskanja značilnih točk v vsakem novem slikovnem okvirju, namesto iskanja podobnih prehodno pridobljenih značilnih točk shranjenih v veliko podatkovno bazo. Njihov sistem temelji na principu metode FastSLAM, ki uporablja filter delcev in lahko teče v realnem času pri uporabi do 250 značilnih točk.

Z metodo PTAM (ang. Parallel Tracking and Mapping) sta avtorja Klein in Murray [2] uvedla idejo o ločitvi niti za sledenje položaja kamere in niti, ki skrbi za oceno položaja značilnih točk znotraj 3D zemljevida. To omogoča, da nit za kartiranje lahko izkoristi čas procesorja, ki ni potreben za sledenje položaja kamere oz. pri več jedrnih procesorjih se lahko obe niti izvajata sočasno. PTAM algoritem, ki sta ga razvila Klein in Murray je učinkovit pristop k implementaciji monokularnega SLAM-a. Ta omogoča simultano določanje položaja kamere in gradnjo 3D strukture okolja. Algoritem PTAM uporablja FAST detektor [44] za iskanje značilnih točk v vsakem novem slikovnem okvirju. Ključne točke, ki imajo visoko oceno glede na Shi-Thomasi-jev kriterij, se smatrajo za dobre in se zato le te uporabljajo za sledenje in gradnjo 3D zemljevida (slika 2.4).

Avtorja Klein in Murray [5] ter tudi Porzi in drugi [50] so v svojem znanstvenem delu pokazali, da je algoritem PTAM mogoče implementirati tudi na mobilni napravi (z dodatno optimizacijo [58]), vendar je njegova uporaba zelo omejena (zaradi omejitve velikosti zemljevida). PTAM ni primeren za določanje položajev kamere znotraj velikih prostorov (mogoče le v kombinaciji z drugim globalnim lokalizacijskim sistemom), lahko pa se uporablja npr. za namen navidezne resničnosti, pri čemer se kamera giba le znotraj majhnega omejenega prostora (kjer lahko predpostavimo, da je ključen slikovni okvir blizu trenutnega). Tudi novejšie različice PTAM-a, kot so SVO [49], LSD-SLAM [59] in ORB SLAM [60], ki veljajo za »*state of the art*« algoritme monokularnega vizualnega SLAM-a, se ne morejo uporabljati kot samostojen lokalizacijski sistem (če nimamo predhodno zgrajenega 3D zemljevida oz. oblaka točk), saj se pri gibanju po večjem (neznanim) prostoru pogosto prekine sledenje položaju kamere, kar zahteva re-lokalizacijo z drugim lokalizacijskim sistemom. Poleg tega so algoritmi SVO, LSD-SLAM in ORB SLAM računsko izjemno zahtevni in se v kombinaciji s pametnim telefonom lahko uporabljajo le na majhnem omejenem prostoru (velikost 3D zemljevida oz. oblaka točk je omejena).



Slika 2.4: PTAM (levo: sledenje značilnih točk, desno: oblak značilnih točk) [50]

Straub in drugi [1, 46] so v svojem delu za re-lokalizacijo oz. ponovno inicializacijo vizualne odometrije (po prekinitvi sledenja npr. zaradi majhnega števila značilnih točk) uporabljali modificirano metodo PTAM, ki temelji na binarnih značilnih točkah. Re-lokalizacija znotraj omejenega prostora je lahko v grobem izvedena na dva načina: glede na bazo slikovnih okvirjev ali glede na bazo značilnih točk, katerih položaj je znan. Prva metoda zahteva, da se shranjujejo posamezne slike iz množice slikovnih

okvirjev (kot so ključni slikovni okvirji pri metodi PTAM) in njihov položaj. Druga metoda zahteva določanje 3D položaja značilnih točk in pomnjenje opisa značilnih točk. Eno izmed prvih del, ki opisuje metodo re-lokalizacije in temelji na bazi slikovnih okvirjih, so predstavili Dellaert in drugi [61]. Ti so kot prvi uporabili Monte Carlo lokalizacijski algoritem (MCL). Njihov sistem uporablja vizualen zemljevid stropa za lokalizacijo robota v prostoru. Metodo, ki temelji na bazi slikovnih okvirjev (z znanim položajem) sta predlagala tudi Klein in Murray [2]. Njena slabost je, da se re-lokalizacija lahko izvede samo, če je bil določen prostor že prej poskeniran s kamero in da so bili ključni slikovni okvirji tega prostora že dodani v bazo podatkov. Podobno so tudi Reitmayr in drugi [62] uporabili slikovne okvirje (z znanim položajem znotraj zemljevida prostora) za ponovno vzpostavitev sledenja po detekciji napake.

Eno izmed prvih del, kjer je opisana re-lokalizacija (po zaznani napaki pri sledenju) na podlagi baze značilnih točk, je delo avtorjev Chekhlov in drugi [63], ki so za robustno primerjavo značilnih točk uporabili vizualno indeksiranje značilnih točk slike (značilne točke so opisane s histogrami prostorskih gradientov v njeni okolici). S primerjavo opisanih značilnih točk, je bil določen položaj kamere, pri čemer je bil uporabljen tri-točkovni (P3P) algoritem [63] v kombinaciji z RANSAC-om [64].

Arth in drugi [3] so bili eni izmed prvih, ki so se lotili razvoja vizualnega lokalizacijskega sistema za mobilno napravo. Zunanji prostor so razdelili na potencialno vidne dele, katerih poznane značilne točke se naložijo iz serverja na pametni telefon. Uporabili so spremenjene SURF značilne točke, ki so jih primerjali z uporabo najbližjega sosedu. Za oceno položaja je bil uporabljen tri-točkovni algoritem, in sicer kot del algoritma RANSAC [64]. Oceno položaja so izboljšali z uporabo robustnega Gaus-Newtonovega algoritma. Pokazali so, da je uporaba principa pri katerem se primerja trenutno zajeta slika z znanim 3D zemljevidom (»image-to-map matching«), najhitrejša. Arth in drugi [3, 10] so v študijah opisali tudi postopek rekonstrukcije ključnih točk v 3D oblaku za zelo razsežne prostore in lokalizacijo z mobilno napravo v takem prostoru. Lokalizacijo so izvedli tako, da so z mobilno napravo zajeli panoramsko sliko (uporabili so metodo FAST za detekcijo vogalov), na kateri so določili značilne točke. Te so se nato primerjale s točkami v bazi (v oblaku točk). S tem pristopom so dosegli natančnost enega metra, s tem da so za boljši rezultat uporabili še pospeškometer in kompas mobilne naprave. Uporabili so tudi metodo vidnih

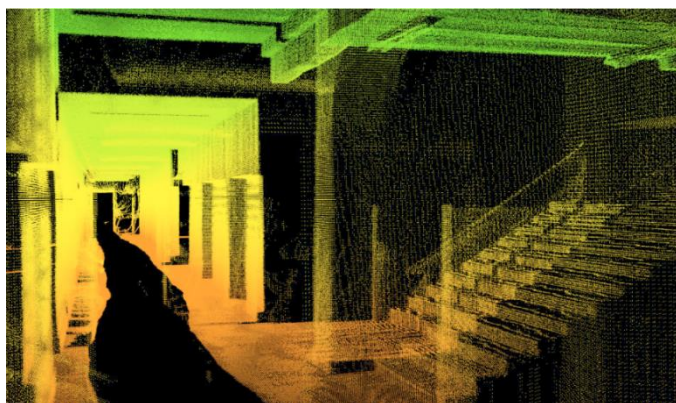
točk (ang. »Potentially Visible Sets«) [65], saj se tako zmanjša število potencialnih točk za primerjavo. Primer oblaka točk je prikazan na sliki 2.5. Zelo podoben pristop, kot so ga predlagali Arth in drugi za lokalizacijo mobilne naprave, so Straub [1, 46] in drugi uporabili za re-lokalizacijo v primeru prekinitve sledenja do katerega pride, npr. zaradi zameglitve slike, hitre spremembe osvetlitve ali ovire. Pri re-lokalizaciji algoritem pridobi opis (z deskriptorjem BRIEF [38]) v ključnih točkah slike, ki so bile izbrane z enakim detektorjem značilnih točk, kot je bil uporabljen pri vizualni odometriji. To zagotavlja, da so ključne točke potencialno ponovno detektirane in nato primerjane s pravilnimi 3D značilnimi točkami znotraj oblaka.



Slika 2.5: Rekonstrukcija točk v oblaku

Arth in drugi [10] so v svojem delu uporabili metodo robustne inkrementalne strukture iz gibanja za 3D redko rekonstrukcijo [7] iz slik (ang. sparse image reconstruction). Postopek rekonstrukcije se izvede v treh korakih. Najprej se zgradi epipolarni graf, ki ima v svojih vozliščih slike, na robovih pa se ujemanje preverja z epipolarno geometrijo (geometrija stereo vida). Postopek primerjanja slik se izvaja z uporabo metode »vreča besed« (ang. bag of words) [10]. V drugem koraku je ta graf pretvorjen v graf rekonstruiranih trojčkov slik (ang. triplet based bottom-up reconstruction). V tretjem koraku so robovi pretvorjenega grafa inkrementalno združeni v končno rekonstrukcijo. Pri strukturi iz gibanja se za ugotavljanje ujemanja značilnih točk med zaporednimi slikovnimi okvirji uporablja sledilnik značilnih točk [66]. Zelo razširjen je sledilnik Kanade-Lucas-Tomasi (KLT) [67].

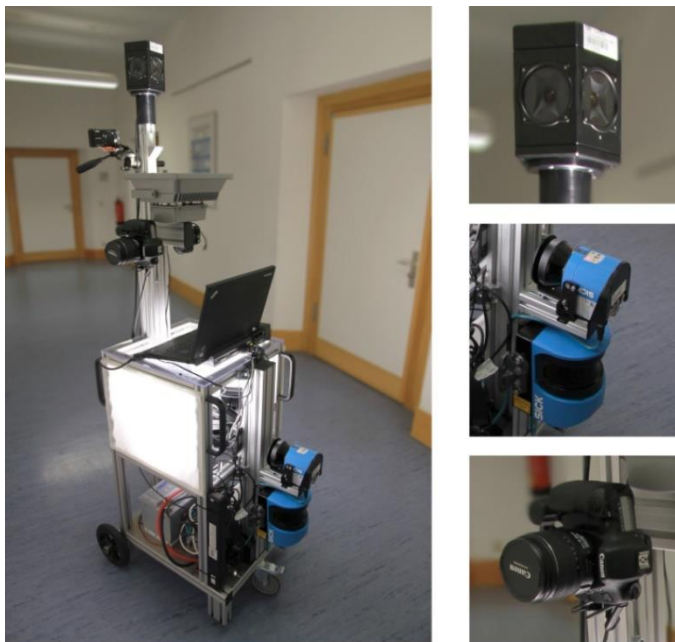
Za bolj natančno izgradnjo zemljevida 3D točk (slika 2.6), ki se skupaj z zajetimi ključnimi slikami uporablja pri globalni vizualni lokalizaciji z metodo CBIR, so raziskovalci na TUM v okviru projekta Navvis [48] sestavili voziček prikazan na sliki 2.7. Ta je opremljen z vodoravno nameščenim laserskim skenerjem, ki pokriva kot 190° in enkoderji za kolesno odometrijo, kar omogoča izgradnjo 2D zemljevida za samo lokalizacijo vozička. Voziček je opremljen še s panoramsko kamero Ladybug3 (ima 6 CCD kamer) in dvema DSLR fotoaparatom Canon EOS 7D in vertikalno nameščenim 270° laserskim skenerjem. Slednje komponente vozička omogočajo zajem številnih slik iz katerih je mogoče (z uporabo fotogrametrije) zgraditi gost oblak točk (ang. dense point cloud), ki vsebujejo tudi opis 2D značilnih točk.



Slika 2.6: 3D zemljevid točk v oblaku

Najbolj razširjen pristop k lokalizaciji s kamero je uporaba predhodno zgrajenega 3D zemljevida ter iskanje in primerjava značilnih točk. Globalna lokalizacija znotraj baze 10k slik in 100M značilnih točk zahteva drugačne pristope k lokalizaciji kot npr. re-lokalizacija znotraj zemljevida, zgrajenega z metodo SLAM. Tehnike, ki se uporabljajo za re-lokalizacijo so primerne za baze velikosti do 1k slik ali 100k značilnih točk. Pri veliki bazi je problem, da deskriptorji ne opišejo značilnih točk dovolj natančno, da bi lahko kar neposredno primerjali značilne točke na trenutni sliki s točkami v bazi. Za določanje omejitev, katere značilne točke je smiselno medsebojno primerjati, se uporablja tehnika CBIR (ang. Content-based Image Retrieval) [68], ki je primerna za lokalizacijo pri uporabi velike baze slik oz. značilnih točk. Pri tej metodi se primerja množica točk iz trenutne slike z množico točk v bazi. Predstavitev slike z

množico značilnih točk (»vrečo značilnk«) imenujemo »Bag of Features« (BoF) model [69].



Slika 2.7: Voziček za gradnjo 3D zemljevida točk

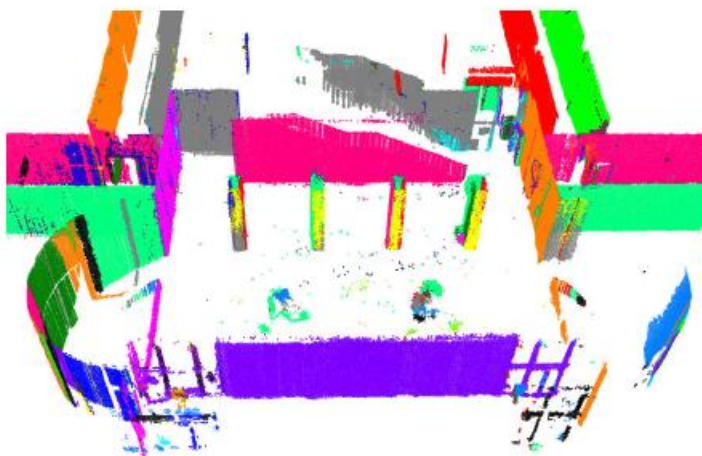
Prvi lokalizacijski sistem, ki temelji na binarnih značilnih točkah in tehniki CBIR-u je bil predlagan s strani avtorjev Galvez-Lopez in Tardos [70]. Vizualne besede so bile pridobljene z diskretizacijo binarnega prostora BRIEF značilnih točk, pri čemer je bil uporabljeno hierarhično besedno drevo. Vsak nivo drevesa je dosežen z deljenjem značilnih točk in uporabo »k-medoids« algoritma (za rojenje).

Za iskanje ustrezne slike je pri tehniki CBIR potrebna kvantizacija značilnih točk v t.i. vizualne besede (ang. »visual words«). Za namen uporabe binarnih značilnih točk so Straub in drugi [1, 46] predlagali algoritem k -tih povprečij rojenja (k-means clustering). Pri CBIR je kvantizator kBM (»k-Binary Means«) uporabljen z namenom, da vsem pridobljenim značilnim točkam iz N slik baze dodeli ID-je vizualnih besed. kBM kvantizator ima M vizualnih besed, saj vse značilne točke baznih slik združi v M rojev.

Pomanjkljivost lokalizacijskega sistema, ki temelji na tehniki CBIR je ta, da je lokalizacija lahko le toliko natančna, kolikor pogosto so zajete slike v bazi. To omejitev so odpravili Huitl in drugi [71] s sintezo (slika 2.8) pogledov notranjega okolja v pozicijah, iz katerih slike niso bile zajete. Projekcija ravnine v slikovni prostor

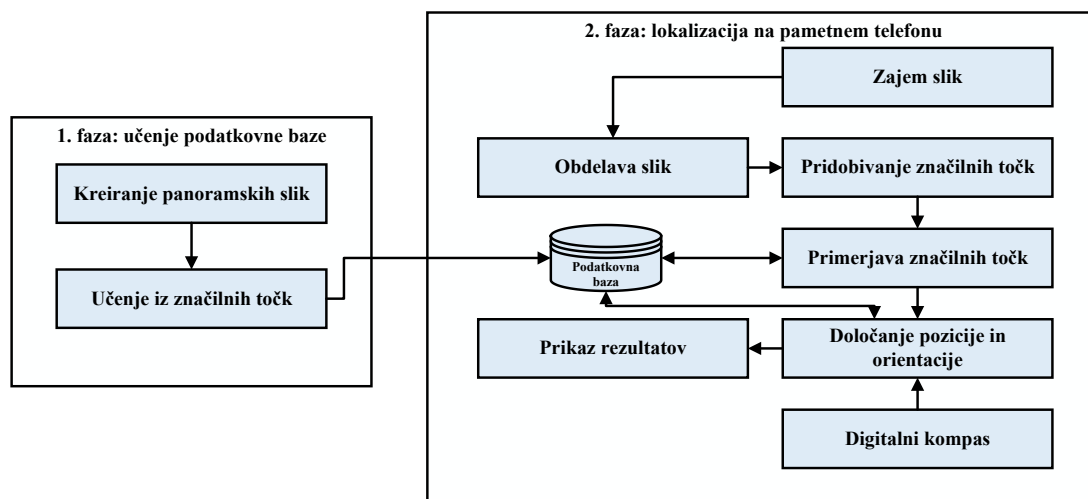
dveh kamer omogoča, da se z uporabo homografije ustvari nove virtualne poglede kamere iz obstoječih slik, kar znatno izboljša natančnost lokalizacije.

Avtorja Li in Wang [66] sta v svojem delu opisala globalno lokalizacijsko tehniko, ki temelji na predhodno zgrajenem 3D zemljevidu in ima veliko skupnega s sistemom Navvis, ki so ga opisali Huitl [71], Straub in drugi [46]. Za detekcijo značilnih točk sta uporabila SIFT značilne točke, ter za njihovo robustno primerjavo algoritem RANSAC. Predlagala sta tudi uporabo križne korelacije pri preverjanju modela homografije.



Slika 2.8: Segmentacija oblaka točk na ravnine, ki so osnova za projektivne transformacije

Avtorja Nguyen in Lee [72] sta v svojem delu opisala sistem za navigacijo v zaprtem prostoru, ki sta ga preizkusila na pametnem telefonu. Sistem obsega več faz, ki so prikazane na sliki 2.9. V fazi učenja je v podatkovno bazo shranjenih več panoramskih slik in njihov položaj v prostoru, na podlagi katerih se med izvajanjem navigacije določi pozicija in orientacija mobilne naprave. Panoramske slike so bile zajete v centrih 3×3 m velikih kvadratov. Za primerjavo slik sta avtorja uporabila modificirano metodo HIP (ang. Histogrammed Intensity Patch) [73]. S to metodo so bile določene tudi HIP značilne točke za vse shranjene panoramske slike. Orientacija je bila določena tako, da so bile panoramske slike razdeljene na štiri dele, kjer vsak pokriva 90° . Glede na to s katerim izmed štirih delov panoramske slike se trenutno zajeta slika najbolj ujema, je določena orientacija mobilne naprave.



Slika 2.9: Lokalizacijski sistem, ki sta ga predlagala Nguyen in Lee [72]

Pinto in drugi so v svojem delu [74] predstavili lokalizacijo v zaprtem prostoru, glede na umetne značilke, in sicer LED diode različnih barv. Sistem omogoča robustno, natančno in zanesljivo oceno položaja. Sistem so preizkusili na robotu, ki je bil opremljen z eno kamero. Zaslonka kamere je bila sinhronizirana z LED diodami preko IR vmesnika. Z uporabo EKF-SLAM-a so zgradili zemljevid, ki določa medsebojni položaj umetnih značilnih točk. Podoben sistem, ki temelji na umetnih značilnih točkah sta predstavila tudi Zachariah in Jansson [75]. Sistem določi začetni položaj z uporabo homografije in zasuk glede na več opaženih značilnih točk. Značilne točke imajo obliko mreže velikosti $q \times q$ in uporabljajo *BCH* kodiranje. S tehniko upravljanja so določili binarno sliko. Nato so razpoznane točke razdeljene v linije in štirikotnike.

Uporaba umetnih značilk, kot so LED diode, QR kode, GRATF itd. zelo poenostavi razvoj algoritma za lokalizacijo z uporabo kamere. Vendar v večini primerov za notranje okolje predstavljajo estetsko motnjo in za lastnika stavbe tudi dodaten strošek.

Wang in drugi [76] so predlagali sistem sledenja oseb v zaprtem prostoru, ki temelji na statičnih kamerah in uporabi inercialnih senzorjev v mobilni napravi. Kamere ne pokrivajo celotnega področja, saj je njihova naloga, da le občasno zaznajo lokacijo določene osebe in tako omogočijo resetiranje akumulirane napake, do katere pride pri uporabi inercialnih senzorjev na pametnem telefonu. Ta način izvedbe lokalizacije ima to prednost, da je mobilni aparat lahko v žepu, kar omogoča bolj natančno določanje

prehojene razdalje (saj pospeškometer ni izpostavljen takšnemu tresenju, kot če napravo držimo v roki).

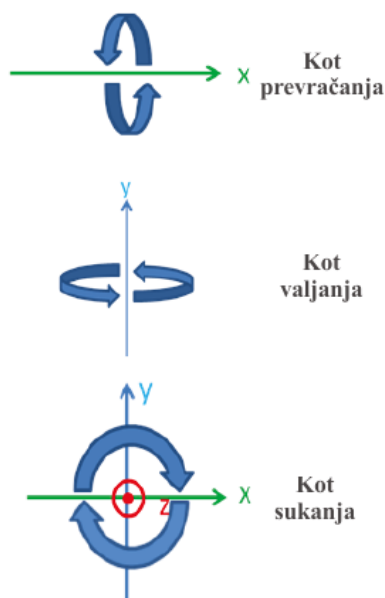
2.2 Lokalizacija z uporabo inercialnih senzorjev

Inercialni navigacijski sistem (ang. Inertial Navigation System – INS) je navigacijska tehnika, ki temelji na meritvah inercialne merilne enote (ang. inertial measurement unit – IMU) [77]. Ta je običajno sestavljena iz 3-osnega pospeškometra (ang. accelerometer), 3-osnega žiroskopa (ang. gyroscope) in 3-osnega magnetometra (ang. magnetometer), kar pomeni, da imamo IMU z devetimi prostostnimi stopnjami. Inercialni navigacijski sistem preko »navigacije na slepo« (ang. dead reckoning oz. deduced reckoning, kar pomeni, da gre za sklepanje ocene položaja) omogoča določanje relativnega položaja mobilnega sistema ali človeka [78] glede na poznano začetno točko in hitrost gibanja. Pri uporabi digitalnega kompasa [20, 79], ki je sestavljen iz vseh treh inercialnih senzorjev, lahko INS določa tudi absolutno orientacijo naprave ali človeka. Prednost inercialnega navigacijskega sistema je ta, da za svoje delovanje ne potrebuje nobene dodatne infrastrukture.

Pospeškometer, ki je del IMU se lahko uporablja za detekcijo gibanja, merjenje pospeškov gibajočega se telesa, detekcijo korakov pri hoji, določanje orientacije telesa glede na smer gravitacijskega pospeška itd [77, 79, 80]. V mobilnih napravah so kot del MEMS sistema (ang. Microelectromechanical systems) vgrajeni trije ortogonalni pospeškometri, ki merijo pospešek v smeri vseh treh osi X , Y in Z . MEMS sistem meri pospešek z zaznavanjem, kako znana masa pritiska na del senzorja, ko sila deluje na to maso. Pospeškometer, ki meri pospešek ortogonalno glede na zemeljsko površino v stanju mirovanja kaže pospešek $\sim 9,81 \text{ m/s}^2$, kar je posledica zemeljske gravitacije. Če želimo dobiti samo linearni pospešek zaradi gibanja telesa v smeri X , Y ali Z , moramo gravitacijski odmik odšteti, pri čemer pa moramo zelo natančno poznati orientacijo telesa. Ker med gibanjem orientacija ne more biti povsem natančno določena, se del gravitacijskega pospeška prišteje k linearnemu pospešku (zaradi gibanja), kar povzroči velike napake pri določanju hitrosti ali dolžine poti (pri integriranju se namreč napaka le akumulira). Pri triosnem pospeškometru se zna zgoditi, da tudi osi niso povsem ortogonalne in da na meritve vpliva pristranskost (ang. bias) senzorja, kar pri dvojnem integriranju povzroči še hitrejšo akumulacijo napake

(kvadratično naraščanje napake). Pristranskost (ang. bias) senzorja se da oceniti z meritvami povprečja skozi daljše časovno obdobje v katerem pospeškometer ni podvržen nobenim pospeškom [20].

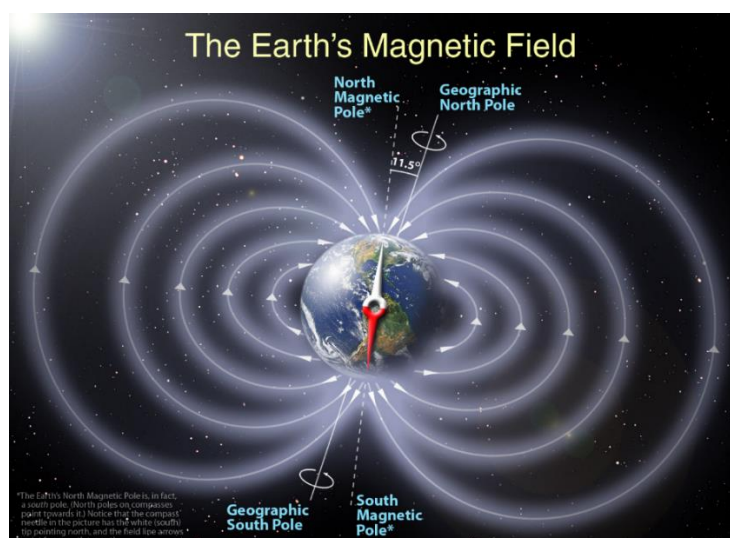
Žiroskop je inercialni senzor, ki je del IMU in meri kotne hitrosti po vseh treh oseh (slika 2.10) iz katerih lahko dobimo relativne zasuke: kot valjanja (»roll«, okoli osi y), kot prevračanja (»pitch«, okoli osi x) in kot sukanja (»yaw«, okoli osi z) [79]. Žiroskop temelji na principu ohranitve vrtilne količine (ang. angular momentum). V mobilnih napravah so vgrajeni tri osni žiroskopi, ki so del MEMS sistema. Ta vsebuje vibrirajoče elemente za merjenje Coriolis-ovega efekta. Žiroskopi so prav tako kot pospeškometri povrženi pristranskosti, kar pomeni, da meritve odstopajo od prave vrednosti. Odstopanje se da oceniti z merjenjem povprečne vrednosti izhoda, ko žiroskop ni izpostavljen nobenim rotacijam. Konstantno odstopanje povzroča, da pri integriranju kotne hitrosti dobimo napako pri kotu, ki narašča linearno s časom [79]. Druga napaka, ki se pojavlja pri žiroskopih pa je »kalibracijska napaka« [20]. Ta se pojavlja zaradi napak pri skalirnih parametrih, netočne poravnave žiroskopa in nelinearnosti. Napako je mogoče zaznati le med rotiranjem žiroskopa pri čemer prinaša dodatno lezenje kotov zaradi akumulacije napak pri integriranju.



Slika 2.10: Žiroskop meri relativne zasuke mobilne naprave po vseh treh oseh

Magnetometer je merilna naprava za merjenje jakosti in/ali smeri magnetnega polja, ki obdaja merilni inštrument [77]. V mobilnih napravah je magnetometer del IMU-ja

in služi (kot del digitalnega kompasa) za določanje absolutne orientacije mobilne naprave. V mobilnih napravah je magnetometer sestavljen iz treh komponent, ki merijo magnetno polje po vseh treh oseh koordinatnega sistema. Pri poznavanju magnetnega polja Zemlje (slika 2.11) lahko z uporabo magnetometra implementiramo digitalni kompas. Slednji nam vrne smer severnega (dejansko je to južni magnetni pol) magnetnega pola, ki je npr. v Ljubljani za 3° premaknjen glede na geografski severni pol. Ta pojav se imenuje deklinacija, pri čemer je stopnja zasuka odvisna od tega, kje na Zemlji se nahajamo. Omeniti pa velja še inklinacijo, ki pove za koliko stopinj je od horizontalne ravnine odmaknjen rezultirajoči vektor magnetnega polja. Inklinacija se tako, kot deklinacija spreminja glede na to, kje na Zemlji se nahajamo.



Slika 2.11: Magnetno polje Zemlje [81]

Pri merjenju jakosti in smeri magnetnega polja prihaja do napak, če se v bližini nahajajo večji železni predmeti ali elektronske naprave, ki spreminjajo magnetno polje. Dodatna napaka se pojavi, če napravo z vgrajenim magnetometrom rotiramo medtem, ko merimo magnetno polje. Pri implementaciji digitalnega kompasa je potrebno poleg magnetometra uporabiti vsaj še pospeškometer, ki pove kako je telefon orientiran glede na zemeljsko površje (glede na smer gravitacijskega pospeška), saj na ta način lahko odpravimo napako (ang. tilt compensation, napaka je odvisna od velikosti inklinacije) v primeru, da je magnetometer nagnjen.

Vsi trije omenjeni inercialni senzorji imajo svoje prednosti in slabosti. Da bi slabosti kar se da odpravili, se senzorji skoraj vedno uporabljajo skupaj. Cilj fuzije

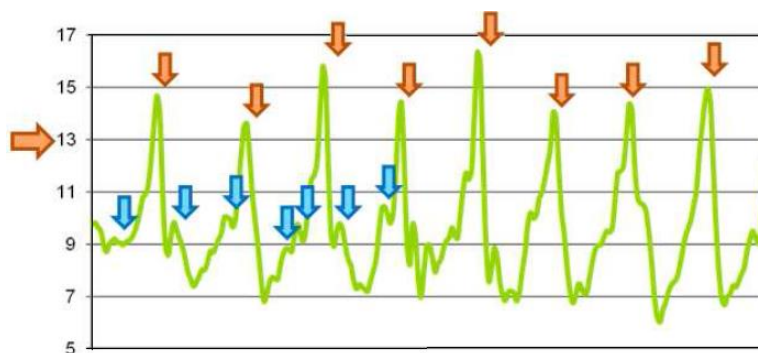
inercialnih senzorjev je združiti informacije iz različnih senzorjev, z namenom, da se odpravi napake (bias, šum, motnje) in doseže boljše rezultate, kot v primeru, če se senzori uporabljajo ločeno [82]. Izraz boljše pomeni bolj natančno, dovršeno in zanesljivo. V sodobnih pametnih telefonih, ki so opremljeni z OS Android 4.0 ali novejši je vgrajenih več t.i. sintetičnih (virtualnih) senzorjev, kot npr. senzor linearnega pospeška, senzor gravitacije in rotacijski vektor [83]. Rotacijski vektor je sintetični senzor, ki združuje informacije (z razširjenim Kalmanovim filtrom) iz pospeškometra, žiroskopa in magnetometra. Pospeškometer in magnetometer se uporabljata za odpravljanje lezenja žiroskopa, žiroskop pa služi za odpravljanje težav pri določanju orientacije in linernih pospeškov. Žiroskop namreč ni občutljiv na gravitacijo. Žiroskop je v pomoč tudi magnetometru, na katerega vplivajo motnje v magnetnem polju. Pospeškometer ima vlogo tudi pri kompenzaciji naklona (ang. tilt compensation), ki je potrebna pri digitalnem kompasu. Za fuzijo inercialnih senzorjev se najpogosteje uporablja Kalmanov filter (ali ena izmed njegovih izvedb) [79] ali tudi komplementarni filter (ang. Complementary filter), ki ga je v svojem delu, za namen fuzije pospeškometra, žiroskopa in magnetometra, predlagal Lawitzki [84].

Za doseg boljših rezultatov je fuzija inercialnih senzorjev potrebna tudi pri navigaciji na slepo (dead reckoning), kjer se ocenjuje položaj mobilne naprave glede na predhodni položaj, ocenjeno hitrost in smer gibanja [79]. Na mobilni napravi je mogoče oceniti hitrost gibanja z uporabo pospeškometra in enojnim integriranjem njegovih meritev (linearnega pospeška). Ta pristop k oceni hitrosti je relativno slab, saj se napaka (pri oceni hitrosti) linearno povečuje s časom [79]. Drugi pristop, ki ga lahko uporabimo za oceno hitrosti, vključuje merjenje časa in detekcijo korakov uporabnika (dobimo dolžino poti). Orientacijo uporabnika pametnega telefona je mogoče določiti z uporabo digitalnega kompasa, ki temelji na fuziji pospeškometra, žiroskopa in magnetometra [78, 79].

Pedometer je naprava, ki z analizo meritev pospeškometra šteje korake osebe [79, 85]. Pedometer se vedno več uporablja tudi v športu za oceno prehojene ali pretečene razdalje. Zato proizvajalci športne opreme že vgrajujejo pedometre tudi v podplat čevljev in zapestnice.

Na sliki 2.12 vidimo potek izhodnega signala iz pospeškometra, ki meri pospešek v smeri pravokotno na zemeljsko površje tekom hoje. Z vsakim korakom, ko pešec

udari s peto ob tla, se pojavi znaten skok signala (koraki so označeni z rjavimi puščicami, ki so usmerjene navzdol).



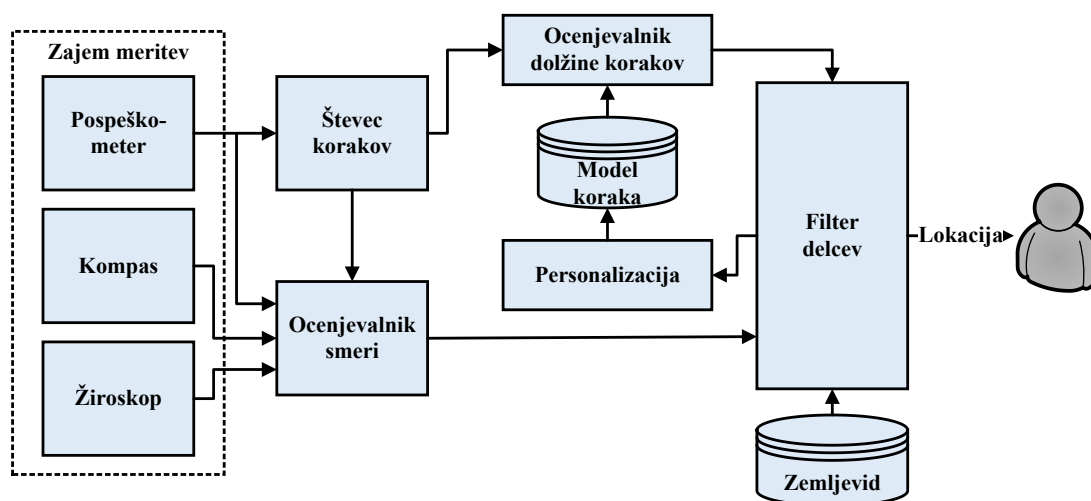
Slika 2.12: Detekcija korakov in gibanja [79]

Korake je mogoče preprosto detektirati tako, da določimo neko mejno vrednost, ki jo mora signal preseči. Lahko pa poiščemo tudi ekstreme signala in določimo, kako velika mora biti razlika med sosednjima ekstremoma (med lokalnim maksimumom in minimumom), da prištejemo nov korak. Oblika signala (iz pospeškometra) je odvisna od tega, kje se pedometer oz. mobilna naprava nahaja (v roki, v žepu) in kakšna je hitrost hoje [79]. Če napravo držimo v roki, se na signalu opazijo manjši skoki zaradi tresljajev, ki lahko vplivajo na zanesljivost detekcije korakov.

Za oceno prehojene razdalje moramo poznati še dolžino koraka, ki jo lahko nastavimo na fiksno vrednost (glede na velikost človeka) ali pa jo izračunamo z izbranim modelom (npr. na podlagi razlike med maksimumi in minimumi pospeška) [86]. Li in drugi [12] so pri implementaciji inercialnega lokalizacijskega sistema uporabili filter delcev, pedometer in digitalni kompas, ki so medsebojno povezani (slika 2.13). Naloga filtra delcev je, da združuje informacije detektorja korakov (pedometer) in informacije o orientaciji (digitalni kompas) ter upošteva omejitve, ki so določene z zemljevidom prostora.

Da lokalizacija (oseb) z uporabo inercialnih senzorjev ne rabi vedno temeljiti le na pedometru in digitalnemu kompasu, so dokazali Gozick in drugi [87], ki so v svojem delu opisali sistem lokalizacije na osnovi merjenja jakosti magnetnega polja. Z uporabo mobilne naprave, ki ima vgrajen magnetometer so posneli jakosti magnetnega polja v 2000 točkah. Ugotovili so, da se magnetno polje znotraj stavbe, kljub številnim anomalijam, manj spreminja kot v zunanjem okolju. Gozick in drugi so lokalizacijo

izvedli z iskanjem korelacije med podatki v bazi in trenutnim potekom jakosti magnetnega polja (upoštevanih je bilo več zaporednih meritev jakosti magnetnega polja). Lokalizacija se lahko izvede tudi po metodah, ki se uporabljajo pri lokalizaciji z Bluetooth ali WiFi signali [87], tj. trenutno izmerjeni jakosti magnetnega polja se poišče najbližjo vrednost iz baze podatkov (metoda najbližjega sosesa). Ker vse meritve znotraj baze pripadajo določenim položajem v prostoru, se na ta način lahko izračuna trenutni položaj. Blankenbach in drugi [88] pa so pri svoji implementaciji lokalizacije preizkusili posebne oddajnike, s katerimi so ustvarili navidezno statično magnetno polje. Na ta način so izboljšali natančnost pozicioniranja na nekaj centimetrov.



Slika 2.13: Inercialni lokalizacijski sistem [12]

2.3 Lokalizacija z uporabo radijskih signalov

Pri lokalizaciji z uporabo radijskih signalov je ključna faza postopka merjenje različnih parametrov radijskih signalov (npr. jakost signala, MAC naslov oddajnika, frekvenca oddajanja paketkov), ki potujejo med mobilno napravo in skupino baznih postaj [4]. Z uporabo radijskih signalov je lokalizacijski sistem v osnovi mogoče implementirati na dva načina. V prvem načinu mobilna naprava sama izvede lokalizacijo (izračuna svoj položaj v prostoru) z upoštevanjem signalov, ki jih oddajajo bazne postaje (ta pristop je uporabljen tudi pri GNSS sistemih). V drugem možnem načinu sta vlogi bazne in mobilne postaje zamenjani, kar pomeni, da bazne postaje analizirajo signale, ki jih

oddajajo mobilne naprave. Bazna postaja sprejete signale obdela in sporoči trenutni položaj mobilne naprave.

Ker sodobni mobilni aparati omogočajo sprejem različnih radijskih signalov, so se temu primerno uveljavili različni pristopi k lokalizaciji, tako z uporabo mobilnega omrežja (GSM, LTE) [89], WiFi omrežja [90], Bluetooth tehnologije [91], FM signalov [92], NFC povezave [93] itd.

S uporabo mobilnega omrežja (GSM, LTE) je na pametnih telefonih mogoče izračunati trenutni položaj z natančnostjo od 50 do 250 m. Lokacija mobilne naprave se določi glede na znan položaj baznih postaj (oddajnikov), ki periodično sporočajo svojo lokacijo [94, 95]. Bolj natančen način radijske lokalizacije omogočata WiFi in Bluetooth omrežja [90]. Na mobilnih napravah (z OS Android) je mogoče prebrati naslednje parametre WiFi omrežja: RSS (ang. Received signal strength), SSID (ang. Service Set Identifier), BSSID (ang. Basic Service Set Identification) in SNR (ang. Signal to Noise Ratio) oz. Bluetooth omrežja: RSSI (ang. Received Signal Strength Indication) in ID oddajnika. Čeprav ima večina javnih zgradb že postavljena WiFi omrežja, v zadnjem času predvsem Bluetooth tehnologija doživlja velik razmah na področju lokalizacije v notranjem okolju [90]. Strm vzpon Bluetooth tehnologije se je pričel s prihodom standarda Bluetooth 4.0 oz. Bluetooth Low Energy (BLE), ki porabi zelo malo energije in omogoča enako ali še boljšo natančnost lokalizacije, kot WiFi tehnologija. Bluetooth tehnologija, ki omogoča komunikacijo na razdalji do 100 m (odvisno od razreda, BLE signali na odprtem sežejo 70 m stran od oddajnika), je robustna, poceni, porabi malo energije, integrirana vezja pa imajo zelo majhne dimenzije (nekaj mm). Glavni pobudnik za uveljavitev Bluetooth tehnologije na področju lokalizacije v notranjem okolju je finsko podjetje Nokia [96], ki vodi tudi največji projekt na tem področju.

Lokacijo mobilne naprave je mogoče preprosto določiti z merjenjem jakosti radijskih signalov (padec jakosti je sorazmeren oddaljenosti od oddajnika), ki jih oddajajo bazne postaje in uporabo metode trilateracije [97]. Za oceno razdalje od oddajnika do sprejemnika je namesto merjenja jakosti signalov možno uporabiti tudi tehnike, ki temeljijo na merjenju časa, npr. algoritem razlike časov prihoda (ang. Time Difference of Arrival – TDOA) [98]. Nekateri Bluetooth oddajniki omogočajo tudi merjenje kota signala do sprejemnika in s tem uporabo triangulacije [99, 100, 101].

Najboljša natančnost lokalizacije je v tem primeru okoli 1 m, sicer pa zelo zavisi od dejstva, kako gosto so postavljeni oddajniki (kar praktično velja za vse tehnike radijske lokalizacije). Če želimo uporabiti triangulacijo pri določanju položaja uporabnika, potrebujemo posebne dostopne točke, ki imajo usmerjene antene, saj le tako lahko merimo fazni zamik in s tem kote sprejetih signalov (AOA – angle of arrival) [94, 102]. Kljub temu, da triangulacija omogoča večjo natančnost lokalizacije kot trilateracija, se ni uveljavila na področju lokalizacije v zaprtem prostoru, predvsem zaradi potrebe po nestandardni strojni opreми.

Vse pogosteje je v sodobnih mobilnih napravah vgrajena tudi tehnologija NFC (ang. Near Field Communication), ki jo je prav tako mogoče uporabiti za lokalizacijo v zaprtem prostoru. NFC deluje na kratki razdalji, in sicer do 10 cm. V tem primeru je potrebno po prostoru razporediti NFC nalepke (NFC tag), ki vsebujejo informacijo o lokaciji. Ko se z mobilno napravo dovolj približamo določeni NFC nalepki, ta sporoči našo lokacijo v stavbi [103] (na mobilni napravi se prikaže zemljevid na katerem je označen naš trenutni položaj). Slabost te metode je v tem, da zahteva namestitve NFC nalepk po celotni stavbi, kar lahko predstavljajo estetsko motnjo, ter tudi to, da mora uporabnik iskati NFC nalepke, da izve svoj položaj.

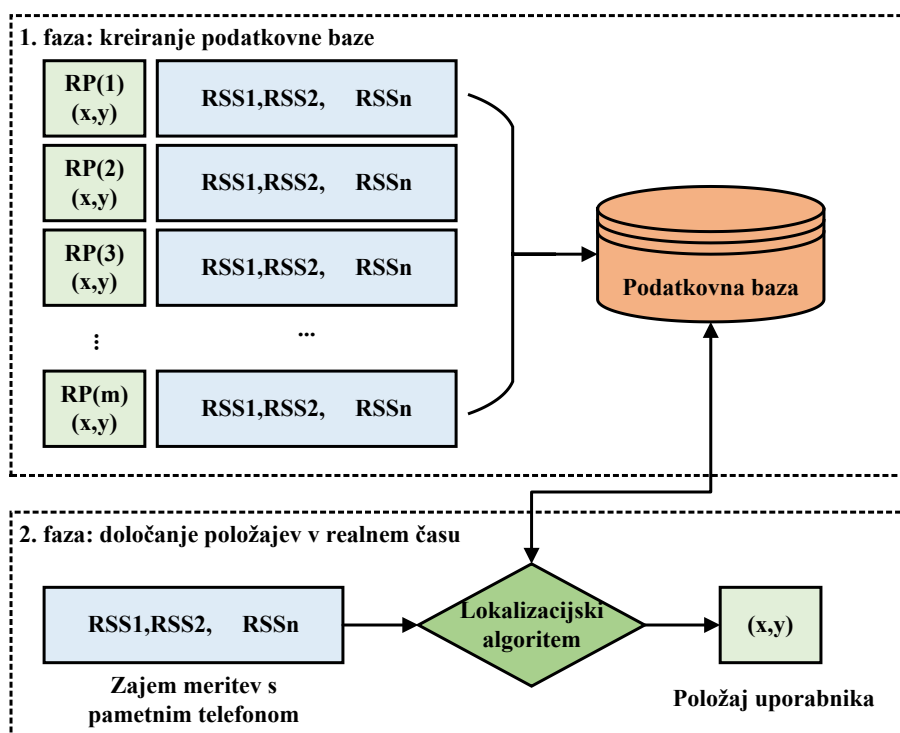
Z uveljavitvijo standarda IEEE 802.11 (ali Wireless Local Area Network – WLAN ali WiFi) so se začele pojavljati ideje o lokalizaciji v zaprtem prostoru z upoštevanjem informacij o jakosti WiFi signalov – RSS v različnih točkah prostora. Prednost tega pristopa je predvsem v tem, da v stavbi ne potrebujemo nameščati dodatne strojne opreme. Namreč v vse več zgradbah je prisotna infrastruktura brezžičnega WiFi omrežja [15, 104-107] in vse več ljudi že uporablja pametne telefone in tablice, ki omogočajo dostop do brezžičnega omrežja.

Na področju lokalizacije so se uveljavili trije osnovni pristopi, ki temeljijo na analizi jakosti signalov brezžičnega omrežja WiFi (ali Bluetooth): najmočnejša bazna postaja, izgradnja modelov jakosti signalov (in trilateracija) in metode, ki temeljijo na principu »prstnih odtisov« (ang. Fingerprint-based localization). Prvi pristop je najbolj preprost, saj se položaj uporabnika določi glede na to, iz katerega oddajnika njegova naprava sprejema najmočnejši signal, pri čemer morajo biti položaji oddajnikov znani. Prednost tega pristopa je računska nezahtevnost, pomanjkljivost pa nizka natančnost, ki je omejena na število oddajnikov (gostoto) [97].

Pri drugem pristopu, ki vključuje izgradnjo modelov jakosti signalov (za vse oddajnike), je položaj uporabnika mobilne naprave določen z uporabo modelov, ki povedo kakšne so razdalje do oddajnikov glede na izmerjene jakosti signalov (upoštevajo se informacija o padanju jakosti (ang. RSS path loss) signala z oddaljevanjem od dostopne točke oz. oddajnika). Za vsak oddajnik dobimo krožnico na kateri se lahko nahaja uporabnik. Z geometrijsko tehniko – trilateracijo [15] se nato določi le eno točko, v kateri naj bi se nahajal uporabnik. Ta pristop k lokalizaciji je relativno učinkovit in enostaven za implementacijo, le natančnost ni zelo visoka, saj na meritve jakosti signalov vplivajo odboji, prisoten šum in interference (posledično se jakost signala nenehno spreminja) zaradi prisotnosti ovir, kot so vrata, stene, strop, ljudje. Poleg odbojev pa se del energije tudi absorbira in spremeni v toploto, kar povzroči padec jakosti signalov. Zato je potrebno modele jakosti signala načrtovati tako, da predvidevajo spremembe v prostoru. Širjenje WiFi signalov in tudi drugih radijskih signalov se da opisati s t.i. LDPL (ang. log distance path loss) modelom [16]. V tem primeru gre za preprosto modeliranje širjenja signalov, pri čemer se ne upoštevajo vplivni faktorji kot so: časovne spremembe, prisotnost oseb in drugih ovir v prostoru ter dinamično spreminjanje WiFi pokritosti. Nurminen in drugi [108] so predlagali modeliranje širjenja radijskih signalov z uporabo metode Metropolis-Hastings (MH), ki temelji na pristopu Monte Carlo (Markov chain Monte Carlo). Kot drugo možnost pa so avtorji predlagali uporabo Gauss-Newtonove metode, ki je računsko manj zahtevna. Tudi relativen zasuk uporabnika glede dostopno točko vpliva na sprejeti signal, saj tekočina v človeškem telesu absorbira del energije signalov. Zato so številni avtorji pri modeliranju upoštevali tudi orientacijo osebe glede na dostopno točko [109, 110], oddaljenost mobilne naprave od človeškega telesa, debelino telesa in druge vplivne faktorje.

Pristop s »prstnimi odtisi« je napram pristopoma najmočnejše bazne postaje in izgradnje modelov jakosti signalov, veliko bolj enostaven za implementacijo ter hkrati tudi manj občutljiv na šume brezžičnih signalov [111]. Posledično lahko s to metodo dosežemo tudi večjo natančnost pozicioniranja v prostoru [19]. Sistem, ki temelji na uporabi »prstnih odtisov« je običajno sestavljen iz dveh faz, in sicer faze učenja (»offline«) in faze določanja položaja (»online«). V fazi učenja je cilj zgraditi podatkovno bazo, ki vsebuje meritve jakosti signalov (iz vseh oddajnikov) za vse

referenčne položaje v zgradbi [109, 110]. V fazi ugotavljanja položaja se trenutno izmerjene jakosti primerjajo z meritvami v bazi in glede na to s katero referenčno točko se trenutni vektor (meritev jakosti signalov) najbolj ujema (npr. glede na metodo KMLN – »K most likely neighbors« [112]), se določi trenutni položaj. Na sliki 2.14 vidimo shematski prikaz postopka (WiFi) lokalizacije s prstnimi odtisi, kot so ga predstavili Bahl in drugi [113] (sistem RADAR). Zgornji del prikazuje prvo fazo – meritve RSS v vseh referenčnih točkah (RP) in izgradnjo podatkovne baze, spodnji del pa prikazuje drugo fazo, ki vključuje meritev trenutnih jakosti signalov, primerjavo s podatki v bazi in končno določitev položaja. Jakost signala se meri v enotah dBm (razmerje moči v decibelih dB na en mW).



Slika 2.14: Shematski prikaz pristopa s prstnimi odtisi [113]

Pri WiFi lokalizaciji s prstnimi odtisi se uporabljata dva algoritma za primerjavo trenutno izmerjenih jakosti signalov in podatkov v bazi (na sliki 2.14: »lokalizacijski algoritem«). Prvi algoritem vključuje izračun evklidske razdalje (lahko bi bila tudi katera druga mera, kot npr. Mahalanobisova, Manhattan [113, 114] itd.) in iskanje K najbližjih sosedov (k -nearest neighbors – KNN), drugi algoritem pa določitev Bayesovega pravila [115]. Položaj uporabnika se določi kot povprečje koordinat, ki

jih vrne algoritem *KNN*. V kompleksnem notranjem okolju lahko meritve RSS znotraj vsake referenčne točke zelo variirajo. Zato so avtorji Savic [116] in drugi ter Swangmuang in drugi [117] predlagali uporabo Bayesovega pravila. Slednji upošteva verjetnostno metodo, ki omogoča najti najbolj verjetno lokacijo l iz referenčnih točk (iskanje vektorja RSS, ki maksimizira pogojno verjetnost $p(l/RSS)$).

Zaradi nihanja jakosti signalov v posameznih referenčnih točkah, so avtorji Altintas in drugi [118] upoštevali tudi zgodovino preteklih položajev sprejemnika in zgodovino meritev jakosti WiFi signalov v posameznih točkah.

Liu [94] in drugi so namesto metode *KNN* ali Bayesovega pravila predlagali uporabo nevronske mreže tj. modela večplastnega perceptrona (ang. Multilayer perceptron (MLP)) z eno prekrito plastjo. Pri uporabi tega pristopa so v fazi učenja jakosti signalov in pripadajoče koordinate položaja uporabljene kot vhod oz. izhod (»target«) za določitev uteži. V fazi določanja položaja se vhodni vektor, ki nosi informacijo o trenutnih jakostih signalov, pomnoži z matriko vhodnih uteži. Rezultat se nato upošteva v prenosni funkciji prekritega sloja nevrona, čigar izhod je nato pomnožen še z matriko uteži naučenega skritega sloja. Izhod iz sistema predstavlja oceno položaja.

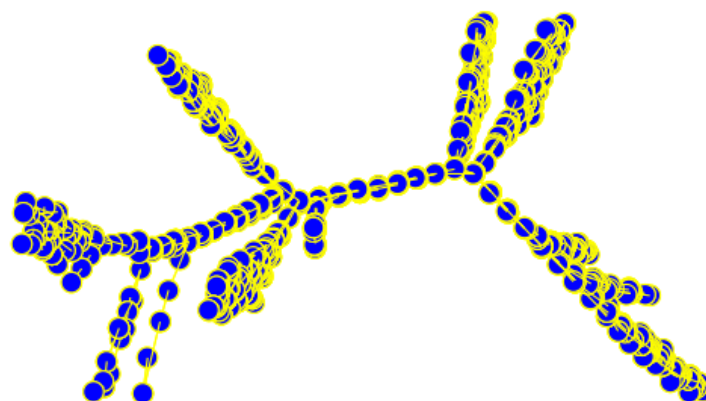
Pri implementaciji lokalizacijskega sistema, ki za svoje delovanje uporablja podatkovno bazo (WiFi) prstnih odtisov, se pojavljajo številni izzivi [102, 116]. Kot prvi problem velja omeniti, da je frekvenčni pas okoli 2,4 Ghz, ki ga poleg IEEE 802.11 WiFi omrežja uporabljajo tudi mobilne naprave, mikrovalovne pečice in drugi brezžični oddajniki, zelo obremenjen. Posledično se vse bolj uporablja tudi frekvenčni pas okoli 5,8 Ghz. Kot drugi problem velja omeniti, da prisotnost človeškega telesa med sprejemnikom (mobilno napravo) in oddajnikom prav tako vpliva na jakost sprejetega signala [119]. Ker so za natančno delovanje tega pristopa pomembne meritve skozi daljše časovno obdobje [112], lahko vsaka sprememba lokacije oddajnika povzroči nestabilno delovanje. To zahteva ponovno učenje prstnih odtisov, kar pomeni veliko dodatnega dela.

Lim in drugi [120] so predlagali avtomatsko posodabljanje podatkovne baze jakosti WiFi signalov. Njihova rešitev je koristna, saj se jakosti v istih referenčnih točkah spreminjajo, še posebej v primeru, če se premakne lokacija dostopne točke ali pa se doda nova dostopna točka.

Mirowski in drugi [121] so predlagali algoritem, ki namesto posameznih jakosti signalov v referenčni točki (v točki prstnega odtisa) upošteva verjetnostno porazdelitev (ki je opisana s histogrami namesto z Gaussovo krivuljo). Za iskanje podobnosti med posameznimi prstnimi odtisi so uporabili »Kullback-Leiber« divergenco, samo lokalizacijo pa so izvedli s t.i. jedrno regresijo (ang. »kernel regression«). Avtorji so pokazali, da njihov pristop prekaša rešitve, ki temeljijo na uporabi *KNN* metode, Kalmanovega filtra in filtra delcev, saj so dosegli natančnost pozicioniranja na 1 m.

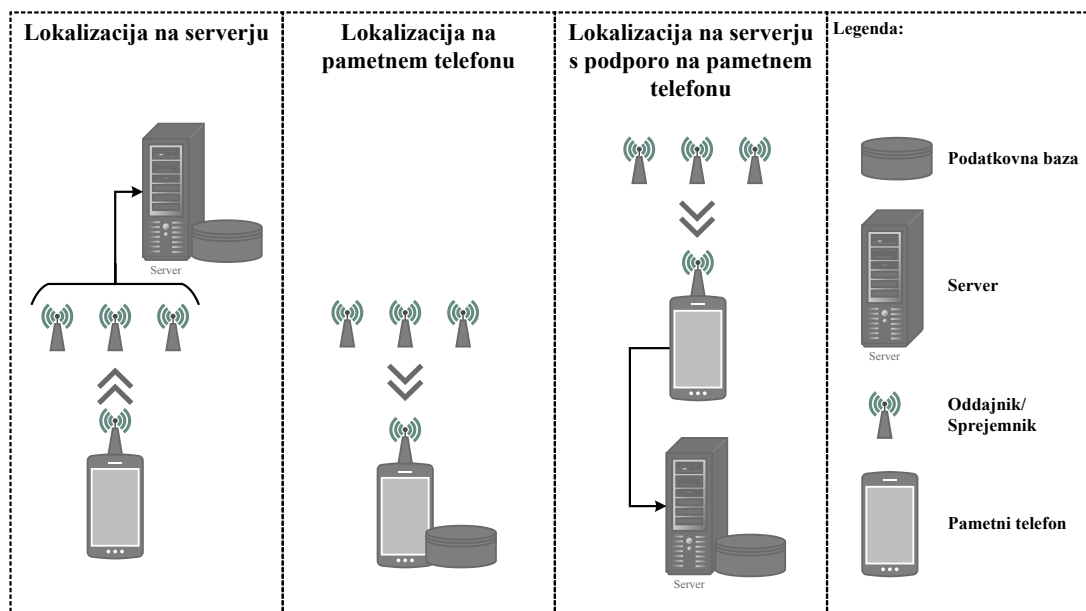
Enega izmed najbolj praktičnih pristopov k uporabi prstnih odtisov sta predlagala Youssef in Agrawala [122], saj je njun sistem Horus pogosto referenca za druge podobne sisteme. Avtorja sta namesto metode najbližjega sosedu uporabila pristop, ki temelji na verjetnostnem modeliranju. Predpostavila sta, da je časovna porazdelitev jakosti signalov v vsaki referenčni točki Gaussova (ali normalna). Natančnost sistema je v povprečju zelo visoka, in sicer okoli 1 m. Na področju radijske lokalizacije se uporablja tudi strojno učenje (npr. metoda SVM – support vector machine) s katerim na hevrističen način ocenimo trenutno lokacijo uporabnika [123]. Poleg metode SVM se uporablja tudi metoda skritega Markovskega modela (HMM-hidden Markov model) [124, 125, 126], FURIA [127] in drugi pristopi strojnega učenja.

Yang in drugi [128] so v svojem delu predlagali sistem LiFS (ang. Locating in Fingerprint Space), ki temelji na prstnih odtisih opisanih v visoko dimenzijskem prostoru (slika 2.15), znotraj katerega so posamezni prstni odtisi povezani glede na zaporedje po katerem so bili zajeti. Torej gre za medsebojno relacijo med prstnimi odtisi ter za povezavo prstnih odtisov z zemljevidom prostora. Avtorji so izgradnjo radijskega zemljevida, ki vključuje prstne odtise, popolnoma avtomatizirali, za kar so uporabili inercialne senzorje mobilne naprave. Razdalja med posameznimi prstnimi odtisi je določena s številom korakov, ki jih naredi uporabnik med posameznima odtisoma. Pri tej tehniki torej gre za primer uporabe metode SLAM. WiFi-SLAM, ki so ga predlagali Ferris in drugi [129], uporablja Gaussove procese pri povezovanju modela človeškega gibanja in prstnih odtisov. WiFi-SLAM so Huang in drugi [130] izboljšali z metodo GraphSLAM. Slednja metoda je predvsem računsko manj potratna.



Slika 2.15: Minimalno razpeto drevo 3D prstnih odtisov [128]

Različni načini izvedbe WiFi/Bluetooth lokalizacije [124] na serverju oz. mobilni napravi so prikazani na sliki 2.16. Večina sistemov za lokalizacijo z uporabo WiFi/Bluetooth omrežja je zgrajeno tako, da bazne postaje oddajajo signal, mobilne naprave pa merijo jakost sprejetih signalov. Cypriani in drugi [131] pa so predlagali obrnjen sistem, pri katerem bazne postaje analizirajo signale mobilnih naprav. Ta rešitev pride v poštev predvsem pri mobilnih napravah, ki onemogočajo merjenje jakosti WiFi signalov (npr. Apple iPhone) in pri uporabi t.i. WiFi značk [132].



Slika 2.16: Različne možnosti izvedbe WiFi/Bluetooth lokalizacije [124]

Alternativni pristop k določanju razdalje med dostopno točko in mobilno napravo so tehnike, ki temeljijo na merjenju časa. Najbolj pogosto se uporabljata metodi TOA (ang. »time of arrival«) in TDOA (ang. »time difference of arrival«). Za določitev položaja uporabnika se ti dve metodi uporabljata v kombinaciji s trilateracijo (torej potrebujemo vsaj tri bazne postaje). Pri tehniki TOA se meri čas, ki ga potrebuje paket z informacijo, da prepotuje razdaljo od bazne postaje do mobilne naprave. V tem primeru morajo biti bazne postaje časovno sinhronizirane z mobilno napravo, kar pogosto ni mogoče na običajnih (WiFi) dostopnih točkah, saj čas ni dovolj natačen (potrebna je posebna strojna oprema). Druga tehnika – TDOA se razlikuje od prve po tem, da meri razliko časa potrebnega za prenos informacije [98]. Posledično v tem primeru ni potrebna časovna sinhronizacija baznih postaj in mobilne naprave. Potrebna pa je medsebojna časovna sinhronizacija baznih postaj. V nekaterih primerih je časovna konstanta potovanja informacije opisana tudi s hiperbolo (namesto krožnice). V tem primeru sta za določitev položaja potrebni vsaj dve bazni postaji. Presečišče dveh hiperbol, ki kažeta konstanten čas potovanja informacije, določa položaj mobilne naprave.

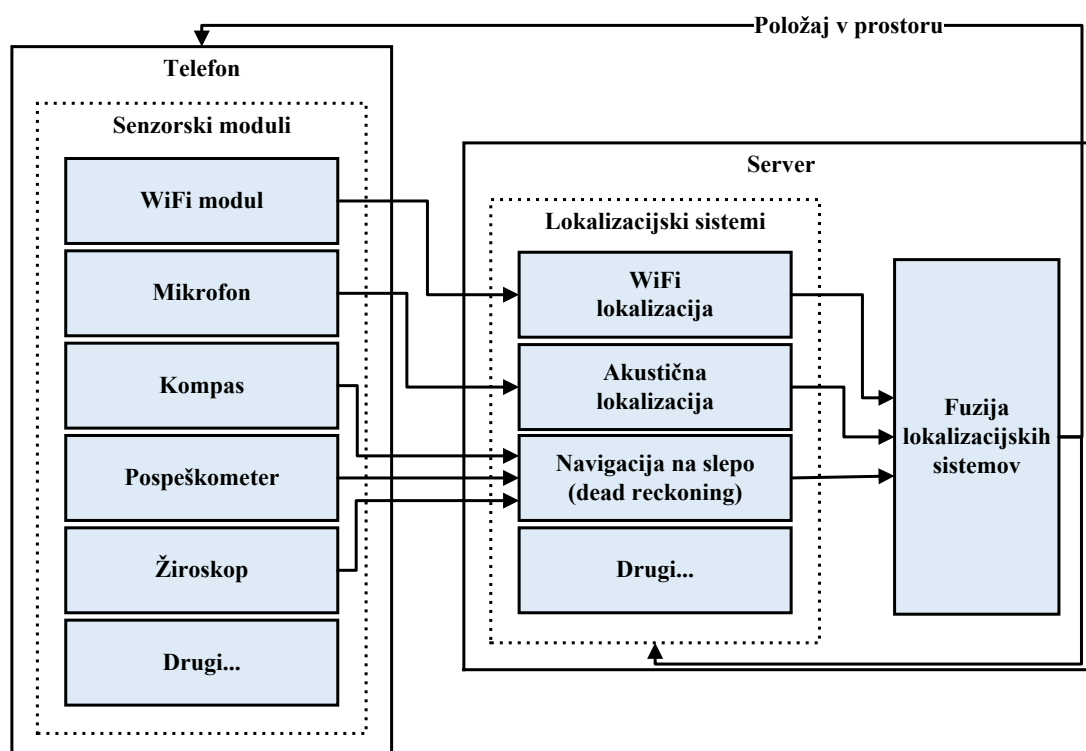
Thomas [133] je v svoji dizertaciji obravnaval različne tehnike lokalizacije, ki se lahko uporabljajo v kombinaciji s kognitivnim radiom (ang. Cognitive Radio). Kognitiven radio je sodoben pristop, uporabljen pri inteligentnih brezžičnih omrežjih, ki prilagajajo razpoložljive kanale glede na obremenjenost. Algoritem zazna komunikacijske kanale, ki so na voljo in nato prilagodi parametre prenosa in sprejema tako, da lahko hkrati deluje več brezžičnih komunikacij v istem spektralnem pasu na eni lokaciji. Pri kognitivnem lokalizacijskem sistemu je ključna komponenta model določanja pasovne širine.

2.4 Fuzija različnih metod lokalizacije

Pri opisu vizualne, inercialne in radijske lokalizacije smo pojasnili prednosti in slabosti različnih pristopov k lokalizaciji. Z namenom, da bi odpravili pomanjkljivosti posameznih metod lokalizacije, so se številni avtorji lotili fuzije (združevanja) več različnih metod [6, 134, 85]. Pojem fuzija se je že zelo dobro uveljavil v povezavi z inercialnimi senzorji, saj uporaba posameznih senzorjev ne vrača pričakovanih rezultatov. Združevanje različnih lokalizacijskih pristopov je dobilo svoje mesto tudi

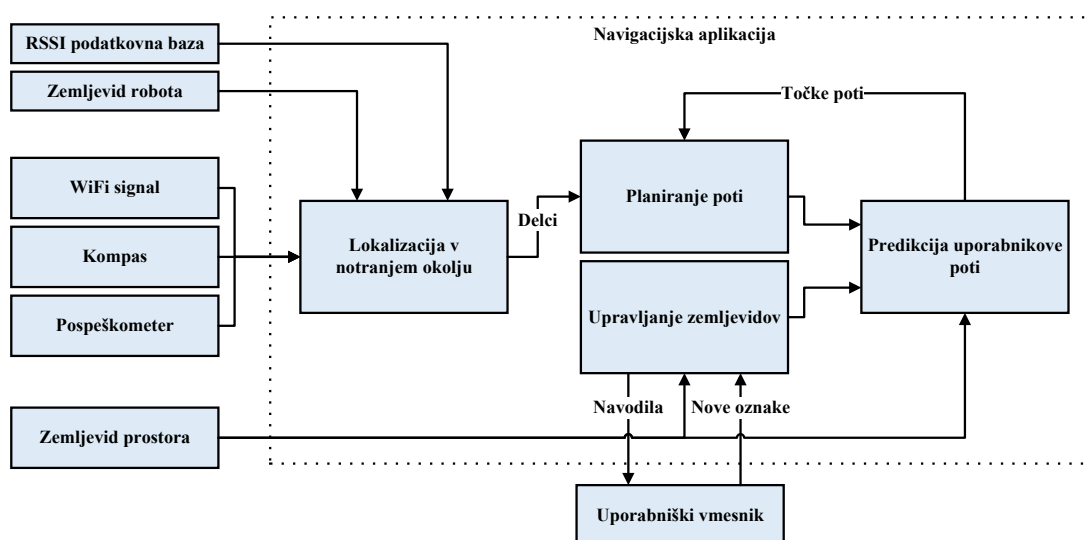
pri lokalizaciji s kamero, saj so za doseg želenih rezultatov velikokrat potrebni tudi inercialni senzorji [18, 50]. Sistem vizualne odometrije, ki sta ga predlagala Lupton in Sukkarieh [18], uporablja poleg kamere tudi inercialne senzorje. Sistem uspešno sledi trajektoriji za okoli 750 zaporednih slikovnih okvirjev, hkrati pa uspešno določi pravo dolžino poti z inercialnimi senzorji. Ti so namreč pomembni za določitev skalirnega faktorja, saj se prave dolžine poti v tem primeru ne da določiti le z eno kamero. Ta sistem lahko oceni trajektorijo z natančnostjo manj kot 1 % dolžine poti.

Inercialni senzorji pa se pogosto kombinirajo tudi z radijsko (WiFi/Bluetooth) lokalizacijo npr. za gradnjo zemljevida prstnih odtisov [128, 135] ali za izboljšanje natančnosti določanja orientacije. Chen in Vadde [136] sta predlagala sistem, ki združuje WiFi lokalizacijo (z uporabo prstnih odtisov in metode *KNN*), uporabo inercialnih senzorjev in uporabo zvoka za natančno lokalizacijo v zaprtem prostoru (slika 2.17). Za združevanje informacij sta avtorja uporabila preprost Kalmanov filter, celoten algoritem lokalizacije pa sta implementirala na serverju (mobilna naprava le pošilja meritve iz senzorjev in komunikacijskih modulov na server).



Slika 2.17: Fuzija različnih metod za natančno lokalizacijo v zaprtem prostoru [136]

Kannan in drugi [134] pa so predlagali implementacijo fuzije različnih lokalizacijskih metod (WiFi lokalizacija, inercialni senzorji) kar na pametnem telefonu (slika 2.18). Njihov sistem deluje prediktivno, saj intuitivno načrtuje uporabnikovo pot glede na njegove namere (npr. smer potovanja, hitrost). Sistem so implementirali na pametnem telefonu z OS Android. Kot je prikazano na sliki 2.18, so Kannan in drugi [134] združili tehniki navigacije na slepo in WiFi lokalizacije s prstnimi odtisi. Za natančno oceno položaja so uporabili znan zemljevid prostora in filter delcev s katerim so združili meritve. V bloku predikcije (na sliki 2.18) se izvaja Markovski odločitveni proces, ki vrne možne nadaljnje poti uporabnika. Pri testiranju natančnosti lokalizacije je bila napaka v povprečju manjša od 2,5 m.



Slika 2.18: Shematski prikaz sistema lokalizacije, ki so ga predlagali Kannan in drugi [134]

Podjetje Microsoft je v okviru svojega projekta lokalizacije v zaprtem prostoru razvilo sistem »Walkie-Markie« [137], ki združuje WiFi pozicioniranje in uporabo inercialnih senzorjev za izgradnjo zemljevida jakosti signalov. Lokalizacijski sistem brez v naprej podanih informacij o stavbi zgradi zemljevid, na katerem so določene glavne poti po zgradbi. Trajektorijam uporabnikov, ki jih sistem določi z inercialnimi senzorji (gre za implementacijo pedometra in digitalnega kompasa) se dodelijo pripadajoče jakosti WiFi signalov (oz. njihove porazdelitve). Glede na dobljene značke (to so prstni odtisi sestavljeni iz jakosti signalov) sistem združuje trajektorije različnih uporabnikov. Več kot sistem združi trajektorij, bolj natančna je končna

trajektorija na zemljevidu. V primeru, da uporabnik določeno pot prehodi vsaj petkrat, jo sistem že lahko dovolj natančno rekonstruira. Nove značke se postavijo samo, če uporabnik hodi konstantno in ne prehitro. Pri testu lokalizacijskega sistema so avtorji ugotovili, da je največja napaka (razlika med dejanskim in ocenjenim položajem) enaka 3 m.

3 Vizualna odometrija

3.1 Uvod v monokularno vizualno odometrijo

Monokularna vizualna odometrija je proces zaporednega ocenjevanja gibanja kamere glede na zaznane premike slikovnih točk iz sekvence slik. Pridevnik »monokularna« nam pove, da ta vrsta odometrije temelji na zajemu slike z eno kamero, v splošnem pa lahko vizualna odometrija temelji tudi na stereo vidu.

Vizualna odometrija je v splošnem sestavljena iz štirih algoritmov, in sicer kalibracije kamere, sledilnika značilnih točk, algoritma za oceno modela togega premika točk in algoritma RANSAC (ki deluje v navezi z algoritmom za oceno modela togega premika). Ker so vsi ti ključni deli vizualne odometrije celota zase, smo jih v nadaljevanju razdelili na štiri podpoglavja. Tako v drugem podpoglavju obravnavamo kalibracijo kamere, v tretjem podpoglavju obravnavamo optični tok na podlagi Kanade-Lucas-Tomasi sledilnika značilnih točk, v četrtem podpoglavju je predstavljen algoritem RANSAC, ki skrbi za izločanje odstopajočih točk pri izbranem modelu togega premika. Opis togega premika ter način ocene togega premika točk v 2D prostoru sta predstavljena v petem podpoglavju. V šestem podpoglavju je predstavljeno, kako štiri omenjene algoritme združimo v celoto in na ta način dobimo algoritem vizualne odometrije.

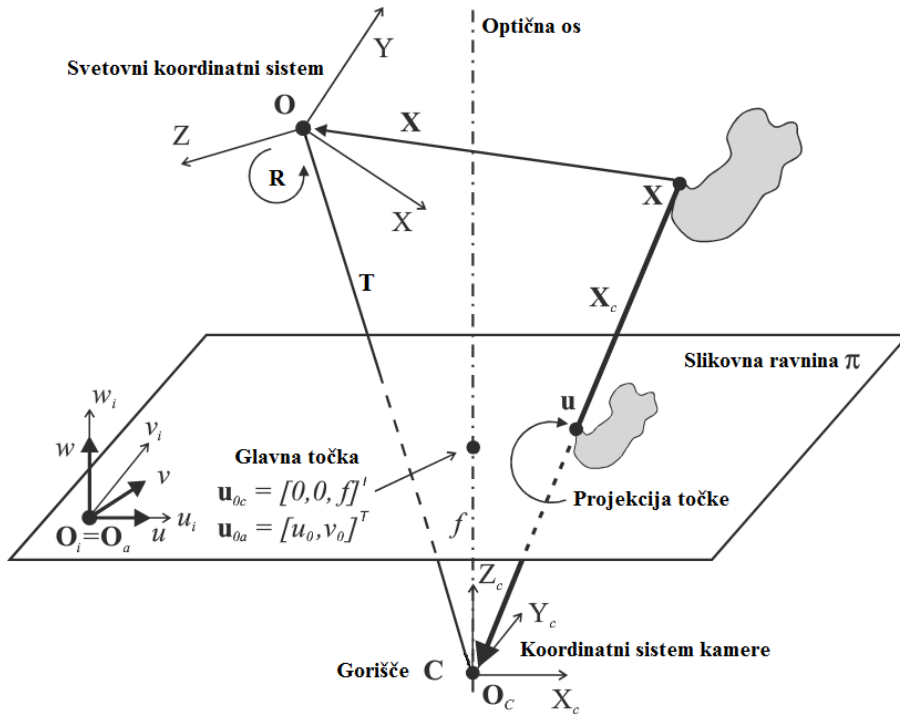
Algoritem smo implementirali na pametnem mobilnem telefonu z operacijskim sistemom Android. Pri implementaciji smo uporabili odprtokodno knjižnico BoofCV, ki je napisana v programskem jeziku Java in združuje množico uporabnih funkcij s področja strojnega vida. Pri preizkusu različnih delov algoritma smo si pomagali tudi s programskim okoljem Matlab.

3.2 Kalibracija kamere

3.2.1 Model kamere luknjičarke

Če predpostavimo, da ima kamera tanko lečo, potem jo lahko opišemo z modelom kamere – luknjičarke (ang. pinhole) [138]. Kljub temu, da ta model ne opisuje kamere povsem natančno, je primeren za veliko aplikacij na področju strojnega vida. Geometrija modela kamere je prikazana na sliki 3.1.

Na sliki 3.1 vidimo, da ravnina π predstavlja slikovno ravnino na katero se projicira opazovani prizor. Leča kamere je postavljena pravokotno na optično os v gorišču **C** (središče projekcije ali optični center). Goriščna razdalja je označena s f in predstavlja razdaljo od središča leče pa do slikovne ravnine. Slikovne točke označimo z malimi črkami, in sicer v evklidskem prostoru kot $\mathbf{u} = [u, v]^T$ ter s homogenimi koordinatami kot $\tilde{\mathbf{u}} = [u, v, w]^T$. Točke v 3D prostoru označimo z velikimi črkami kot $\mathbf{X} = [X, Y, Z]^T$ ali s homogenimi koordinatami kot $\tilde{\mathbf{X}} = [X, Y, Z, W]^T$.



Slika 3.1: Geometrija modela kamere luknjičarke

Pri kameri se izvaja linearna transformacija iz 3D projektivnega prostora P^3 v 2D projektivni prostor P^2 . Projicirano točko $\mathbf{u}$ določa optični žarek, ki se odbije od točke $\mathbf{X}$ opazovane scene, potuje skozi optični center in nato zadane slikovno ravnino.

Štiri koordinatne sisteme, ki so prikazani na sliki 3.1, lahko opišemo na naslednji način:

1. Svetovni koordinatni sistem ima izhodišče v točki $\mathbf{O}$. Točko $\mathbf{X}$ zapišemo v svetovnem koordinatnem sistemu kot: $\mathbf{X} = [X, Y, Z]^T$.
2. Koordinatni sistem kamere ima izhodišče v točki $\mathbf{O}_c$, ki je enaka gorišču $\mathbf{C}$. Os Z_c je poravnana z optično osjo ter kaže iz gorišča proti slikovni ravnini. Povezavo med svetovnim koordinatnim sistemom in koordinatnim sistemom kamere določata translacija $\mathbf{T}$ in rotacija $\mathbf{R}$.
3. Evklidski koordinatni sistem slike (označuje indeks i) ima osi poravnane s koordinatnim sistemom kamere. Osi u_i in v_i ležita na slikovni ravnini.
4. Afini koordinatni sistem slike (označuje indeks a) ima koordinatne osi u , v , w in izhodišče $\mathbf{O}_a$, ki se ujema z izhodiščem evklidskega koordinatnega sistema $\mathbf{O}_i$. Osi u in w sta poravnani z osemi u_i in w_i , os v pa lahko ima različno orientacijo kot os v_i .

Razlog za vpeljavo afinega koordinatnega sistema je dejstvo, da se lahko pri slikovnih točkah pojavi striženje (ang. shear) zaradi slabo poravnanih elementov kamere.

Kot smo že omenili, prvo transformacijo med svetovnim koordinatnim sistemom in koordinatnim sistemom kamere določata translacijski vektor $\mathbf{T}$ in rotacijska matrika $\mathbf{R}$. Točko $\mathbf{X}$ (v nehomogenih koordinatah) v koordinatnem sistemu (K.S.) kamere zapišemo kot: $\mathbf{X}_c = \mathbf{R}\mathbf{X} + \mathbf{T}$. V homogenih koordinatah pa lahko zapišemo:

$$\tilde{\mathbf{X}}_c = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \tilde{\mathbf{X}}. \quad (3.1)$$

Drugo transformacijo med K.S. kamere in evklidskim K.S. slike lahko zapišemo na podlagi projekcije 3D točke $\mathbf{X}_c$ (izražena v K.S. kamere) v točko $\mathbf{u}_i$ na slikovni ravnini:

$u_i = \frac{x_c f}{z_c}$ in $v_i = \frac{y_c f}{z_c}$. V homogenih koordinatah lahko preslikavo $P^3 \rightarrow P^2$ zapišemo kot:

$$\tilde{\mathbf{u}}_i \cong \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{X}}_c. \quad (3.2)$$

Če je $f = 1$ dobimo točke $\mathbf{u}_i$ na normalizirani slikovni ravnini.

Tretjo transformacijo med evklidskim K.S. in afinim K.S. določajo intrinzični parametri v matriki A (pomen parametrov je pojasnjen v nadaljevanju). Tako lahko transformacijo $P^2 \rightarrow P^2$ zapišemo kot:

$$\tilde{\mathbf{u}} \cong A\tilde{\mathbf{u}}_t = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{\mathbf{u}}_t. \quad (3.3)$$

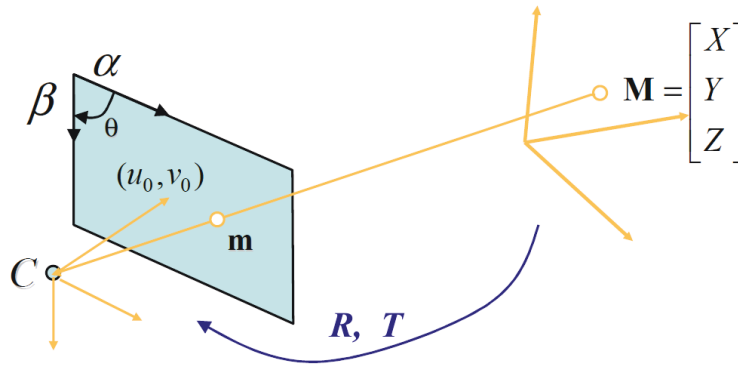
Sedaj lahko zapišemo celotno transformacijo točk iz 3D projektivnega prostora v 2D projektivni prostor:

$$\tilde{\mathbf{u}} \cong A \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \tilde{\mathbf{X}} = A[R | T]\tilde{\mathbf{X}} = P\tilde{\mathbf{X}}. \quad (3.4)$$

Če imamo točke opazovane scene zapisane v homogenih koordinatah, potem lahko perspektivno projekcijo izrazimo s projekcijsko matriko P velikosti 3×4 .

3.2.2 Uvod v kalibracijo

Označimo z $\mathbf{m} = [u, v]^T$ točko, ki se nahaja na sliki (2D prostor) ter z $\mathbf{M} = [X, Y, Z]^T$ točko, ki se nahaja v 3D prostoru (slika 3.2) [139].



Slika 3.2: Model kamere luknjičarke (ang. pinhole camera)

Točki $\mathbf{m}$ in $\mathbf{M}$ lahko zapišemo v homogenih koordinatah kot $\tilde{\mathbf{m}} = [u, v, 1]^T$ ter $\tilde{\mathbf{M}} = [X, Y, Z, 1]^T$. Homogene koordinate se uporabljajo v projektivni geometriji, pri čemer je njihova prednost ta, da lahko točke, vključno s točkami v neskončnosti, prikažemo z uporabo končnih koordinat. Kadar homogene koordinate točke množimo z neničelnim skalarjem, nam dobljene koordinate predstavljajo isto točko. Za točko s koordinatama (x, y) v evklidskem prostoru dobimo homogene koordinate tako, da

dodamo tretjo koordinato $(x, y, 1)$. Če je tretja koordinata enaka nič, potem ta točka v evklidskem prostoru predstavlja točko v neskončnosti.

Za opis kamere uporabimo model luknjičarke (ang. pinhole camera), ki določa povezavo med 3D točko M in njeno projekcijo na sliko – točko m :

$$s\tilde{m} = A[R \mid T]\tilde{M}, \quad (3.5)$$

kjer je s skalirni faktor. Matriki R (rotacijska matrika) in T (translacijska matrika) predstavljata ekstrinzične parametre, ki opisujejo povezavo med svetovnim koordinatnim sistemom in koordinatnim sistemom kamere. A je matrika intrinzičnih parametrov kamere in je podana kot:

$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.6)$$

kjer je $(u_0 \ v_0)$ optično središče (ang. principal point), $\alpha = fs_x$ in $\beta = fs_y$ sta goriščni razdalji izraženi v slikovnih točkah glede na koordinatni osi u in v . f je goriščna razdalja izražena v milimetrih, s_x in s_y pa sta skalirna faktorja glede na koordinatni osi u in v , ki določata število slikovnih točk na milimeter. Parameter γ (ang. skew) predstavlja popačenje slikovne točke (v primeru, da posamezna slikovna točka ni kvadratna).

Brez da bi izgubili na splošnosti, lahko predpostavimo, da točke kalibracijskega objekta ležijo na ravnini $Z = 0$ svetovnega koordinatnega sistema. Če označimo z r_i i -ti stolpec matrike R in to upoštevamo v enačbi (3.5), dobimo:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A[r_1 \ r_2 \ r_3 \ T] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A[r_1 \ r_2 \ T] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \quad (3.7)$$

Ker je Z vedno enaka nič, lahko točko M zapišemo kot $M = [X \ Y]^T$ oz. v homogenih koordinatah kot $\tilde{M} = [X \ Y \ 1]^T$. Povezavo med točkama m in M lahko sedaj zapišemo kot:

$$s\tilde{m} = H\tilde{M}, \quad (3.8)$$

pri čemer je $H = A[r_1 \ r_2 \ T]$ homografija, ki je določena do λ skalirnega faktorja natančno. Torej lahko zapišemo:

$$[h_1 \ h_2 \ h_3] = \lambda A[r_1 \ r_2 \ T]. \quad (3.9)$$

Če upoštevamo, da sta vektorja r_1 in r_2 ortonormalna, lahko zapišemo dve enačbi za omejitev intrinzičnih parametrov:

$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2 = 0, \quad (3.10)$$

$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2. \quad (3.11)$$

Omejitvi (3.10) in (3.11) potrebujemo, ker ima homografija osem prostostnih stopenj, ekstrinzičnih parametrov pa je le šest (trije za rotacijo in trije za translacijo). $\mathbf{A}^{-T} \mathbf{A}^{-1}$ opisuje sliko absolutnega stožca (ang. absolute conic).

3.2.3 Ocena homografije med kalibracijskim objektom in njegovo sliko

Iz množice načinov za oceno homografije med ravninsko kalibracijsko tarčo (ravnina pokrita z vzorcem šahovnice) in njeno sliko, smo izbrali tehniko, ki temelji na kriteriju maksimalne verjetnosti [139]. Ker v praksi točke $\mathbf{m}_i$, ki ležijo na sliki, vsebujejo šum, enačba (3.8) ne drži. Predpostavimo, da točke $\mathbf{m}_i$ vsebujejo Gaussov šum, ki ima srednjo vrednost 0 in kovariančno matriko $\Lambda_{\mathbf{m}_i}$. Maksimalno verjetnost ocene homografije $\mathbf{H}$ dosežemo z minimiziranjem izraza:

$$\sum_i (\mathbf{m}_i - \hat{\mathbf{m}}_i)^T \Lambda_{\mathbf{m}_i}^{-1} (\mathbf{m}_i - \hat{\mathbf{m}}_i), \quad (3.12)$$

kjer je $\hat{\mathbf{m}}_i = \frac{1}{\bar{h}_3 \mathbf{M}_i} \begin{bmatrix} \bar{\mathbf{h}}_1^T \mathbf{M}_i \\ \bar{\mathbf{h}}_2^T \mathbf{M}_i \end{bmatrix}$ in $\bar{\mathbf{h}}_i$ je i -ta vrstica matrike $\mathbf{H}$.

V praksi predpostavimo, da velja $\Lambda_{\mathbf{m}_i} = \sigma^2 \mathbf{I}$ (σ^2 je varianca točk $\mathbf{m}_i$) za vse i , saj so točke pridobljene neodvisno in z enakim postopkom. Potem lahko izraz iz enačbe (3.12) zapišemo kot problem nelinearne minimizacije, ki temelji na metodi najmanjših kvadratov $\min_H \sum_i \|\mathbf{m}_i - \hat{\mathbf{m}}_i\|^2$. Slednji problem se reši z uporabo algoritma Levenberg-Marquardt.

Če definiramo matriko $\mathbf{x} = [\bar{\mathbf{h}}_1^T, \bar{\mathbf{h}}_2^T, \bar{\mathbf{h}}_3^T]^T$, potem lahko enačbo (3.8) zapišemo kot:

$$\tilde{\mathbf{m}} \times \mathbf{H} \tilde{\mathbf{M}} = ([\tilde{\mathbf{m}}]_{\times} \otimes \tilde{\mathbf{M}}^T) \mathbf{x} = \begin{bmatrix} \tilde{\mathbf{M}}^T & 0^T & -u \tilde{\mathbf{M}}^T \\ 0^T & \tilde{\mathbf{M}}^T & -v \tilde{\mathbf{M}}^T \end{bmatrix} \mathbf{x} = 0 \quad (3.13)$$

oz. $\mathbf{L} \mathbf{x} = 0$, kjer je $\mathbf{L}$ matrika velikosti $2n \times 9$ (n je število točk, $\otimes$ pa predstavlja Kroneckerjev produkt). Ker je matrika $\mathbf{x}$ definirana do skalirnega faktorja natančno, je rešitev desni singularni vektor matrike $\mathbf{L}$, ki je povezan z najmanjšo singularno vrednostjo (lastni vektor matrike $\mathbf{L}^T \mathbf{L}$, ki je povezan z najmanjšo lastno vrednostjo (dodatek A)). Če želimo, da je rešitev enačbe (3.13) dobro pogojena, moramo točke podati v normalizirani obliki.

3.2.4 Ocena intrinzičnih in ekstrinzičnih parametrov

Kalibracijske parametre v prvem koraku ocenimo z upoštevanjem analitične rešitve. V drugem koraku to rešitev optimiziramo z uporabo nelinearne optimizacijske tehnike, ki temelji na kriteriju maksimalne verjetnosti. V tretjem koraku je upoštevano še popačenje leče.

Naj bo

$$\begin{aligned} \mathbf{B} &= \mathbf{A}^{-T} \mathbf{A}^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} = \\ &= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2 \beta} & \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} \\ -\frac{\gamma}{\alpha^2 \beta} & \frac{\gamma^2}{\alpha^2 \beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0 \gamma - u_0 \beta)^2}{\alpha^2 \beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix}. \end{aligned} \quad (3.14)$$

Za matriko $\mathbf{B}$ velja, da je simetrična matrika, ki jo lahko definiramo kot 6D vektor $\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T$. Če zapišimo i -ti stolpec matrike $\mathbf{H}$ kot $\mathbf{h}_i = [h_{i1}, h_{i2}, h_{i3}]^T$, potem velja:

$$\mathbf{h}_i^T \mathbf{B} \mathbf{h}_j = \mathbf{v}_{ij}^T \mathbf{b}, \quad (3.15)$$

kjer je $\mathbf{v}_{ij} = [\mathbf{h}_{i1} \mathbf{h}_{j1}, \mathbf{h}_{i1} \mathbf{h}_{j2} + \mathbf{h}_{i2} \mathbf{h}_{j1}, \mathbf{h}_{i2} \mathbf{h}_{j2}, \mathbf{h}_{i3} \mathbf{h}_{j1} + \mathbf{h}_{i1} \mathbf{h}_{j3}, \mathbf{h}_{i3} \mathbf{h}_{j2} + \mathbf{h}_{i2} \mathbf{h}_{j3}, \mathbf{h}_{i3} \mathbf{h}_{j3}]^T$.

Omejitvi (3.10) in (3.11) lahko zapišemo v obliki dveh homogenih enačb kot:

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = 0. \quad (3.16)$$

Če imamo n slik kalibracijske ravnine, imamo tudi n enačb (3.16), ki jih združimo in zapišemo kot:

$$\mathbf{V} \mathbf{b} = 0, \quad (3.17)$$

kjer je $\mathbf{V}$ matrika velikosti $2n \times 6$. Če želimo enolično rešitev $\mathbf{b}$, mora biti $n \geq 3$. Rešitev za enačbo (3.17) dobimo z uporabo singularnega razcepa, kot lastni vektor matrike $\mathbf{V}^T \mathbf{V}$, ki mu pripada najmanjša lastna vrednost.

Ko imamo ocenjen vektor $\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T$, lahko izračunamo vse intrinzične parametre:

$$\begin{aligned} v_0 &= (B_{12} B_{13} - B_{11} B_{23}) / (B_{11} B_{22} - B_{12}^2), \\ \lambda &= B_{33} - (B_{13}^2 + v_0 (B_{12} B_{13} - B_{11} B_{23})) / B_{11}, \end{aligned}$$

$$\begin{aligned}
\alpha &= \sqrt{\lambda/B_{11}}, \\
\beta &= \sqrt{\frac{\lambda B_{11}}{B_{11}B_{22}-B_{12}^2}}, \\
\gamma &= -B_{12}\alpha^2\beta/\lambda, \\
u_0 &= \frac{\gamma v_0}{\beta} - B_{13}\alpha^2/\lambda.
\end{aligned} \tag{3.18}$$

Ko imamo določeno matriko intrinzičnih parametrov $\mathbf{A}$, lahko glede na enačbo (3.8) izračunamo še ekstrinzične parametre:

$$\begin{aligned}
\mathbf{r}_1 &= \lambda^{-1}\mathbf{A}^{-1}\mathbf{h}_1, \\
\mathbf{r}_2 &= \lambda^{-1}\mathbf{A}^{-1}\mathbf{h}_2, \\
\mathbf{r}_3 &= \mathbf{r}_1 \times \mathbf{r}_2, \\
\mathbf{T} &= \lambda^{-1}\mathbf{A}^{-1}\mathbf{h}_3,
\end{aligned} \tag{3.19}$$

kjer je $\lambda = \|\mathbf{A}^{-1}\mathbf{h}_1\| = \|\mathbf{A}^{-1}\mathbf{h}_2\|$. Matrika $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$, ki jo dobimo, v splošnem ne zadošča lastnostim rotacijske matrike. Zato moramo za matriko $\mathbf{Q} = \mathbf{R}$ poiskati najboljši približek, kot je to opisano v nadaljevanju.

3.2.5 Določanje rotacijske matrike

Želimo najti tako rotacijsko matriko, ki bo predstavljala najboljšo aproksimacijo dane matrike $\mathbf{Q}$ velikosti 3×3 . Torej želimo poiskati $\mathbf{R}$ najmanjšo Frobeniusovo normo razlike matrik $\mathbf{R}$ in $\mathbf{Q}$:

$$\min_{\mathbf{R}} \|\mathbf{R} - \mathbf{Q}\|_F^2, \tag{3.20}$$

pri čemer mora veljati $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ ($\mathbf{R}$ je ortogonalna matrika in $\mathbf{I}$ je enotska matrika). Ker velja $\|\mathbf{R} - \mathbf{Q}\|_F^2 = \text{sled}((\mathbf{R} - \mathbf{Q})^T(\mathbf{R} - \mathbf{Q})) = 3 + \text{sled}(\mathbf{Q}^T \mathbf{Q}) - 2\text{sled}(\mathbf{R}^T \mathbf{Q})$, je problem enak iskanju maksimuma $\text{sled}(\mathbf{R}^T \mathbf{Q})$.

Če je singularni razcep matrike $\mathbf{Q}$ enak $\mathbf{Q} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, kjer je $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$ (pri čemer so $\sigma_1, \sigma_2, \sigma_3$ singularne vrednosti matrike $\mathbf{Q}$). Če definiramo neko novo ortogonalno matriko $\mathbf{Z} = \mathbf{V}^T \mathbf{R}^T \mathbf{U}$, potem lahko nadalje zapišemo:

$$\begin{aligned}
\text{sled}(\mathbf{R}^T \mathbf{Q}) &= \text{sled}(\mathbf{R}^T \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T) = \text{sled}(\mathbf{V}^T \mathbf{R}^T \mathbf{U} \mathbf{\Sigma}) \\
&= \text{sled}(\mathbf{Z} \mathbf{\Sigma}) = \sum_{i=1}^3 z_{ii} \sigma_i \leq \sum_{i=1}^3 \sigma_i.
\end{aligned}$$

Maksimum izraza $\text{sled}(\mathbf{R}^T \mathbf{Q})$ dobimo torej pri $\mathbf{R} = \mathbf{U}\mathbf{V}^T$, saj je potem $\mathbf{Z} = \mathbf{I}$.

3.2.6 Ocena maksimalne verjetnosti

Podana rešitev za oceno kalibracijskih parametrov temelji na minimiziranju algebrajske razdalje, ki nima fizičnega pomena. Ocenjene parametre lahko popravimo z upoštevanjem ocene maksimalne verjetnosti.

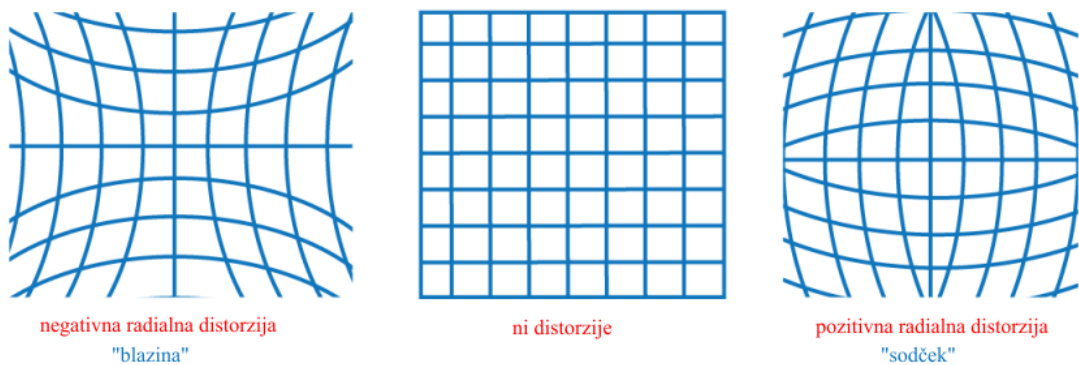
Imamo n slik kalibracijske ravnine in m točk iz te ravnine, ki vsebujejo šum. Maksimalno oceno verjetnosti dobimo z minimiziranjem naslednje funkcije:

$$\sum_{i=1}^n \sum_{j=1}^m \|\mathbf{m}_{ij} - \hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{T}_i, \mathbf{M}_j)\|^2, \quad (3.21)$$

kjer je $\hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{T}_i, \mathbf{M}_j)$ projekcija točke $\mathbf{M}_j$ na sliki i (skladno z enačbo (3.8)). Rotacijska matrika je parametrizirana z vektorjem $\mathbf{r}$ (ta vsebuje tri parametre), ki je paralelen z rotacijsko osjo in ima velikost, ki je enaka kotu vrtenja. Matrika $\mathbf{R}$ in vektor $\mathbf{r}$ sta povezana z Rodriguesovo formulo [140]. Enačbo (3.21) minimiziramo z uporabo algoritma Levenberg-Marquardt.

3.2.7 Kompenzacija radialne distorzije

Pri zajemu slike s kamero, ta v večini primerov vsebuje popačenje (distorzijo) zaradi leče. Najbolj pogosto se pojavlja radialna distorzija. Pri slednji se točke lahko odmaknejo v negativni ali pozitivni radialni smeri. Tako slika dobi obliko sodčka ali blazine (slika 3.3). Pogostejše pride do distorzije v obliki sodčka (ang. barrel), ki jo povzročajo širokokotne leče.



Slika 3.3: Levo je prikazano popačenje slikovnih točk pri negativni radialni distorziji in desno pri pozitivni radialni distorziji

Distorzijo zaradi leče lahko odpravimo tako, da jo opišemo z modelom, ki ga nato upoštevamo pri vseh zajetih slikah. Označimo z (u, v) slikovno točko, ki ne vsebuje

popačenja in z ($\acute{u}, \acute{v}$) slikovno točko, ki se nahaja na popačeni sliki. Idealne točke (u, v) predstavljajo projekcijo točk, ki se nahajajo na kalibracijski ravnini. Projekcija je določena preko modela kamere luknjičarke. Slikovne točke v normaliziranih koordinatah označimo na nepopačeni sliki z (x, y) in na popačeni sliki z $(\acute{x}, \acute{y})$. Popačeno točko lahko zapišemo kot:

$$\acute{x} = x + x(k_1(x^2 + y^2) + k_2(x^2 + y^2)^2), \quad (3.22)$$

$$\acute{y} = y + y(k_1(x^2 + y^2) + k_2(x^2 + y^2)^2), \quad (3.23)$$

kjer sta k_1 in k_2 koeficienta radialne distorzije. Če upoštevamo $\acute{u} = u_0 + \alpha\acute{x} + \gamma\acute{y}$ in $\acute{v} = v_0 + \beta\acute{y}$ ter $\gamma = 0$, dobimo:

$$\acute{u} = u + (u - u_0)(k_1(x^2 + y^2) + k_2(x^2 + y^2)^2), \quad (3.24)$$

$$\acute{v} = v + (v - v_0)(k_1(x^2 + y^2) + k_2(x^2 + y^2)^2), \quad (3.25)$$

Koeficienta k_1 in k_2 ocenimo tako, da najprej izračunamo vse kalibracijske parametre, z uporabo katerih dobimo idealne slikovne točke (u, v). Nato z upoštevanjem enačb (3.24) in (3.25) zapišemo dve enačbi za vse slikovne točke:

$$\begin{bmatrix} (u - u_0)(x^2 + y^2) & (u - u_0)(x^2 + y^2)^2 \\ (v - v_0)(x^2 + y^2) & (v - v_0)(x^2 + y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \acute{u} - u \\ \acute{v} - v \end{bmatrix}. \quad (3.26)$$

Če imamo n slik in na vsaki sliki m točk, potem je vseh enačb $2 \cdot m \cdot n$. Enačbo (3.26) lahko zapišemo v matrični obliki kot: $\mathbf{D}\mathbf{k} = \mathbf{d}$, kjer je $\mathbf{k} = [k_1, k_2]^T$. Rešitev dobimo po metodi najmanjših kvadratov:

$$\mathbf{k} = (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{d}. \quad (3.27)$$

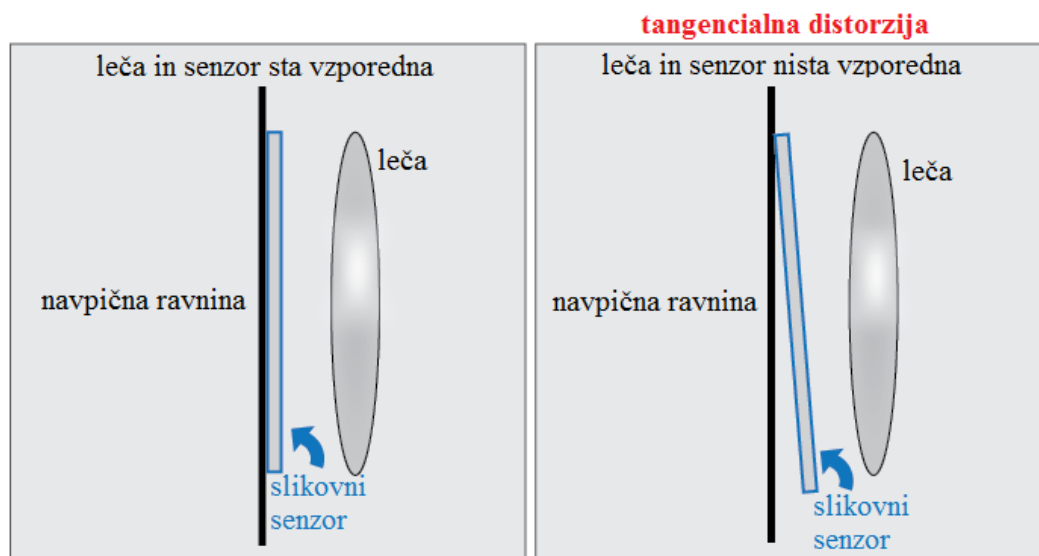
Ko imamo koeficienta k_1 in k_2 , lahko uporabimo enačbo (3.21), v kateri $\hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{T}_i, \mathbf{M}_j)$ zamenjamo z enačbama (3.24) in (3.25), da popravimo ostale kalibracijske parametre.

Postopka za iskanje kalibracijskih parametrov in parametrov radialne distorzije izmenično ponavljamo toliko časa, dokler ne dosežemo konvergence. Vendar, ker je ta konvergenca počasna, je bolje oceniti vse parametre hkrati z minimiziranjem naslednjega izraza:

$$\sum_{i=1}^n \sum_{j=1}^m \|\mathbf{m}_{ij} - \hat{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{T}_i, \mathbf{M}_j)\|^2, \quad (3.28)$$

kjer je $\hat{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{T}_i, \mathbf{M}_j)$ projekcija točke $\mathbf{M}_j$ na sliki i skladno z enačbo (3.8), pri čemer upoštevamo tudi radialno distorzijo s koeficientoma k_1 in k_2 . Minimizacija je izvedena z uporabo algoritma Levenberg-Marquardt.

Med distorzijami leč poznamo poleg radialne tudi tangencialno distorzijo. Slednja se pojavi v primeru, da so elementi kamere slabo poravnani (kot prikazuje slika 3.4, slikovni senzor in leča nista vzporedna), kar povzroči, da se točke odmaknejo v tangencialni smeri.



Slika 3.4: V primeru, da leča in senzor nista vzporedna, se pojavi tangencialna distorzija

Za opis radialne in tangencialne distorzije skupaj se uporablja Brownov »Plumb Bob« model distorzije (»Camera Calibration Toolbox for Matlab« [141]):

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_5 r^6) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2k_3 xy + k_4(r^2 + 2x^2) \\ k_3(r^2 + 2y^2) + 2k_4 xy \end{bmatrix}, \quad (3.29)$$

kjer so $[\hat{x} \ \hat{y}]$ koordinate popačene točke, $[x \ y]$ koordinate nepopačene točke ter $r = x^2 + y^2$. Prvi del enačbe predstavlja radialno komponento, drugi pa tangencialno komponento distorzije.

Povzetek postopka kalibracije kamere:

1. Zajamemo več slik kalibracijskega objekta pod različnimi koti.
2. Detektiramo vogale na kalibracijskem objektu.
3. Izračunamo homografije za vse zajete slike z uporabo enačbe (3.13).
4. Homografije upoštevamo v enačbi (3.17), s katero dobimo vektor **b** preko SVD.
5. Ocenimo pet intrinzičnih parametrov z uporabo vektorja **b** in enačb (3.18).

6. Z enačbami (3.19) izračunamo rotacijske matrike in translacijske vektorje za vse zajete slike.
7. Ocenimo koeficiente radialne distorzije.
8. Popravimo vse kalibracijske parametre in koeficienta radialne distorzije po enačbi (3.28).

3.2.8 Rezultati kalibracije kamere

Da bi lahko za preizkus algoritmov uporabili pametni telefon (Samsung Galaxy S4), smo za njegovo hrbtno kamero najprej morali pridobiti kalibracijske (intrinzične in ekstrinzične) parametre. Kamera telefona Galaxy S4 je sestavljena iz CMOS slikovnega senzorja Sony IMX091PQ, ki ima ločljivost 13,25 M in leče, ki ima goriščno razdaljo $f = 31$ mm (ekvivalentno za 35 mm senzor) in zaslonko $f/2,2$ (ang. lens aperture). Velikost senzorja je $1/3.06$ oz. $4,69 \times 3,52$ mm (diagonala je enaka 5,87 mm). Pri maksimalnem efektivnem številu slikovnih točk, to je pri ločljivosti 4208×3120 , je posamezna slikovna točka velika $1,12 \mu\text{m}$. Faktor izreza (ang. crop factor) je enak $\frac{diag_{35}}{diag} = \frac{43,3}{5,87} = \frac{31}{4,2} = 7,38$.

Kalibracijo smo izvedli na pametnem telefonu Android z uporabo knjižnice BoofCV [142] ter za namen primerjave hkrati tudi v okolju Matlab z uporabo »Computer Vision System Toolbox« [143]. Kalibracija kamere na pametnem telefonu Android temelji na delu Zhang [138], ki smo ga v celoti opisali. Prav tako tudi kalibracija v okolju Matlab, le da je v tem primeru algoritem še nekoliko izpopolnjen. Na sliki 3.5 vidimo primer zajete slike, na kateri se nahaja kalibracijski objekt (šahovnica). Z rdečimi pikami so označeni detektirani vogali. Slike smo zajeli s pametnim telefonom Samsung Galaxy S4 pri ločljivosti 320×240 slikovnih točk.

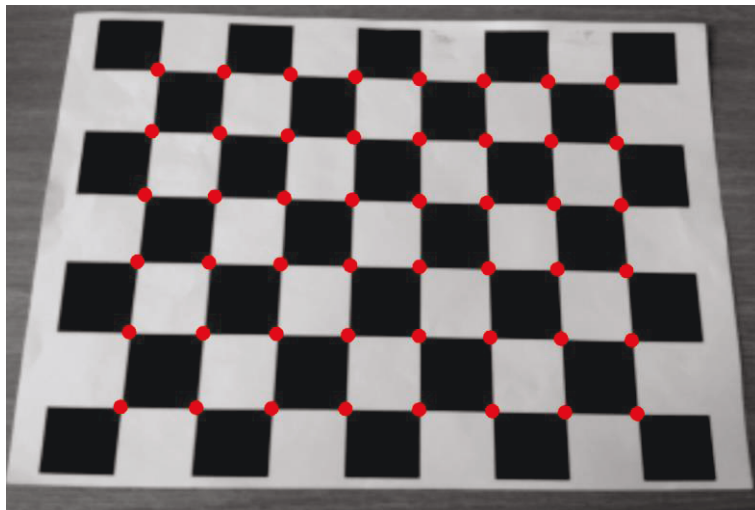
Kalibracijski objekt (šahovnico) smo slikali pod različnimi koti, kot je prikazano na sliki 3.6.

Kot rezultat kalibracije kamere na pametnem telefonu (Samsung Galaxy S4) smo dobili naslednje intrinzične parametre:

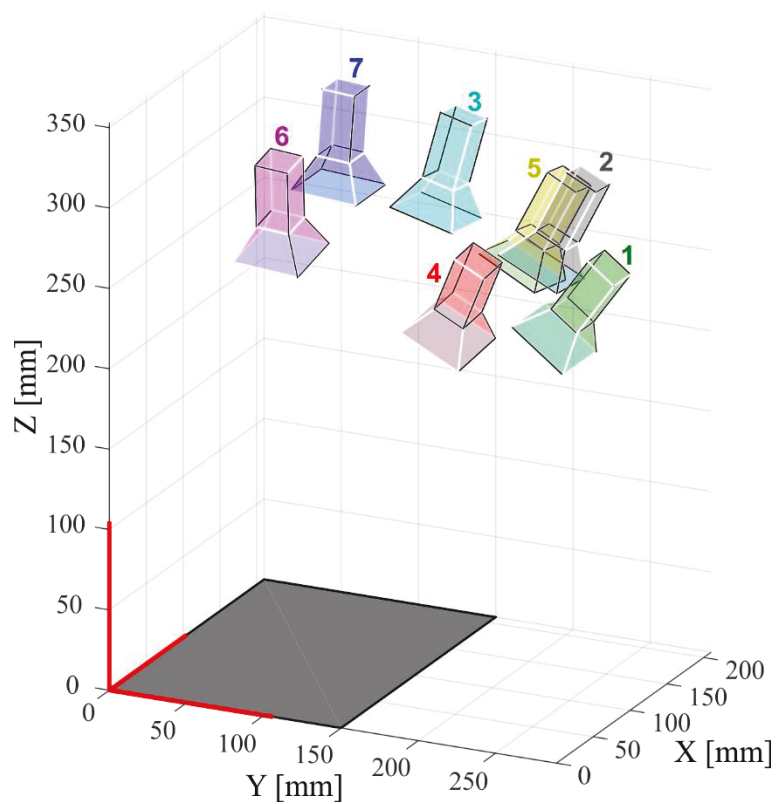
$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 294,24 & 0 & 160,55 \\ 0 & 293,84 & 124,35 \\ 0 & 0 & 1 \end{bmatrix}.$$

V okolju Matlab pa smo hkrati dobili naslednje intrinzične parametre:

$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 292,74 & 0 & 162,44 \\ 0 & 292,86 & 124,39 \\ 0 & 0 & 1 \end{bmatrix}.$$



Slika 3.5: Detekcija vogalov pri kalibraciji kamere



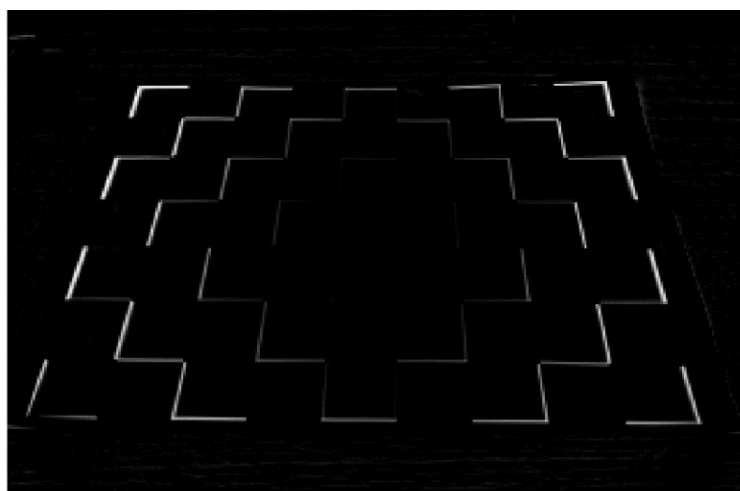
Slika 3.6: Pri kalibraciji smo šahovnico slikali pod različnimi koti

Če iz specifikacij kamere preberemo goriščno razdaljo, ki je enaka 4,2 mm in izračunamo gostoto slikovnih točk pri ločljivosti 320×240 ($s_x = 68,23$ slikovnih točk na mm in $s_y = 68,18$ slikovnih točk na mm), potem lahko izračunamo goriščno razdaljo v slikovnih točkah: $\alpha = 286,57$ slikovnih točk in $\beta = 286,36$ slikovnih točk. Če ta rezultat primerjamo z rezultatom, ki smo ga dobili pri kalibracijah na pametnem telefonu in v okolju Matlab, vidimo, da je napaka v obeh primerih manjša od 3 %. Zavedati se moramo, da podana goriščna razdalja (4,2 mm) ni povsem točna.

Parameter »optično središče« $[u_0, v_0]$ nam pove, za koliko je premaknjeno središče slike glede na središče leče. Intrinzični parameter γ (popačenje slikovne točke) je v obeh primerih 0. To pomeni, da je v algoritmu predpostavljeno, da je posamezna slikovna točka pravokotna (kot med osema je 90°).

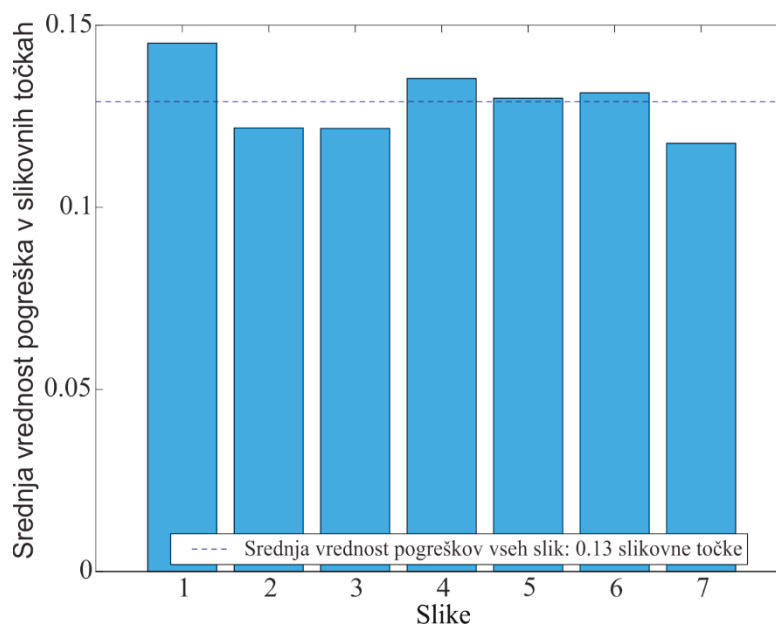
V okolju Matlab smo izračunali tako radialno kot tangencialno komponento distorzije. Za tangencialno distorzijo se je izkazalo, da je dovolj majhna, da jo lahko zanemarimo (koeficienti so reda 10^{-3}). Zato pri kalibraciji na pametnem telefonu upoštevamo le radialno distorzijo, ki je bolj izrazita. Na pametnem telefonu smo dobili naslednji vrednosti koeficientov distorzije: $k_1 = 0,157$ in $k_2 = -0,424$ ter v okolju Matlab: $k_1 = 0,1482$ in $k_2 = -0,3750$.

Na sliki 3.7 je prikazana absolutna razlika dveh slik, pri čemer je prva popačena, druga pa nepopačena (upoštevana je radialna distorzija). Opazimo lahko, da je pri večjem radiju distorzija bolj opazna.

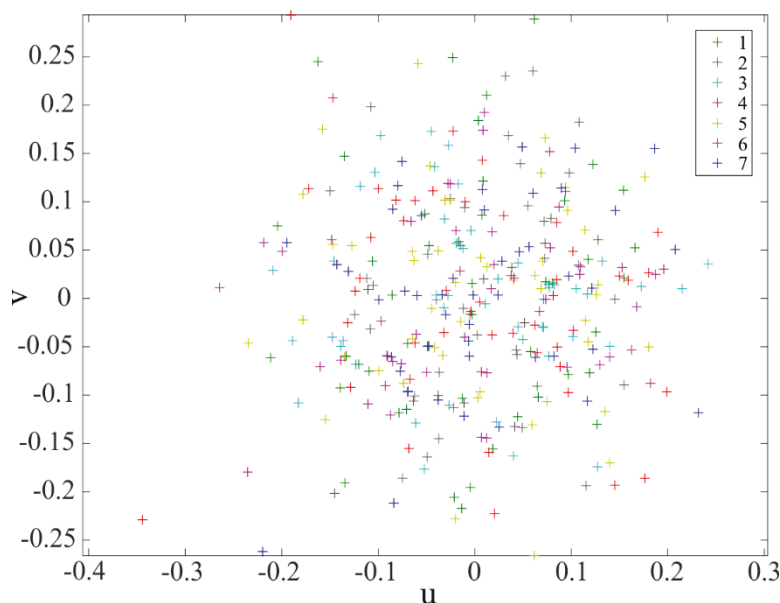


Slika 3.7: Absolutna razlika med popačeno in nepopačeno sliko

Kalibracijo ovrednotimo z oceno re-projekcijskega pogreška (ta je enak evklidski razdalji med detektiranimi točkami in točkami, ki jih dobimo tako, da točke modela šahovnice preslikamo na slikovno ravnino z upoštevanjem intrinzičnih in ekstrinzičnih parametrov). Pri kalibraciji kamere na pametnem telefonu smo dobili povprečno vrednost re-projekcijskega pogreška 0,1458, v okolju Matlab pa 0,1290. Pogrešek, ki ga dobimo pri kalibraciji v okolju Matlab, je prikazan na slikah 3.8 in 3.9.



Slika 3.8: Srednja vrednost pogreška pri re-projekciji



Slika 3.9: Pogrešek pri re-projekciji v slikovnih točkah (za vseh 7 slik)

Iz slike 3.9 lahko sklepamo, kako natančna je kalibracija, saj bližje kot so točke koordinatnemu izhodišču, boljša je kalibracija.

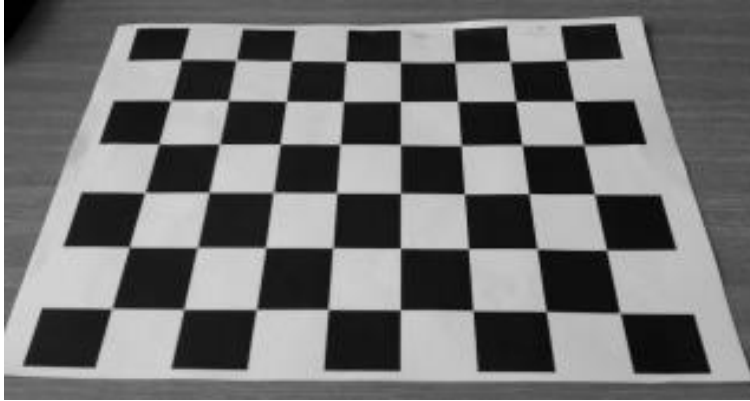
Pri izbrani sliki 3.10 lahko s kalibracijskim algoritmom izračunamo rotacijsko matriko in translacijski vektor (ekstrinzični parametri) med šahovnico in kamero.

Rotacijska matrika:

$$\mathbf{R} = \begin{bmatrix} 0,9998 & -0,0132 & 0,0122 \\ 0,0028 & 0,7832 & 0,6217 \\ -0,0177 & -0,6216 & 0,7832 \end{bmatrix}.$$

Translacijski vektor:

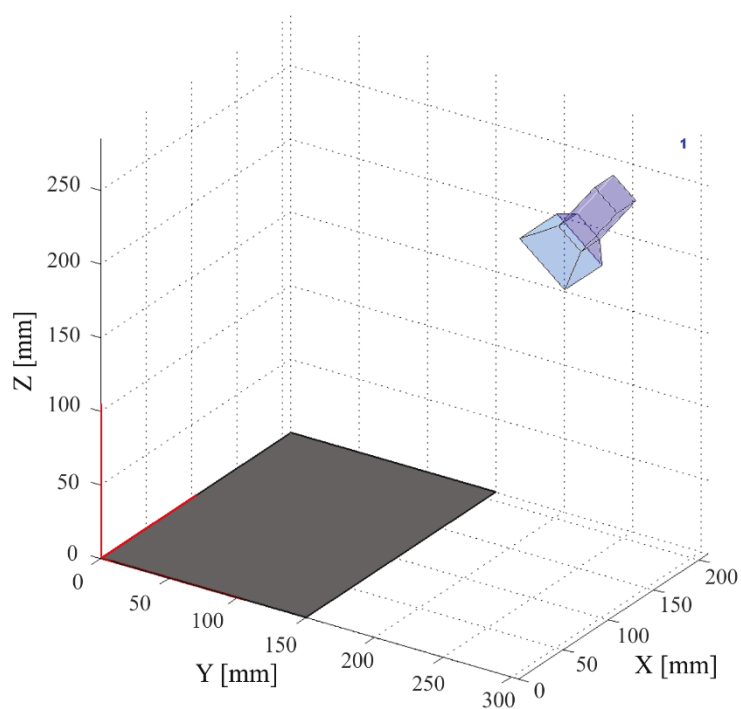
$$\mathbf{T} = [-112,1690 \quad -59,6918 \quad 384,2370]^T.$$



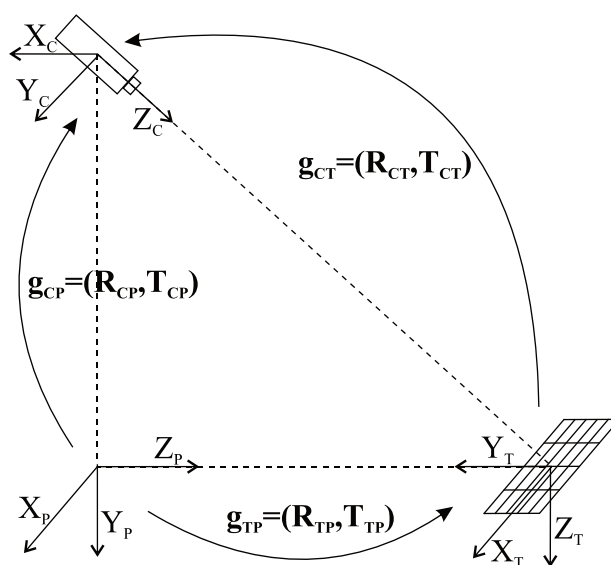
Slika 3.10: Primer slike, za katero smo izračunali ekstrinzične parametre

Na sliki 3.11 vidimo vizualizacijo ekstrinzičnih parametrov, ki jih dobimo pri zajemu slike 3.10.

Za namen vizualne odometrije potrebujemo opis togega premika med K.S. podlage in K.S. kamere: $\mathbf{g}_{CP} = (\mathbf{R}_{CP}, \mathbf{T}_{CP})$. Ta togi premik lahko izračunamo tako, da upoštevamo togi premik med šahovnico in kamero ($\mathbf{g}_{CT} = (\mathbf{R}_{CT}, \mathbf{T}_{CT})$) ter togi premik med K.S. podlage in šahovnice ($\mathbf{g}_{TP} = (\mathbf{R}_{TP}, \mathbf{T}_{TP})$): $\mathbf{g}_{CP} = \mathbf{g}_{CT} \mathbf{g}_{TP}$ (slika 3.12).



Slika 3.11: Vizualizacija ekstrinzičnih parametrov



Slika 3.12: Transformacije med koordinatnimi sistemi podlage (P), šahovnice (T) in kamere (C)

3.3 Optični tok

Optični tok je način opisovanja navideznega gibanja površine, objektov in robov, ki se pojavi zaradi relativnega gibanja med opazovalcem in sceno. Optični tok je predstavljen kot vektorsko polje, katerega komponente so vektorji [144].

Optični tok se je uveljavil na področju avtonomnih mobilnih sistemov za namen detekcije gibanja ter v splošnem na področju strojnega vida za namen segmentacije objektov itd.

Namen metod optičnega toka (Kanade-Lucas-Tomasi, Horn-Schunck itd.) je določiti gibanje med dvema slikama, ki sta zajeti v času t in $t + \Delta t$. Te metode optičnega toka se imenujejo diferencialne metode, saj temeljijo na lokalni aproksimaciji signala slike s Taylorjevo vrsto. Za optični tok mora veljati, da je neodvisen od spreminjanja osvetlitve ter gibanja nepomembnih objektov (npr. gibanje senc).

Vzemimo 2-D primer in predpostavimo, da so se slikovne točke slike I s koordinatami (x, y) in intenziteto $I(x, y, t)$ po času Δt premaknile za Δx in Δy , pri čemer velja:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t). \quad (3.30)$$

Če predpostavimo, da so premiki slikovnih točk majhni, lahko z uporabo Taylorjeve vrste zapišemo:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t. \quad (3.31)$$

Iz zgornjih dveh enačb sledi:

$$\frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t = 0 \quad (3.32)$$

in

$$\frac{\partial I}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} \frac{\Delta t}{\Delta t} = 0, \quad (3.33)$$

kar lahko dalje zapišemo kot:

$$\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} = 0, \quad (3.34)$$

kjer sta V_x in V_y komponenti vektorja hitrosti navideznega premika ali optičnega toka za slikovno točko z intenziteto $I(x, y, t)$, $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ in $\frac{\partial I}{\partial t}$ pa so parcialni odvodi slike pri (x, y, t) . Parcialne odvode $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ in $\frac{\partial I}{\partial t}$ lahko zapišemo tudi kot I_x , I_y in I_t :

$$I_x V_x + I_y V_y = -I_t \quad (3.35)$$

ali tudi kot:

$$\nabla I^T \cdot \mathbf{V} = -I_t, \quad (3.36)$$

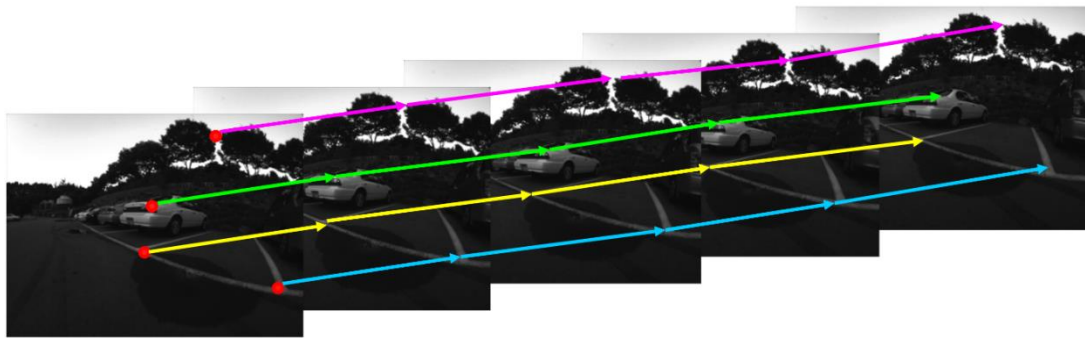
kjer je $\mathbf{V} = [V_x \ V_y]^T$. Enačba (3.36) ima dve neznanki, ki pa jih ne moremo dobiti z reševanjem te enačbe. Če predpostavimo, da se slikovne točke v okolici opazovane točke obnašajo enako kot ta, lahko zapišemo enačbo (3.35) za vse slikovne točke, ki se nahajajo znotraj opazovanega okna. Tako dobimo predoločen sistem, ki bi ga lahko rešili po metodi najmanjših kvadratov.

3.3.1 Kanade-Lucas-Tomasi sledilnik značilnih točk

Kanade-Lucas-Tomasi (KLT) algoritem temelji na treh člankih, in sicer v prvem sta Lucas in Kanade [145] razvila idejo o lokalnem iskanju sorodnih točk z uporabo gradientov uteženih z aproksimacijo drugega odvoda slike. V drugem članku [146] sta Tomasi in Kanade dopolnila idejo iz članka [145] tako, da sta predlagala način, kako izbrati značilne točke, ki izboljšajo sledenje skozi zaporedne slike. V tretjem članku sta Shi in Tomasi [147] predlagala način, kako dodatno preveriti, ali se sledenje značilnim točkam izvaja pravilno.

Problem, ki ga rešuje algoritem KLT, lahko definiramo kot iskanje premika med dvema zaporednima slikama, ki ju označimo z I in J . Ti dve sliki sta sivinski sliki velikosti $N_x \times N_y$ slikovnih točk. Naj bo $I(x, y)$ intenziteta sivinske slike I v točki (x, y) . Ideja algoritma KLT je enaka, kot smo jo predstavili pri optičnem toku. Torej, imamo točko $\mathbf{u} = (u_x, u_y)$ na sliki I , cilj algoritma pa je najti točko $\mathbf{v}$ na sliki J , pri čemer sta vrednosti $I(\mathbf{u})$ in $J(\mathbf{v})$ podobni ($\mathbf{v} = \mathbf{u} + \mathbf{d} = (u_x + d_x, u_y + d_y)$). Optični tok v točki $\mathbf{u}$ je $\mathbf{d} = [d_x \ d_y]^T$.

Kot vidimo na sliki 3.13, lahko z uporabo algoritma KLT sledimo istim značilnim točkam skozi sekvenco zaporednih slik. Sledenje posamezni točki se konča v primeru, da ta »izgine«. To se zgodi v primeru, ko opazovana točka ni več na sliki ali pa v primeru, ko sta okolici opazovane točke na slikah I in J preveč različni (cenilka $\varepsilon(\mathbf{d})$ je večja od določenega praga).

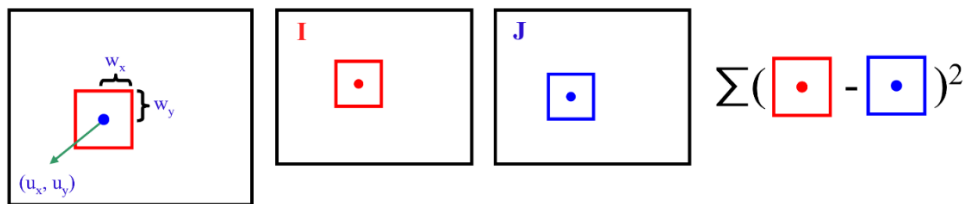


Slika 3.13: Sledenje značilnim točkam z uporabo algoritma KLT skozi sekvenco slik

Cenilko $\varepsilon(\mathbf{d})$ lahko definiramo kot:

$$\varepsilon(\mathbf{d}) = \varepsilon(d_x, d_y) = \sum_{x=u_x-\omega_x}^{u_x+\omega_x} \sum_{y=u_y-\omega_y}^{u_y+\omega_y} (I(x, y) - J(x + d_x, y + d_y))^2, \quad (3.37)$$

pri čemer sta ω_x in ω_y parametra, ki določata velikost integracijskega okna: $(2\omega_x + 1) \times (2\omega_y + 1)$ (slika 3.14). Tipične vrednosti za ω_x in ω_y so 2~7 slikovnih točk.



Slika 3.14: Ocena napake pri iskanju premika med slikama I in J

3.3.2 Piramidni KLT algoritem

Klasični KLT algoritem deluje dobro le v primeru, da so premiki slikovnih točk majhni. Pri večjih premikih pa potrebujemo piramidno implementacijo tega algoritma [148, 149].

Piramida je večslojni prikaz iste slike pri različnih resolucijah (oz. množica slik različne velikosti, ki so pridobljene iz referenčne slike). Posamezne sloje piramide dobimo z večkratnim glajenjem in podvzorčenjem (ang. subsampling). V algoritmu KLT lahko uporabimo Gaussovo nizkopasovno piramido, ki temelji na Gaussovem glajenju slike (ang. Gaussian smoothing). Matematično gledano, glajenje predstavlja konvolucijo slike z Gaussovo funkcijo. Gaussovo glajenje deluje kot nizkopasovni

filter (Fourierova transformacija Gaussove funkcije je prav tako Gaussova funkcija) in zato iz slike izloči visokofrekvenčne komponente slike.

Če imamo dano sliko I potem ničti nivo piramide označimo z I^0 , pri čemer velja, da je $I^0 = I$. Piramidno predstavitev določene slike gradimo rekurzivno:

$$I^0 \rightarrow I^1 \rightarrow I^2 \rightarrow I^3 \rightarrow I^{L_m} \dots (L_m: 2 \sim 4), \quad (3.38)$$

pri čemer največji indeks L_m predstavlja najmanjšo sliko v piramidi (najmanjša resolucija slike). Primer piramide s štirimi nivoji ($L_m = 3$) vidimo na sliki 3.15.

Kot smo že omenili, želimo z algoritmom KLT poiskati točko $\mathbf{v} = \mathbf{u} + \mathbf{d}$, ki ustreza točki $\mathbf{u}$ na sliki I . Točka $\mathbf{u}$ pa je v piramidi glede na nivo določena kot: $\mathbf{u}^L = \mathbf{u}/2^L$ (ker je vsaka naslednja slika v piramidi manjša za dvakrat od predhodne).

Pri piramidnem KLT algoritmu za vsak nivo piramide izračunamo razdaljo $\mathbf{d}^{L_m}$, pri čemer upoštevamo naslednji vrstni red:

- $\mathbf{d}^{L_m}$ izračunamo za piramidni nivo L_m ,
- izračunamo $\mathbf{d}^{L_m-1}$ za nivo L_m-1 , pri čemer upoštevamo predhodni $\mathbf{d}^{L_m}$,
- ta postopek nadaljujemo do nivoja 0.



Slika 3.15: Primer piramide s štirimi nivoji ($L_m = 3$)

Naj bo $\mathbf{g}^L = [g_x^L \ g_y^L]^T$ ocena začetnega premika (oz. translacija točke in njene okolice do sorodne točke in njene okolice na drugi sliki J pri izbranem piramidnem nivoju L).

Napako zaradi neujemanja slikovnih točk med dvema slikama (pri upoštevanju premika $\mathbf{d}^L = [d_x^L \ d_y^L]$) na nivoju L zapišemo kot:

$$E^L(\mathbf{d}^L) = \varepsilon^L(d_x^L, d_y^L) = \sum_{x=u_x^L-\omega_x}^{u_x^L+\omega_x} \sum_{y=u_y^L-\omega_y}^{u_y^L+\omega_y} (I^L(x, y) - J^L(x + g_x^L + d_x^L, y + g_y^L + d_y^L))^2. \quad (3.39)$$

Velikost okna $(2\omega_x + 1) \times (2\omega_y + 1)$ je enaka na vseh nivojih piramide.

Zaradi upoštevanja ocene premika $\mathbf{g}^L$ je razdalja $\mathbf{d}^L$ na vseh nivojih majhna, kar pomeni, da jo lahko na posameznem nivoju izračunamo s klasičnim KLT algoritmom. Za vsak naslednji nivo $L - 1$ v piramidi moramo izračunati novo oceno začetnega premika kot:

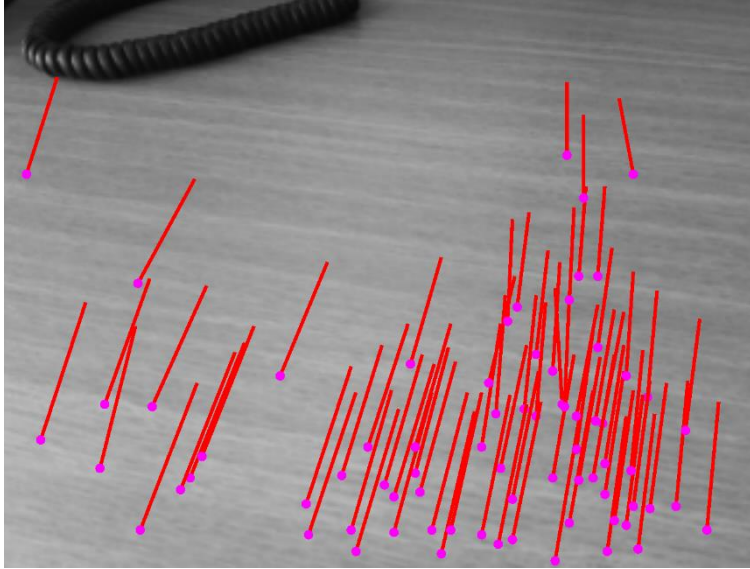
$$\mathbf{g}^{L-1} = 2(\mathbf{g}^L + \mathbf{d}^L). \quad (3.40)$$

Za najnižji nivo L_m je ocena začetnega premika enaka $\mathbf{g}^{L_m} = [0 \ 0]^T$. Končni optični tok dobimo kot:

$$\mathbf{d} = \sum_{L=0}^{L_m} 2^L \mathbf{d}^L. \quad (3.41)$$

Prednost piramidnega KLT algoritma je v tem, da je lahko maksimalni vektor premika $\mathbf{d}$ večji za faktor $(2^{L_m+1} - 1)$ kot pri običajnem KLT algoritmu (pri $L_m = 3$ je lahko premik 15-krat večji).

Na sliki 3.16 vidimo prikaz optičnega toka, ki ga dobimo z uporabo piramidnega KLT algoritma na pametnem telefonu Android.



Slika 3.16: Optični tok, ki ga dobimo z uporabo piramidnega KLT algoritma na pametnem telefonu

3.3.3 Standardni KLT algoritem

Z uporabo standardnega KLT algoritma želimo za vsak nivo piramide poiskati tako razdaljo $\mathbf{d}^L$, ki minimizira funkcijo ε^L . Za vse točke v oknu $\forall(x, y) \in [p_x - \omega_x, p_x + \omega_x] \times [p_y - \omega_y, p_y + \omega_y]$ definiramo dve novi sliki:

$$A(x, y) = I^L(x, y) \text{ in } B(x, y) = J^L(x + g_x^L, y + g_y^L). \quad (3.42)$$

Označimo vektor premika in vektor točke kot:

$$\mathbf{v} = [v_x \ v_y]^T = \mathbf{d}^L \quad (3.43)$$

$$\mathbf{p} = [p_x \ p_y]^T = \mathbf{u}^L. \quad (3.44)$$

Za posamezen nivo v piramidi izračunamo razdaljo $\mathbf{v}$, ki minimizira funkcijo:

$$\varepsilon(\mathbf{v}) = \varepsilon(v_x, v_y) = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} (A(x, y) - B(x + v_x, y + v_y))^2. \quad (3.45)$$

Da dobimo minimum funkcije (3.45), izračunamo njen odvod in ga enačimo z nič:

$$\frac{\partial \varepsilon(\mathbf{v})}{\partial \mathbf{v}} \big|_{\mathbf{v}=\mathbf{v}_{opt}} = [0 \ 0] \quad (3.46)$$

$$\frac{\partial \varepsilon(\mathbf{v})}{\partial \mathbf{v}} = -2 \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} (A(x, y) - B(x + v_x, y + v_y)) \cdot \left[\frac{\partial B}{\partial x} \ \frac{\partial B}{\partial y} \right]. \quad (3.47)$$

Če $B(x + v_x, y + v_y)$ zapišemo s Taylorjevo vrsto okoli točke $\mathbf{v} = [0 \ 0]^T$, dobimo:

$$\frac{\partial \varepsilon(\mathbf{v})}{\partial \mathbf{v}} \approx -2 \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \left(A(x, y) - B(x, y) - \left[\frac{\partial B}{\partial x} \ \frac{\partial B}{\partial y} \right] \mathbf{v} \right) \cdot \left[\frac{\partial B}{\partial x} \ \frac{\partial B}{\partial y} \right], \quad (3.48)$$

kjer je $\delta I(x, y) = A(x, y) - B(x, y)$ razlika dveh slikovnih okvirjev in $\nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix} =$

$\left[\frac{\partial B}{\partial x} \ \frac{\partial B}{\partial y} \right]^T$ gradient slike. Parcialne odvode izračunamo kot:

$$I_x(x, y) = \frac{\partial A(x, y)}{\partial x} = \frac{A(x+1, y) - A(x-1, y)}{2}, \quad (3.49)$$

$$I_y(x, y) = \frac{\partial A(x, y)}{\partial y} = \frac{A(x, y+1) - A(x, y-1)}{2}. \quad (3.50)$$

Iz enačbe (3.48) sledi:

$$\frac{1}{2} \frac{\partial \varepsilon(\mathbf{v})}{\partial \mathbf{v}} \approx \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} (\nabla I^T \mathbf{v} - \delta I) \nabla I^T, \quad (3.51)$$

$$\frac{1}{2} \left[\frac{\partial \varepsilon(\mathbf{v})}{\partial \mathbf{v}} \right]^T \approx \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \left(\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \mathbf{v} - \begin{bmatrix} \delta I I_x \\ \delta I I_y \end{bmatrix} \right) \quad (3.52)$$

in nadalje:

$$\mathbf{G} = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \quad (3.53)$$

$$\mathbf{b} = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} \delta I I_x \\ \delta I I_y \end{bmatrix}, \quad (3.54)$$

$$\frac{1}{2} \left[\frac{\partial \varepsilon(\mathbf{v})}{\partial \mathbf{v}} \right]^T \approx \mathbf{G}\mathbf{v} - \mathbf{b}, \quad (3.55)$$

$$\mathbf{v}_{opt} = \mathbf{G}^{-1} \mathbf{b}. \quad (3.56)$$

Z enačbo (3.56) dobimo optimalno razdaljo $\mathbf{v}_{opt}$, ki minimizira funkcijo $\varepsilon(\mathbf{v})$. Za matriko $\mathbf{G}$ mora veljati, da je obrnljiva.

Ker je klasični Lukas-Kanade algoritem primeren le za majhne premike slikovnih točk (zaradi aproksimacije s Taylorjevo vrsto prvega reda), potrebujemo iterativno verzijo tega algoritma [148]. Zato že zapisane enačbe nekoliko preuredimo in zapišemo začetno oceno premika, ki jo dobimo v predhodnem koraku kot: $\mathbf{v}^{k-1} = [v_x^{k-1} v_y^{k-1}]^T$, pri čemer je k iteracijski indeks. Z B_k označimo novo premaknjeno sliko glede na $\mathbf{v}^{k-1}$ za $\forall(x, y) \in [p_x - \omega_x, p_x + \omega_x] \times [p_y - \omega_y, p_y + \omega_y]$:

$$B_k(x, y) = B(x + v_x^{k-1}, y + v_y^{k-1}). \quad (3.57)$$

Cilj je izračunati $\boldsymbol{\eta}^k = [\eta_x^k \eta_y^k]$, ki minimizira:

$$\varepsilon^k(\boldsymbol{\eta}^k) = \varepsilon(\eta_x^k, \eta_y^k) = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} (A(x, y) - B_k(x + \eta_x^k, y + \eta_y^k))^2. \quad (3.58)$$

Za vsako iteracijo izračunamo optični tok kot: $\boldsymbol{\eta}^k = \mathbf{G}^{-1} \mathbf{b}_k$, kjer se za vsako iteracijo spremenita $\mathbf{b}_k$ in δI_k kot:

$$\mathbf{b}_k = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} \delta I_k(x, y) I_x(x, y) \\ \delta I_k(x, y) I_y(x, y) \end{bmatrix},$$

$$\delta I_k(x, y) = A(x, y) - B_k(x, y). \quad (3.59)$$

Parcialna odvoda $I_x(x, y)$ in $I_y(x, y)$ se izračunata le enkrat na začetku iteracijskega postopka. Matrika $\mathbf{G}$ velikosti 2×2 se prav tako ne spreminja znotraj iteracijske zanke. Ko izračunamo $\boldsymbol{\eta}^k$, lahko določimo celoten premik med slikama kot:

$$\mathbf{v}^k = \mathbf{v}^{k-1} + \boldsymbol{\eta}^k. \quad (3.60)$$

Iteracijski postopek se ponavlja toliko časa, dokler ne pade vrednost $\boldsymbol{\eta}^k$ pod nek določen prag ali dokler ni doseženo maksimalno število iteracij (običajno je pet iteracij že dovolj, da dosežemo konvergenco). Začetni premik med slikama je enak $\mathbf{v}^0 = [0 \ 0]^T$. Če predpostavimo, da za konvergenco potrebujemo K iteracij, potem lahko zapišemo optični tok za vsak nivo piramide ($L - 1, L - 2, \dots, 0$) kot:

$$\mathbf{v} = \mathbf{d}^L = \mathbf{v}^k = \sum_{k=1}^K \boldsymbol{\eta}^k. \quad (3.61)$$

Povzetek postopka izračuna optičnega toka:

Cilj: Naj bo $\mathbf{u}$ točka na sliki I . Iščemo točko $\mathbf{v}$ na sliki J , ki ustreza točki $\mathbf{u}$.

Zgradimo piramidno predstavitev slik I in J : $\{I^L\}_{L=0,\dots,L_m}$ in $\{J^L\}_{L=0,\dots,L_m}$

Določimo začetno vrednost premika v piramidi: $\mathbf{g}^{L_m} = [g_x^{L_m} \ g_y^{L_m}]^T = [0 \ 0]^T$

for $L = L_m : -1 : 0$

Lokacija točk $\mathbf{u}$ na sliki I^L : $\mathbf{u}^L = [p_x \ p_y]^T = \mathbf{u} / 2^L$

Odvod slike I^L po x : $I_x(x, y) = \frac{I^L(x+1, y) - I^L(x-1, y)}{2}$

Odvod slike I^L po y : $I_y(x, y) = \frac{I^L(x, y+1) - I^L(x, y-1)}{2}$

Gradientna matrika: $\mathbf{G} = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} I_x^2(x, y) & I_x(x, y)I_y(x, y) \\ I_x(x, y)I_y(x, y) & I_y^2(x, y) \end{bmatrix}$

Inicializacija premika: $\mathbf{v}^0 = [0 \ 0]^T$

for $k = 1 : K$ (ali dokler ni $\|\boldsymbol{\eta}^k\| < \text{določen prag}$)

Razlika slik: $\delta I_k(x, y) = I^L(x, y) - J^L(x + g_x^L + v_x^{k-1}, y + g_y^L + v_y^{k-1})$

Vektor neujemanja slik: $\mathbf{b}_k = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} \delta I_k(x, y) I_x(x, y) \\ \delta I_k(x, y) I_y(x, y) \end{bmatrix}$

Optični tok (Lucas-Kanade): $\boldsymbol{\eta}^k = \mathbf{G}^{-1} \mathbf{b}_k$

Premik v naslednji iteraciji: $\mathbf{v}^k = \mathbf{v}^{k-1} + \boldsymbol{\eta}^k$

end for

Končni optični tok na nivoju L : $\mathbf{d}^L = \mathbf{v}^k$

Začetni premik za $L - 1$: $\mathbf{g}^{L-1} = [g_x^{L-1} \ g_y^{L-1}]^T = 2(\mathbf{g}^L + \mathbf{d}^L)$

end for

Končni vektor optičnega toka: $\mathbf{d} = \mathbf{g}^0 + \mathbf{d}^0$

Lokacija točk na sliki J : $\mathbf{v} = \mathbf{u} + \mathbf{d}$

3.3.4 Iskanje značilnih točk

Prvi korak, ki ga moramo izvesti pred uporabo KLT algoritma za določitev optičnega toka, je detekcija značilnih oz. zanimivih točk. Čeprav lahko v kombinaciji s KLT algoritmom uporabljamo poljubne značilne točke, sta Shi in Tomasi [147] predlagala

uporabo t.i. Shi-Tomasi detektorja vogalov, ki na sliki izbere točke, ki jim je lahko slediti skozi sekvenco slik.

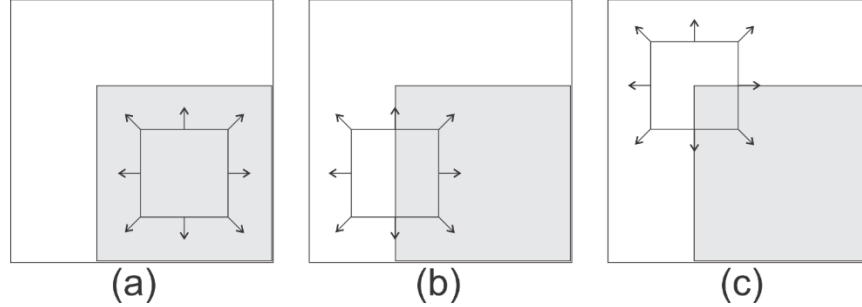
Detekcija vogalov na sliki je pristop, ki se uporablja na področju računalniškega vida za namen pridobivanja določenih vrst zanimivih točk (ang. interest points). Algoritmi za detekcijo vogalov se pogosto uporabljajo pri detekciji gibanja, poravnavi slik, sledenju objektov, združevanju slik, prepoznavi objektov na sliki itd.

Vogal (ang. corner) lahko definiramo kot presečišče dveh robov ali tudi kot točko, ki ji pripadata dva dominantna robova v različnih smereh. Za vogale ter v splošnem za značilne točke naj bi veljalo, da imajo dobro definiran položaj na sliki in so robustno (ponovljivo) detektirane. To pomeni, da je zanimiva točka lahko vogal ali pa tudi npr. točka za katero velja, da je v njej lokalni minimum ali maksimum intenzitete slike. Tako v praksi večino detektorjev vogalov zaznava zanimive točke, ki ne ležijo nujno na vogalu nekega predmeta, kot bi si lahko predstavljali (npr. z detektorjem vogalov lahko detektiramo »vogal« tudi sredi mize). Kvaliteto detektorja vogalov lahko ovrednotimo s ponovljivostjo detekcije iste točke oz. vogala na več slikah iste scene pri različni osvetljenosti, vključeni translaciji, rotaciji in drugih oblikah transformacij. Najbolj enostaven način za detekcijo vogalov je z izračunom korelacije, vendar je ta način računsko potraten ter neoptimalen.

Enega izmed prvih detektorjev vogalov je predstavil avtor Moravec [150, 151], po katerem se ta detektor tudi imenuje. Iz tega detektorja so Harris in drugi [151, 152] razvili t.i. Harrisov detektor vogalov, ki je bil prvi, ki se je uveljavil tudi v praksi. Shi in Tomasi [147] pa sta Harrisov detektor še nekoliko izpopolnila in ga poimenovala Shi-Tomasijev detektor vogalov. Ker ta detektor temelji na detektorjema Moravec in Harris, bomo v nadaljevanju najprej predstavili ta dva detektorja.

Pri detektorju Moravec je vogal definiran kot točka, za katero velja, da njeno lokalno območje (ang. patch) po intenziteti ni podobno lokalnim območjem sosednjih točk (intenziteta slike se spreminja v vseh smereh od te točke [151]). Algoritem vključuje preverjanje, ali je prisoten vogal ali ne za vsako slikovno točko posebej. Na sliki 3.17 vidimo tri primere, pri čemer v primeru (a) ne detektiramo vogala, saj se intenziteta slike ne spreminja v nobeni smeri, v primeru (b) detektiramo rob, saj se intenziteta v smeri pravokotno na rob spreminja, v smeri roba pa so spremembe majhne (če okno premaknemo vzdolž roba, se to le malo razlikuje od referenčnega okna), v

primeru (c) pa detektiramo vogal, saj se intenziteta spreminja v vseh smereh (vsa sosednja okna se po intenziteti razlikujejo od referenčnega okna).



Slika 3.17: (a) »ravno« področje – v okolici ni sprememb intenzitete, (b) rob – ni sprememb intenzitete v smeri roba, (c) vogal – velike spremembe intenzitete v vseh smereh

Podobnost med lokalnimi območji (ang. patch) slikovnih točk je merjena z vsoto kvadratov razlik (ang. sum of squared differences - SSD), pri čemer nižja vrednost pomeni večjo podobnost. V primeru, da slikovna točka leži na vogalu, potem se njena sosednja območja razlikujejo po intenziteti v vseh smereh. Jakost vogala je definirana kot minimalni SSD med lokalnim območjem točke in sosednjimi področji. Glavni problem detektorja Moravec je, da ni izotropen (ni enoličen v vseh smereh). To pomeni, da bo v primeru, ko je prisoten rob, ki ni v smeri sosednjih točk, najmanjša vrednost SSD velika in bo posledično algoritem to točko detektiral kot vogal.

Harris in drugi [152] so Moravec detektor vogalov izboljšali tako, da so neposredno upoštevali še parcialne odvode SSD-ja, ki pripada vogalu glede na smer.

Če imamo določeno lokalno območje okoli točke (x, y) in ga premaknemo za $(\Delta x, \Delta y)$, potem je utežena vsota kvadratov razlik (SSD) med tema dvema lokalnima območjema podana kot:

$$S(\Delta x, \Delta y) = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} w(x, y) (I(x + \Delta x, y + \Delta y) - I(x, y))^2, \quad (3.62)$$

kjer lahko $I(x + \Delta x, y + \Delta y)$ aproksimiramo s Taylorjevo vrsto ($w(x, y)$ predstavlja uteži dobljene z Gaussovim oknom). Naj bosta I_x in I_y parcialna odvoda od I , da velja:

$$I(x + \Delta x, y + \Delta y) \approx I(x, y) + I_x(x, y)\Delta x + I_y(x, y)\Delta y. \quad (3.63)$$

V naslednjem koraku lahko zapišemo aproksimacijo za uteženo vsoto kvadratov razlik kot:

$$S(\Delta x, \Delta y) \approx \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} w(x, y) (I_x(x, y)\Delta x + I_y(x, y)\Delta y)^2, \quad (3.64)$$

kar lahko nadalje zapišemo še v matrični obliki kot:

$$S(\Delta x, \Delta y) \approx [\Delta x, \Delta y] \mathbf{G} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \quad (3.65)$$

pri čemer je $\mathbf{G}$ strukturni tenzor (ang. second moment matrix):

$$\mathbf{G} = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}. \quad (3.66)$$

Vogal je označen z velikimi spremembami vrednosti S v smeri x in y . Z analizo lastnih vrednosti matrike $\mathbf{G}$, je ta oznaka lahko izražena na naslednji način; matrika $\mathbf{G}$ naj bi imela dve veliki lastni vrednosti za vsak vogal. Glede na velikost lastnih vrednosti (λ_1, λ_2), lahko sklepamo naslednje:

1. Če je $\lambda_1 \approx 0$ in $\lambda_2 \approx 0$, potem slikovna točka (x, y) ni vogal.
2. Če je $\lambda_1 \approx 0$ in λ_2 neka velika pozitivna vrednost, to pomeni, da smo našli rob.
3. Če sta λ_1 in λ_2 veliki pozitivni vrednosti to pomeni, da smo našli vogal.

Shi-Tomasijev detektor robov v celoti temelji na Harrisovem detektorju robov. Glavna razlika med njima je v izbiri »kriterija« (za določitev vogalov), ki dejansko tudi naredi Shi-Tomasijev detektor boljši od originala. Pri Harrisovem detektorju je kriterij (ki je računsko manj zahteven kot izračun lastnih vrednosti) določen kot:

$$R = \det(\mathbf{G}) - k(\text{trace}(\mathbf{G}))^2, \quad (3.67)$$

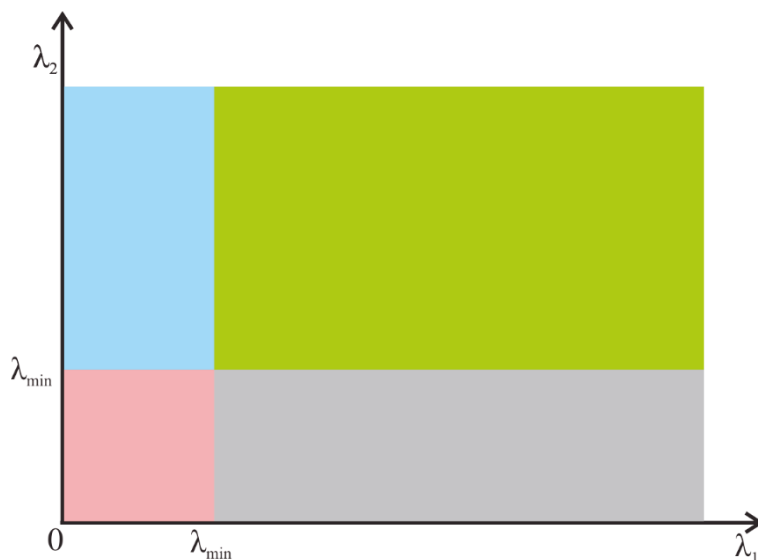
$$\det(\mathbf{G}) = \lambda_1 \lambda_2, \quad (3.68)$$

$$\text{trace}(\mathbf{G}) = \lambda_1 + \lambda_2, \quad (3.69)$$

pri čemer je parameter k poljubna vrednost med 0,04 in 0,15. Ker je strukturni tenzor $\mathbf{G}$ simetrična matrika, jo lahko zapišemo (z uporabo razcepa z lastnimi vrednostmi) tudi kot:

$$\mathbf{G} = \mathbf{R}^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \mathbf{R}. \quad (3.70)$$

Shi-Tomasijev (tudi Kanade-Tomasijev) [147] detektor vogalov neposredno upošteva $R = \min(\lambda_1, \lambda_2)$, saj to izboljša stabilnost detektiranja vogalov pri sledenju. V primeru, da je R večji od predhodno izbrane vrednosti λ_{min} , vemo, da smo detektirali vogal. Slika 3.18 prikazuje štiri področja, pri čemer le zeleno področje označuje, pri katerih lambdah detektiramo vogal.



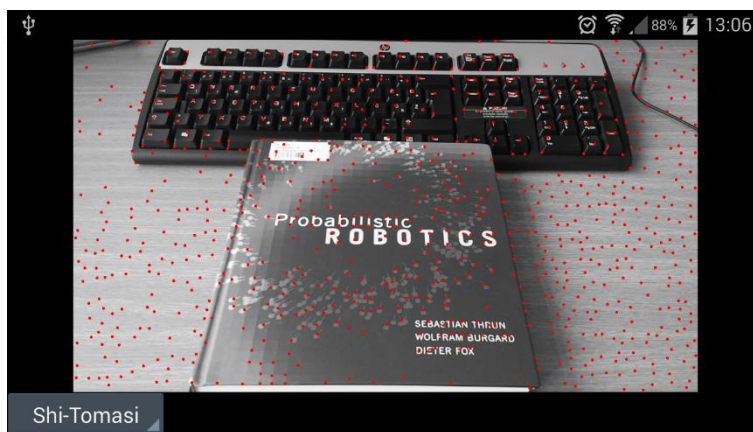
Slika 3.18: Področja, ki povedo, ali smo detektirali vogal (zeleno področje), rob (sivo in modro področje) ali nič (roza področje)

Tako za Harrisov, kot tudi Shi-Tomasijev detektor vogalov velja, da sta invariantna na rotacijo ter da nista invariantna na skaliranje slike in afine transformacije. Detektorja sta invariantna na spremembe osvetljenosti. Invariantnost na afine transformacije lahko dosežemo s tem, da sliko analiziramo pri različnih skalah. To pomeni, da na isti sliki večkrat detektiramo robove, pri čemer vsakič spremenimo velikosti okna. V praksi se ne spreminja velikost okna, vendar se iz ene slike tvori piramida slik, ki imajo različno skalo (kako izbrati pravilno skalo je predlagal Lindeberg [153]).

Iskanje značilnih točk (vogalov) lahko strnemo v pet korakov:

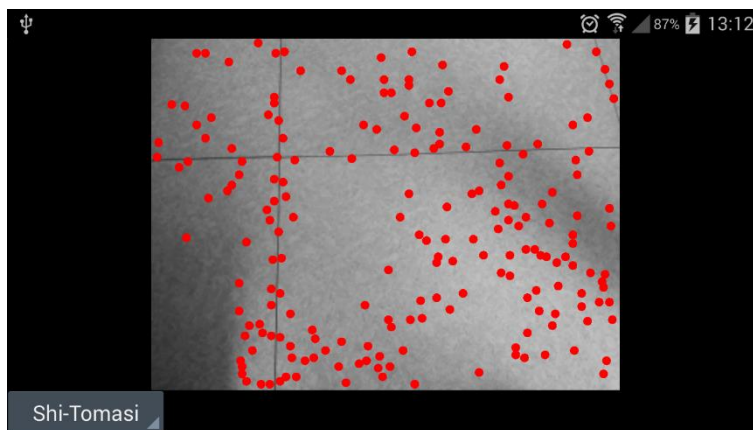
- Izračunamo matriko $\mathbf{G}$ in poiščemo njeno minimalno lastno vrednost λ_{min} za vse slikovne točke slike I .
- Med vsemi lastnimi vrednostmi λ poiščemo največjo lastno vrednost λ_{max} , ki pripada sliki I .
- Obdržimo le tiste slikovne točke, ki imajo λ_{min} večjo od določenega praga (npr. $0,1 \cdot \lambda_{max}$).
- V vseh območjih velikosti 3×3 obdržimo le tisto slikovno točko, katere je λ_{min} največja (obdržimo lokalne maksimume).
- Obdržimo podmnožico tistih slikovnih točk, za katere velja, da je minimalna razdalja med katerimkoli parom točk večja od določenega praga.

Na sliki 3.19 vidimo označene vogale, ki smo jih detektirali z uporabo Shi-Tomasijevega detektorja vogalov. Algoritem smo preizkusili na pametnem telefonu Android.



Slika 3.19: Detektirani vogali z uporabo algoritma Shi-Tomasi pri velikosti slike 1280×720 slikovnih točk

Za primerjavo smo dodali sliko 3.20, ki prikazuje detekcijo vogalov na tleh (homogena tekstura) z uporabo algoritma Shi-Tomasi pri velikosti slike 320×240 slikovnih točk.

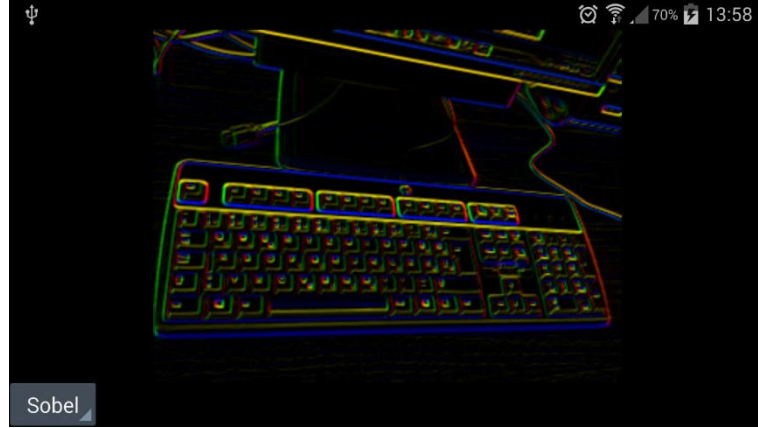


Slika 3.20: Detektirani vogali z uporabo algoritma Shi-Tomasi pri velikosti slike 320×240 slikovnih točk

3.3.5 Sobel operator

Sobel operator (tudi Sobel filter) [154, 155] se uporablja na področjih obdelave slik in strojnega vida predvsem kot del algoritmov za detekcijo robov (ali vogalov), in sicer

za izračun gradienta slike. Matematično je ta operator diskretni diferenčni operator, ki omogoča aproksimativen izračun gradienta funkcije intenzitete slike. Kot rezultat operatorja Sobel, dobimo za vsako točko slike gradientni vektor ali normo tega vektorja. Na sliki 3.21 je za primer prikazana velikost (magnituda) gradienta slike.



Slika 3.21: Velikost gradienta slike, ki ga dobimo z uporabo operatorja Sobel

Sobel operator temelji na konvoluciji slike s preprostim filtrom (masko) v horizontalni in vertikalni smeri in je posledično računsko nezahteven postopek. Po drugi strani pa je gradientna aproksimacija relativno grobo podana, kar pomeni, da lahko predvsem pri visoko frekvenčnih spremembah v sliki dobimo slabši rezultat.

Filter lahko opišemo z dvema matrikama velikosti 3×3 , ki jih uporabimo pri konvoluciji s sliko in tako dobimo približen izračun odvodov za horizontalne in vertikalne spremembe. Če označimo z I izvorno sliko ter z G_x in G_y slike, ki v vsaki točki vsebujeta približek odvoda v horizontalni in vertikalni smeri:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I, \quad (3.71)$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I, \quad (3.72)$$

kjer $*$ označuje operator konvolucije.

Nad filtrom Sobel lahko izvedemo dekompozicijo [156] in tako dobimo produkt dveh delov, ki predstavljata povprečenje in diferenciacijo filtra. To pomeni, da kot rezultat konvolucije dobimo gradient z vključenim glajenjem. Kot primer lahko zapišemo G_x kot:

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}. \quad (3.73)$$

Za vsako točko slike lahko združimo aproksimacijo gradienta v horizontalni in vertikalni smeri in tako dobimo velikost gradienta:

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}. \quad (3.74)$$

Smer gradienta pa izračunamo kot:

$$\theta = \text{atan2}(\mathbf{G}_y, \mathbf{G}_x). \quad (3.75)$$

Kljub temu, da Sobel operator ne omogoča najbolj natančno določanje gradienta slike, je še vedno uporaben pri implementaciji praktičnih aplikacij. Za računanje gradienta slike se uporabljajo tudi drugi operatorji kot npr. Prewitt, ki ravno tako kot operator Sobel deluje na principu konvolucijske maske.

3.4 RANSAC

Metoda RANSAC (RANdom SAmple Consensus) [157, 158, 159] je iterativna metoda, s katero ocenimo parametre matematičnega modela iz množice podatkov, ki vsebuje zelo odstopajoče primerke (ang. outliers). Metoda je nedeterministična v smislu, da proizvede sprejemljivo dobre rezultate le z določeno verjetnostjo, pri čemer ta verjetnost narašča s številom izvedenih iteracij.

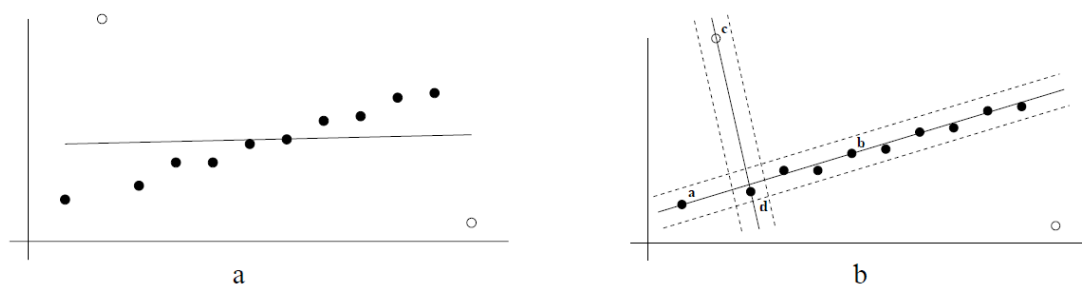
Z algoritmom RANSAC lahko robustno ocenimo parametre modela tudi, ko se med merjenimi vzorci pojavljajo meritve, ki zelo odstopajo od točne vrednosti (napačnih meritev je lahko do 50 %).

Grobo rečeno, algoritem deluje v dveh korakih. V prvem koraku naključno izberemo iz množice vzorcev minimalno število vzorcev za ocenitev parametrov modela. Ocenimo model in nato v drugem koraku preverimo, koliko od preostalih točk se dobro prilega ocenjenemu modelu. Če je število točk, ki se prilega, večje od določenega praga, sprejmemo model in si zapomnimo prilegajoče se točke, sicer ponovimo postopek.

Na sliki 3.22 vidimo primerjavo metode najmanjših kvadratov (levo) in metode RANSAC (desno) na preprostem primeru iskanja premice skozi množico točk v 2D prostoru. V primeru metode najmanjših kvadratov vidimo, da se premica slabo prilega točkam, saj podatki vsebujejo dve zelo odstopajoči točki (prazna krogca), ki močno

vplivata na rezultat. V primeru metode RANSAC pa izbrana premica, ki gre skozi točki a in b , dobro opisuje deset točk, kar pomeni, da sta bili odstopajoči točki ustrezno odstranjeni. Pri postopku iskanja najboljše premice je preizkušenih več različnih premic, ki se jih oceni glede na to, koliko točkam se izbrana premica dobro prilega. Ena izmed možnih premic gre skozi točki d in c , vendar vidimo, da ta premica dobro opisuje le ti dve točki.

Ker pri iskanju optimalnega modela ne vemo v naprej, ali trenutni model dobro opisuje maksimalno število dobrih točk, bi bilo idealno, da bi preizkusili vse modele, ki temeljijo na kombinaciji m -tih točk (m je minimalno potrebno število točk za izgradnjo modela). Vendar ker je teh kombinacij veliko (npr. za premico je $m = 2$, kar pomeni, da je število vseh kombinacij enako $n \cdot (n - 1)/2$, kjer je n število vseh točk), je ta način računsko potraten v primeru, da je n veliko število. V algoritmu RANSAC se preverjanju vseh kombinacij izognemo tako, da upoštevamo grobo oceno o deležu dobrih točk. Tako lahko iskanje modela z maksimalnim številom dobrih točk zaključimo po K iteracijah.



Slika 3.22: Primerjava metode najmanjših kvadratov (a) in metode RANSAC (b)

Naj bo p verjetnost, da smo našli množico točk, ki ne vsebuje zelo odstopajočih točk. Z w označimo verjetnost, da smo izbrali dobro točko iz množice n -tih točk. Zato w predstavlja delež dobrih točk med danimi podatki: $w = \text{št. dobrih točk} / n$. Vrednost w^m predstavlja verjetnost, da smo pri izbiri m točk izbrali samo dobre točke. Vrednost $1 - w^m$ predstavlja verjetnost, da je med m izbranimi točkami vsaj ena slaba. Če s K označimo število iteracij, potem je vrednost $(1 - w^m)^K$ verjetnost, da algoritem RANSAC nikoli ne izbere m točk, pri čemer bi bile vse dobre. Ta verjetnost mora biti enaka vrednosti $1 - p$, kjer je p verjetnost, da je bila najdena množica z m elementi, ki ne vsebuje odstopajočih točk. Torej velja:

$$1 - p = (1 - w^m)^K. \quad (3.76)$$

Iz enačbe (3.76) izrazimo K in dobimo:

$$K = \frac{\log(1-p)}{\log(1-w^m)}. \quad (3.77)$$

Če poznamo delež dobrih točk w , potem lahko glede na enačbo (3.77) napovemo, koliko iteracij K potrebujemo, da bi z verjetnostjo p našli množico točk, ki ne vsebuje slabih točk. Npr. če želimo, da je verjetnost uspeha $p = 99\%$ in vemo, da je delež dobrih točk 50% ($w^m = 0,5^m$, $m = 2$), potem lahko izračunamo, da je $K = 16$ dovolj iteracij.

V praksi ne potrebujemo natančno vedeti deleža dobrih točk v množici podatkov. Bolj napredne implementacije algoritma RANSAC ocenjujejo delež dobrih točk adaptivno korak za korakom [160]. In sicer, na začetku izberemo najslabšo možno situacijo, kar pomeni, da v podatkih ni dobrih točk, torej velja $1 - w = 1$. Iz enačbe (3.77) vidimo, da to pomeni, da je $K = \infty$. Nato ponavljamo postopek iskanja maksimalnega števila dobrih točk ob izbranem modelu in glede na ta rezultat zmanjšujemo potrebno število iteracij K :

```

 $K = \infty$ , št.ponovitev = 0
ponavljamo dokler je  $K > \text{št.ponovitev}$  {
    Izberemo  $m$  točk, izračunamo model in določimo število dobrih točk
    Izračunamo  $w = \text{št.dobrih točk} / n$ 
    Izračunamo  $K$  po enačbi (3.77)
     $\text{št.ponovitev} = \text{št.ponovitev} + 1$ 
}

```

Algoritem RANSAC lahko strnemo v naslednjih petih točkah:

1. Predpostavimo, da imamo n podatkovnih točk $\mathbf{X} = \{x_1, \dots, x_n\}$, za katere želimo najti model, ki dobro opisuje te točke. Model je določen na podlagi m točk ($m \leq n$, za premico je $m = 2$, pri določanju togega premika je $m = 3$, pri določanju homografije je $m = 4$).
2. Postavimo iteracijski števec $k = 1$.
3. Izberemo m naključnih točk iz množice X in izračunamo model.

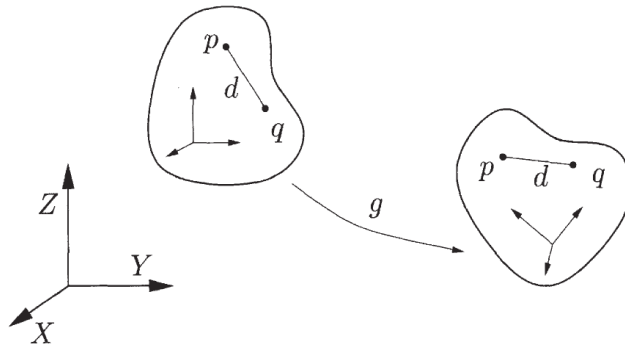
4. Pri upoštevanju tolerančne vrednosti ε določimo, koliko elementov iz X ustreza dobljenemu modelu. Če je teh elementov več kot je nastavljen prag t , potem ponovno izračunamo model, ki temelji na vseh elementih, ki so ustrezali prejšnjemu modelu.
5. Postavimo $k = k + 1$. Če je $k < K$ za nek določen K , se vrnemo na korak 3. V nasprotnem primeru zadnji dobljeni model razglasimo za najboljšega.

Algoritem temelji na izbiri treh parametrov:

- ε določa, koliko največ lahko točke iz množice X odstopajo od točk, ki jih dobimo z uporabo modela, da lahko rečemo, da model dobro opisuje točke iz X .
- t določa, koliko elementov iz X mora model dobro opisati, da lahko rečemo, da je model ustrezen za nadaljnjo uporabo. Za parameter naj bi veljalo: $t - m > 5$.
- K določa, kolikokrat se mora izvesti algoritem, da dobimo ustrezen model. Da bi našli m dobrih točk, na katerih temelji model, mora biti $K > \frac{\log(1-p)}{\log(1-w^m)}$, kjer je w verjetnost, da se naključno izbrana točka iz X razlikuje za manj kot ε od točke dobljene iz modela. V praksi lahko izberemo $K = 2w^{-m}$ ali $K = 3w^{-m}$, kjer w^{-m} predstavlja približno vrednost standardne deviacije.

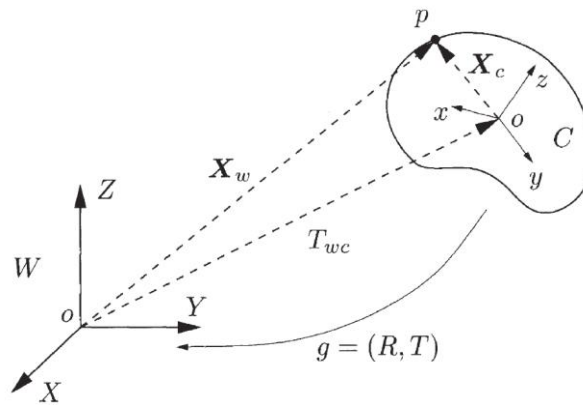
3.5 Togi premik

Pri togem premiku (sestavljenem iz več rotacij in translacij) [161] ne potrebujemo opisovati gibanja posameznih točk, ampak opišemo le gibanje ene točke, ki predstavlja gibanje celotne množice točk. Razlog, da lahko to storimo, je v tem, da se pri togem premiku razdalja med posameznimi točkami ne spreminja (slika 3.23). Poleg tega se pri togem gibanju ohranja tudi orientacija. Torej preslikava $\mathbf{g}: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ predstavlja togo gibanje, če se pri tem poleg norme vektorjev (vektorji so definirani s točkami p in q) ohranja tudi njihov vektorski produkt.



Slika 3.23: Togi premik

Na sliki 3.24 je prikazano gibanje kamere, katere koordinatni sistem je označen kot $C: (x, y, z)$ glede na svetovni koordinatni sistem $W: (X, Y, Z)$. Transformacija $\mathbf{g}_{wc} = (\mathbf{R}_{wc}, \mathbf{T}_{wc})$ določa togo preslikavo točk iz koordinatnega sistema kamere v svetovni koordinatni sistem.



Slika 3.24: Togi premik med svetovnim koordinatnim sistemom in koordinatnim sistemom kamere

Matrika $\mathbf{R}_{wc}$ predstavlja rotacijsko matriko, $\mathbf{T}_{wc}$ pa translacijski vektor. Rotacijska matrika $\mathbf{R}_{wc}$ je ortogonalna, kar pomeni, da velja: $\mathbf{R}_{wc}^T \mathbf{R}_{wc} = \mathbf{R}_{wc} \mathbf{R}_{wc}^T = \mathbf{I}$. Inverz rotacijske matrike je enak transponirani matriki: $\mathbf{R}_{wc}^{-1} = \mathbf{R}_{wc}^T$. Determinanta rotacijske matrike $\mathbf{R}_{wc}$ je enaka +1. Prostor vseh ortogonalnih matrik v $\mathbb{R}^3$ lahko zapišemo kot: $SO(3) = \{\mathbf{R} \in \mathbb{R}^{3 \times 3} \mid \mathbf{R}^T \mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = +1\}$. Opazovano točko p zapišemo v svetovnem koordinatnem sistemu kot: $\mathbf{X}_w = [X_{1w}, X_{2w}, X_{3w}]^T \in \mathbb{R}^3$ in v koordinatnem sistemu kamere kot $\mathbf{X}_c = [X_{1c}, X_{2c}, X_{3c}]^T \in \mathbb{R}^3$. Z upoštevanjem transformacije $\mathbf{g}_{wc} = (\mathbf{R}_{wc}, \mathbf{T}_{wc})$ lahko zapišemo povezavo:

$$\mathbf{X}_w = \mathbf{R}_{wc} \mathbf{X}_c + \mathbf{T}_{wc} \quad (3.78)$$

oz.

$$\mathbf{X}_c = \mathbf{R}_{wc}^T (\mathbf{X}_w - \mathbf{T}_{wc}). \quad (3.79)$$

Množico vseh možnih transformacij opisuje prostor vseh togih premikov $SE(3)$:

$$SE(3) = \{\mathbf{g} = (\mathbf{R}, \mathbf{T}) \mid \mathbf{R} \in SO(3), \mathbf{T} \in \mathbb{R}^3\}.$$

Vsako tego preslikavo $\mathbf{g} = (\mathbf{R}, \mathbf{T})$ pa lahko zapišemo tudi v obliki ene matrike velikosti 4×4 , vendar moramo za ta namen vpeljati t.i. homogene koordinate. Te dobimo tako, da vektorju $\mathbf{X} = [X_1, X_2, X_3]^T \in \mathbb{R}^3$, ki opisuje točko p , dodamo enko in tako dobimo vektor v $\mathbb{R}^4$:

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix} \in \mathbb{R}^4. \quad (3.80)$$

Togo transformacijo lahko sedaj zapišemo v »linearni« obliki kot:

$$\mathbf{X}_w = \begin{bmatrix} \mathbf{R}_{wc} & \mathbf{T}_{wc} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X}_c \\ 1 \end{bmatrix} = \mathbf{g}_{wc} \mathbf{X}_c, \quad (3.81)$$

kjer je $\mathbf{g}_{wc} \in \mathbb{R}^{4 \times 4}$ homogena predstavitev togega premika $\mathbf{g}_{wc} = (\mathbf{R}_{wc}, \mathbf{T}_{wc}) \in SE(3)$. Inverz matrike dobimo kot:

$$\mathbf{g}_{wc}^{-1} = \begin{bmatrix} \mathbf{R}_{wc} & \mathbf{T}_{wc} \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{R}_{wc}^T & -\mathbf{R}_{wc}^T \mathbf{T}_{wc} \\ 0 & 1 \end{bmatrix} \in SE(3). \quad (3.82)$$

3.5.1 Izračun togega premika

Iščemo tak togi premik med dvema množicama točk $\{\mathbf{x}_i\}$ in $\{\mathbf{p}_i\}$; $i = 1:N$ v dvodimenzionalnem prostoru, ki bo minimiziral naslednjo kriterijsko funkcijo:

$$f(\mathbf{p}) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{R}(\vartheta) \cdot \mathbf{p}_i - \mathbf{T}\|^2, \quad (3.83)$$

kjer je $\mathbf{R}_{2 \times 2}$ (v 3D prostoru je velikost matrike 3×3) rotacijska matrika, ϑ kot vrtenja in $\mathbf{T}_{2 \times 1}$ (v 3D prostoru je velikost vektorja 3×1) translacijski vektor.

Če označimo geometrijsko središče vseh točk iz prve množice s $\mathbf{c} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$ in geometrijsko središče vseh točk iz druge množice s $\mathbf{c}' = \frac{1}{N} \sum_{i=1}^N \mathbf{p}_i$, potem lahko translacijo zapišemo kot:

$$\mathbf{T} = \mathbf{c} - \mathbf{R}\mathbf{c}'. \quad (3.84)$$

Da bi lahko določili translacijo, moramo najprej rešiti problem ocene rotacije med dvema množicama točk, in sicer med $\hat{\mathbf{p}}_i = \mathbf{p}_i - \mathbf{c}'$ ter $\hat{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{c}$. Za točke $\hat{\mathbf{p}}_i$ in $\hat{\mathbf{x}}_i$ velja, da imajo geometrijsko središče v koordinatnem izhodišču.

Rotacijsko matriko lahko dobimo z uporabo ortogonalnega Procrustesovega algoritma [162, 163]. Pri Procrustesovem problemu v splošnem iščemo geometrijske transformacije, kot so rotacija, translacija in skaliranje, s katerimi lahko dosežemo optimalno prekrivanje dveh likov oz. točk, ki opisujejo ta dva lika.

Če želimo primerjati dva lika (ki sta predstavljena z dvema množicama točk), moramo najprej doseči, da imata podoben položaj in velikost. Nato izvedemo primerjavo z upoštevanjem Procrustesove razdalje, ki jo izračunamo kot povprečje vsote kvadratov razlik med izbranimi točkami.

Arun in drugi [164] so v svojem delu predstavili način reševanja Procrustesovega problema z uporabo singularnega razcepa - SVD (opis v dodatku A) korelacijske matrike. Korelacijsko matriko $\mathbf{C}$ velikosti 2×2 (oz. 3×3 v 3D prostoru) zapišemo kot:

$$\mathbf{C} = \sum_{i=1}^N \hat{\mathbf{p}}_i \hat{\mathbf{x}}_i^T = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T, \quad (3.85)$$

kjer $\mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$ predstavlja singularni razcep. Matriki $\mathbf{U}$ in $\mathbf{V}$ sta ortonormalni matriki, $\mathbf{\Sigma}$ pa je diagonalna matrika z nenegativnimi elementi. Rotacijsko matriko izračunamo iz dobljenih matrik $\mathbf{U}$ in $\mathbf{V}$ singularnega razcepa korelacijske matrike kot [164]:

$$\mathbf{R} = \mathbf{V} \mathbf{U}^T. \quad (3.86)$$

Za matriko $\mathbf{R}$ moramo vedno preveriti velikost njene determinante, saj mora biti ta enaka +1, da je rotacijska matrika pravilno določena. V nekaterih primerih v 3D prostoru, ko so točke komplanarne, pa se lahko zgodi, da je determinanta enaka -1 in v tem primeru moramo matriko $\mathbf{R}$ izračunati kot:

$$\mathbf{R} = \mathbf{V}' \mathbf{U}^T, \quad (3.87)$$

kjer je $\mathbf{V}' = [\mathbf{v}_1, -\mathbf{v}_2]$. $\mathbf{v}_2$ je stolpični vektor, ki pripada singularni vrednosti (ki je enaka nič) matrike $\mathbf{C}$.

Ob znani rotacijski matriki lahko zapišemo:

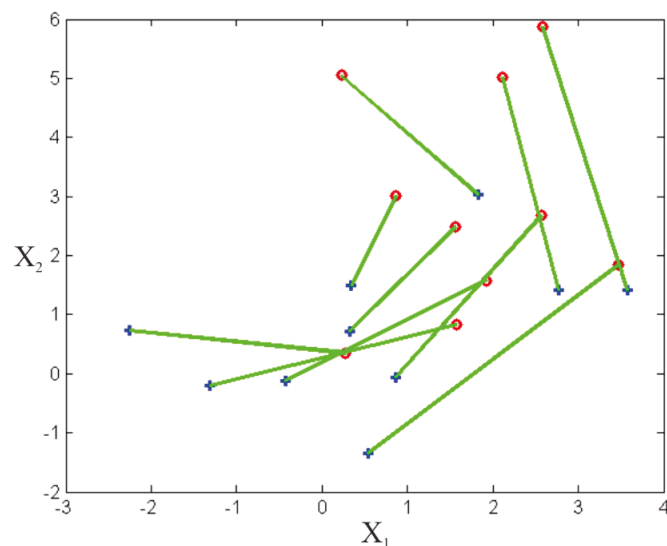
$$\hat{\mathbf{x}}_i = \mathbf{R} \hat{\mathbf{p}}_i. \quad (3.88)$$

Kot vrtenja lahko izračunamo iz rotacijske matrike $\mathbf{R} = \begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) \\ \sin(\vartheta) & \cos(\vartheta) \end{bmatrix}$ kot:

$$\vartheta = \text{atan2} \frac{R_{21}}{R_{11}}.$$

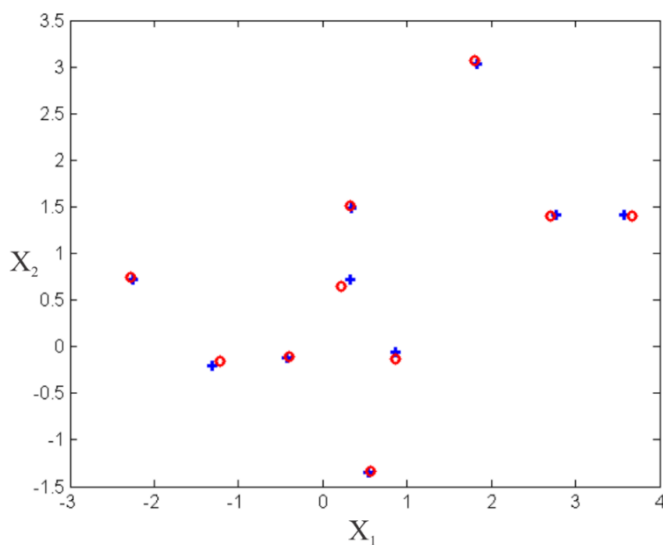
Ko imamo rotacijsko matriko znano, lahko izračunamo še translacijsko matriko z upoštevanjem enačbe (3.84). Za oceno togega premika v 3D prostoru potrebujemo minimalno tri točke v vsaki množici oz. dve, če iščemo premik v 2D prostoru.

Za namen prikaza delovanja algoritma za določanje togega premika smo generirali deset naključnih točk, ki imajo normalno porazdelitev s srednjo vrednostjo 0 in standardnim odklonom 1 (modri križci na sliki 3.25). Nato smo te točke zarotirali za rotacijsko matriko $\mathbf{R} = \begin{bmatrix} 0,5 & -\sqrt{3}/2 \\ \sqrt{3}/2 & 0,5 \end{bmatrix}$ in premaknili za translacijski vektor $[2 \ 2]^T$ (rdeči krogci na sliki 3.25) ter jim dodali šum. Po opisanem postopku smo izračunali korelacijsko matriko in na njej izvedli singularni razcep (SVD). Nato smo izračunali rotacijsko in translacijsko matriko po enačbah (3.84) in (3.87). Da bi lahko preverili, ali so dobljeni rezultati pravilni, smo točke iz druge množice zarotirali za dobljeno rotacijsko matriko in nato še translirali za translacijski vektor. Na sliki 3.26, kjer so prikazane obe množici točk, lahko vidimo, da smo z opisanim algoritmom pravilno določili togi premik.



Slika 3.25: Togi premik prve množice točk (modri križci) za kot $\vartheta = 60^\circ$ in translacijo $[2 \ 2]^T$ v drugo množico točk (rdeči krogci)

Na sliki 3.26 je mogoče opaziti, da se točke iz obeh množic ne ujemajo povsem, kar je posledica dodanega šuma.



Slika 3.26: Prekrivanje točk iz prve množice in točk iz druge množice, ki so premaknjene glede na izračunani rotacijski in translacijski matriki

3.6 Opis monokularne vizualne odometrije

Vizualna odometrija (VO) je proces zaporednega ocenjevanja gibanja kamere glede na zaznane premike slikovnih točk iz sekvence slik [165]. Izraz »vizualna odometrija« je v literaturi pogosto zamenjan z izrazom »struktura iz gibanja« (ang. Structure From Motion), vendar se moramo zavedati, da izraza ne pomenita enako, saj pri vizualni odometriji ocenjujemo le relativni položaj kamere, ne gradimo pa strukture okolja.

Pri implementaciji monokularne vizualne odometrije so bile upoštevane naslednje predpostavke:

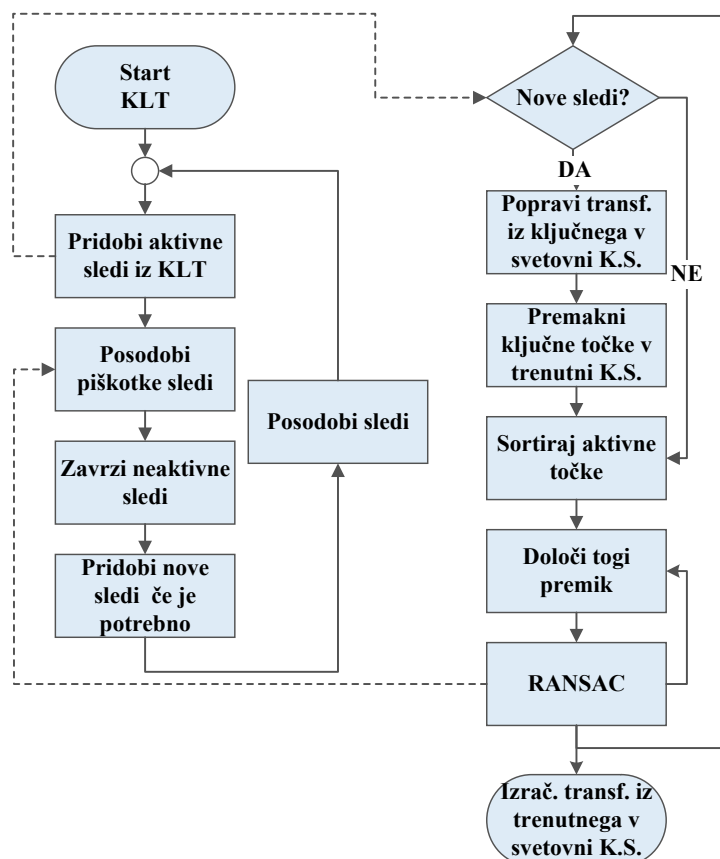
- kamera vidi ravno ploskev (tla),
- točke, ki se ne nahajajo na tleh, so v neskončnosti,
- togi premik med koordinatnima sistemoma kamere in ploskve je ves čas enak.

Zadnja predpostavka pogosto drži za avtonomna vozila in robote opremljene s kamero ter do neke mere tudi za človeka (na katerega je pritrjena kamera oz. pametni telefon), ki se pri normalni hoji giba po ravni podlagi (če zanemarimo dviganje in spuščanje telesa med hojo).

Kot je že bilo omenjeno, algoritem vizualne odometrije sestoji iz štirih osnovnih komponent, ki so bile opisane v prejšnjih podpoglavjih. Prva komponenta je kalibracijski algoritem, ki pa se ne izvaja kot del vizualne odometrije, ampak se izvede le enkrat kot inicializacijski postopek. S tem se določi transformacija med

koordinatnim sistemom kamere in podlage. Algoritem KLT, algoritem za določanje togega premika in algoritem RANSAC pa so tri osnovne komponente vizualne odometrije, ki se morajo izvajati v realnem času na izbranem hardveru – pametnem telefonu. Ker imajo sodobni pametni telefoni vgrajene večjedrne procesorje, smo algoritem vizualne odometrije zgradili tako, da računsko bolj potratni deli algoritma vizualne odometrije tečejo na ločenih nitih procesorja. To pomeni, da se posamezni sestavni deli vizualne odometrije lahko izvajajo simultano.

Na sliki 3.27 je prikazan diagram poteka vizualne odometrije, ki je razdeljen na dva dela, kar je skladno s tem, da je bil algoritem pri implementaciji razdeljen na dve procesorski niti. Da bi bila oba dela algoritma glede računske potratnosti približno enaka, smo algoritem KLT oz. določanje sledi značilnih točk dodelili eni niti (leva polovica diagrama), algoritem določanja togega premika in algoritem RANSAC pa drugi niti (desna polovica diagrama). Ker algoritem KLT deluje bolje pri manjših premikih značilnih točk, je pomembno, da se algoritem izvaja hitro in tako obdela večje število sličic na sekundo (tako so premiki značilnih točk manjši).



Slika 3.27: Diagram poteka izračuna togega premika v 2D

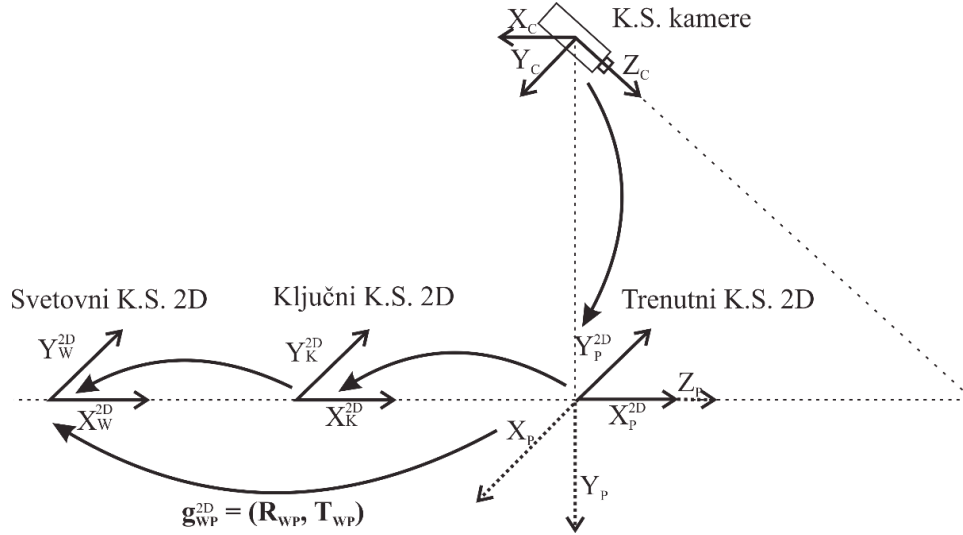
Ker morajo vsi sestavni deli algoritma vizualne odometrije, ki tečejo na ločenih nitih delovati kot celota, morata niti med sabo tudi ustrezno komunicirati. Tako si niti med sabo izmenjujeta podatke, kot je prikazano na sliki 3.27 s črtkanimi povezavami. Nit, ki skrbi za določanje sledi značilnih točk, pošilja informacije o trenutno aktivnih sledeh točk drugi niti, druga nit pa pošilja informacije prvi niti o tem, katere sledi so bile uporabljene pri določanju togega premika in katere zelo odstopajo glede na izbrani model togega premika. Te informacije se shranijo v t.i. piškotek posamezne sledi. V KLT niti se ti piškotki ob vsaki novi sličici osvežijo in glede na dane informacije se po potrebi izvede brisanje in dodajanje novih sledi (desna stran diagrama na sliki 3.27).

V nadaljevanju sledi opis dela algoritma, ki je na sliki 3.27 predstavljen z desno polovico sheme. Ta del algoritma v osnovi sestoji iz algoritma za določanje togega premika in algoritma RANSAC, z uporabo katerih dobimo željeno togo preslikavo iz K.S. kamere v svetovni K.S., ki predstavlja izhod vizualne odometrije.

Pri vizualni odometriji zaznane premike izračunavamo v 2D prostoru ter nato končen rezultat pretvorimo v 3D prostor. Pri izračunu togega premika v 2D so pomembni trije koordinatni sistemi, in sicer svetovni (W), ključni (K) in trenutni (P) K.S., ki so ob zagonu vizualne odometrije poravnani (slika 3.28). Z vsako novo sličico dobimo nov položaj točk aktivnih sledi v trenutnem K.S. Svetovni K.S., ki je določen s prvo sličico, se ne spreminja in predstavlja referenčni K.S., v katerem dobimo končen rezultat vizualne odometrije, tj. relativni položaj kamere glede na začetni položaj. Ključni K.S. se premakne le v primeru, ko dodajamo nove sledi, in sicer v trenutni K.S. V tem primeru se osveži tudi toga preslikava iz ključnega v svetovni K.S., ki postane enaka preslikavi g_{WP}^{2D} .

Iz slikovnih točk, ki jih določajo aktivne sledi, lahko dobimo točke na normalizirani slikovni ravnini, tako da upoštevamo intrinzične parametre in radialno distorzijo. Te točke lahko nato preslikamo v 2D prostor ravnine (na tla) tako, da so koordinate izražene v trenutnem K.S. ravnine. Kot vidimo na sliki 3.28, ima trenutni K.S. (2D) osi obrnjene tako, da je os Y_P^{2D} enaka osi $-X_P$ K.S. ravnine, X_P^{2D} os pa je enaka osi Z_P . Predno točke uporabimo za izračun togega premika iz trenutnega v ključni K.S., pa je potrebno preveriti, katere zaznane značilne točke ležijo na podlagi (na sliki 3.27 je ta del algoritma označen kot »sortiraj aktivne točke«). To storimo tako, da slikovno točko

opišemo v K.S. ravnine in nato pogledamo, ali je njena Y komponenta pozitivna (to pomeni, da žarek, ki gre skozi to točko na slikovni ravnini, prebada tla).



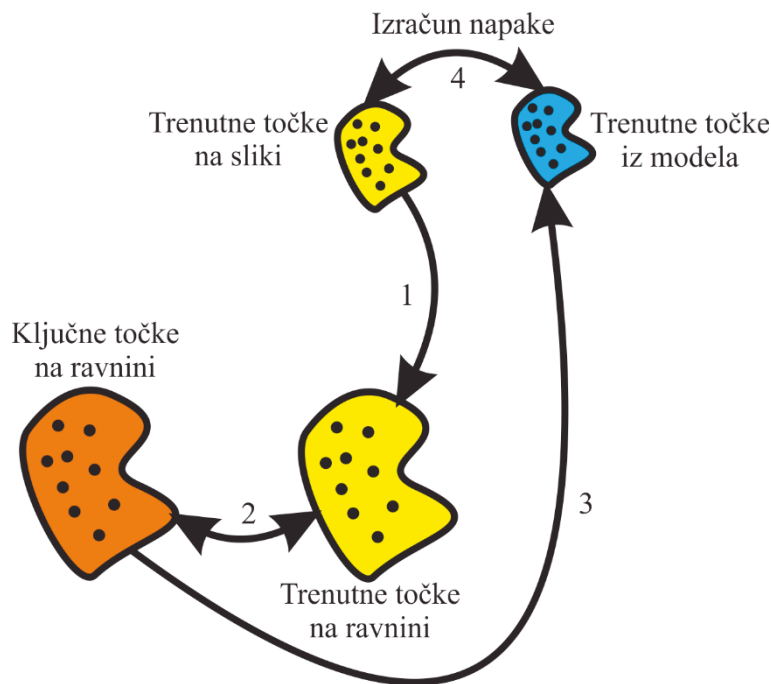
Slika 3.28: Togi premik v 2D iz trenutnega K.S. v svetovni K.S. dobimo tako, da upoštevamo sprotne izračune za togi premik iz ključnega K.S. v trenutni K.S.

Točko (x, y) , ki jo na normalizirani slikovni ravnini določa žarek, ki gre iz gorišča proti tlom, lahko zapišemo v homogenih koordinatah kot $(x, y, 1)$. Da bi lahko v K.S. kamere določili točko na tleh $(\lambda x, \lambda y, \lambda)$, ki jo določa žarek skozi $(x, y, 1)$, potrebujemo še skalirni faktor λ . Ta faktor določa oddaljenost točke v smeri Z koordinatnega sistema kamere. Ta problem rešimo tako, da najprej z upoštevanjem rotacijske matrike R_{PC} zavrtimo točko na normalizirani slikovni ravnini $(x, y, 1)$ ter tako dobimo točko (X, Y, Z) v K.S., ki ima osi enako poravnane kot K.S. podlage (v 3D). Če želimo, da bi ta točka ležala na podlagi, mora biti njena Y koordinata enaka višini kamere od tal. Tako določimo, da je točka na ravnini enaka $(Z \cdot t, -X \cdot t)$, kjer je $t = h/Y$ (to razmerje določa, da točka leži na tleh) in h višina kamere. To dobimo iz poznane translacijskega vektorja T_{PC} , ki ga izračunamo v fazi kalibracije.

Ko so ustrezne točke preslikane iz slikovne ravnine na podlago v trenutni K.S. (2D prostor), lahko izračunamo model togega premika točk g_{KP}^{2D} iz trenutnega v ključni K.S. Ta togi premik predstavlja jedro celotne vizualne odometrije. Točke ključne sličice so predstavljene s ključnim K.S., točke trenutne sličice pa s trenutnim K.S. Togi premik g_{PK}^{2D} med ključnimi točkami in trenutnimi točkami na ravnini se izračunava sproti z uporabo Procrustesove analize. Ključni K.S. se premakne (točke se preslikajo

glede na transformacijo $\mathbf{g}_{PK}^{2D}$ v trenutni K.S. vsakič, ko se v algoritmu določijo nove sledi (ko število vseh sledi pade pod določen prag). V tem primeru se spremeni tudi preslikava $\mathbf{g}_{WK}^{2D}$ iz ključnega v svetovni K.S. glede na transformacijo $\mathbf{g}_{KP}^{2D}$. Algoritem se ponavlja, tako da se za vsako novo sličico (ko se osvežijo aktivne sledi) izračuna preslikava iz trenutnega K.S. v svetovni K.S. z uporabo transformacije $\mathbf{g}_{KP}^{2D}$. Vse 2D toge premike opišemo s translacijo $\mathbf{T} = [X, Y]^T$ in kotom zasuka φ (yaw), ki določa rotacijsko matriko $\mathbf{R} = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix}$.

Pri določanju togih premikov $\mathbf{g}_{PK}^{2D}$ med ključnimi in trenutnimi točkami na ravnini z uporabo Procrustesove analize, se primernost posameznega modela preverja z algoritmom RANSAC (slika 3.29). Ta algoritem preizkuša, kateri model, ki je bil izračunan na podlagi treh naključno izbranih parov točk, ustreza največjemu deležu ostalih parov točk.



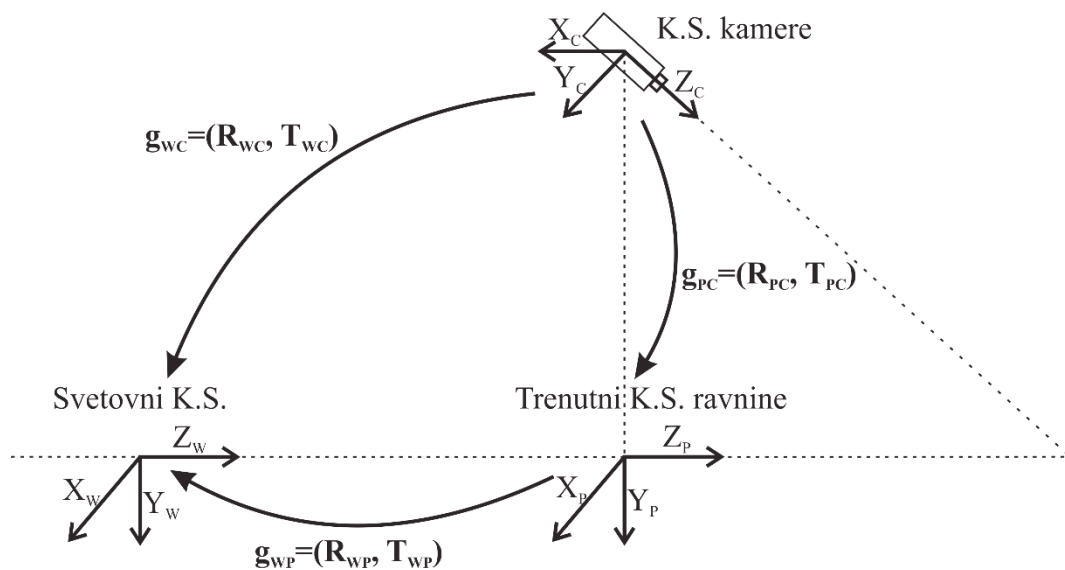
Slika 3.29: Preverjanje ustreznosti togega premika z algoritmom Ransac

Kot lahko vidimo na sliki 3.29, se pri izračunu modela togega premika točke iz normalizirane slikovne ravnine najprej preslikajo na tla v trenutni K.S. (korak 1 na sliki 3.29). Nato se na podlagi trenutnih in ključnih točk izračuna model, ki določa togi premik (korak 2). Zatem dobljeni model uporabimo skupaj s togim premikom med

K.S. podlage in kamere ter na ta način dobimo trenutne točke iz modela v normaliziranih slikovnih koordinatah (korak 3). Te točke primerjamo s pravimi trenutnimi točkami tako, da izračunamo evklidsko razdaljo (korak 4). Ob znani razdalji med točkami vemo, katere točke model opisuje dobro in katere ne. Razdalja med točkami je izražena v slikovnih točkah. Ko z algoritmom RANSAC ugotovimo, katere točke ne ustrezajo dobljenemu modelu, dodamo pripadajoče sledi na seznam za izbris. Posamezna sled je izločena (v niti KLT, tj. leva stran diagrama na sliki 3.27), če ni bila uporabljena v zadnjih nekaj (parameter, ki ga nastavimo) slikah. Ko število aktivnih sledi upade pod določen prag, se z uporabo algoritma KLT generirajo nove sledi.

Da bi lahko izračunali preslikavo iz svetovnega K.S. v K.S. kamere, potrebujemo najprej izračunati togi premik v 3D iz trenutnega K.S. v svetovni K.S. (orientacija K.S. v 3D je enaka, kot jo ima trenutni K.S., ki je na sliki 3.28 označen s pikčastimi osmi): $\mathbf{T}_{WP}^{3D} = [-T_Y, 0, T_x]$, $\mathbf{R}_{WP}^{3D} = \text{roty}(-\varphi)$ (predznak minus dobimo, ker sta osi vrtenja v 2D in 3D v nasprotni smeri).

Po izračunu preslikav $\mathbf{g}_{PW}$ in $\mathbf{g}_{CP}$ v 3D (slika 3.30), lahko izračunamo togi premik iz svetovnega K.S. v K.S. kamere: $\mathbf{g}_{CW} = (\mathbf{R}_{CW}, \mathbf{T}_{CW}) = \mathbf{g}_{CP} \mathbf{g}_{PW}$ ($\mathbf{g}_{CP}$ izračunamo v fazi kalibracije oz. izračuna ekstrinzičnih parametrov). Če izračunamo inverzno preslikavo $\mathbf{g}_{CW}^{-1} = \mathbf{g}_{WC} = (\mathbf{R}_{WC}, \mathbf{T}_{WC})$, potem nam translacijski vektor $\mathbf{T}_{WC}$ pove, kje se glede na svetovni K.S. v danem trenutku nahaja kamera. Ta vektor nam torej predstavlja izhod algoritma vizualne monokularne odometrije.

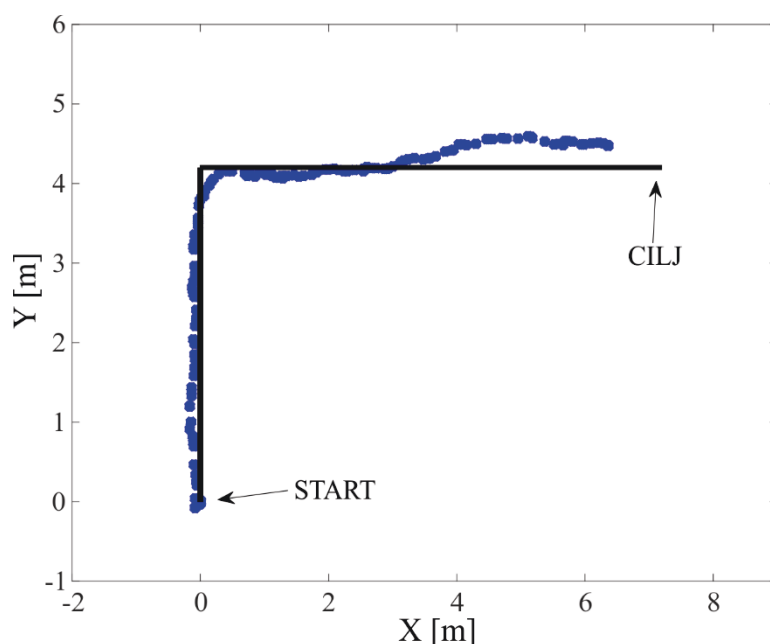


Slika 3.30: Togi premik $\mathbf{g}_{WC}$ iz svetovnega K.S. v K.S. kamere

3.7 Rezultati preizkusa vizualne odometrije

Algoritem vizualne odometrije smo implementirali na pametnem telefonu Android, kjer se lahko izvaja v realnem času, in sicer lahko v eni sekundi obdela od 15 (Samsung Galaxy S4 s procesorjem Qualcomm Snapdragon 600) do 30 (Samsung Galaxy S6 s procesorjem Exynos 7420 Octa) sličic, kar je povsem zadovoljivo pri normalni hoji (2-3 korake na sekundo). Znatna pohiترitev delovanja je bila dosežena predvsem s razdelitvijo algoritma vizualne odometrije na dva dela, ki se lahko izvajata sočasno na ločenih procesorskih jedrih.

Za namen preizkusa vizualne odometrije smo pametni telefon fiksirali na pešca, ki je hodil po referenčni črti dolžine 12 m, ki je bila narisana na podlagi (črna črta na sliki 3.31). Pametni telefon je bil s hrbtno stranjo obrnjen proti podlagi tako, da je optična os kamere imela približen naklon 45° glede na podlago.

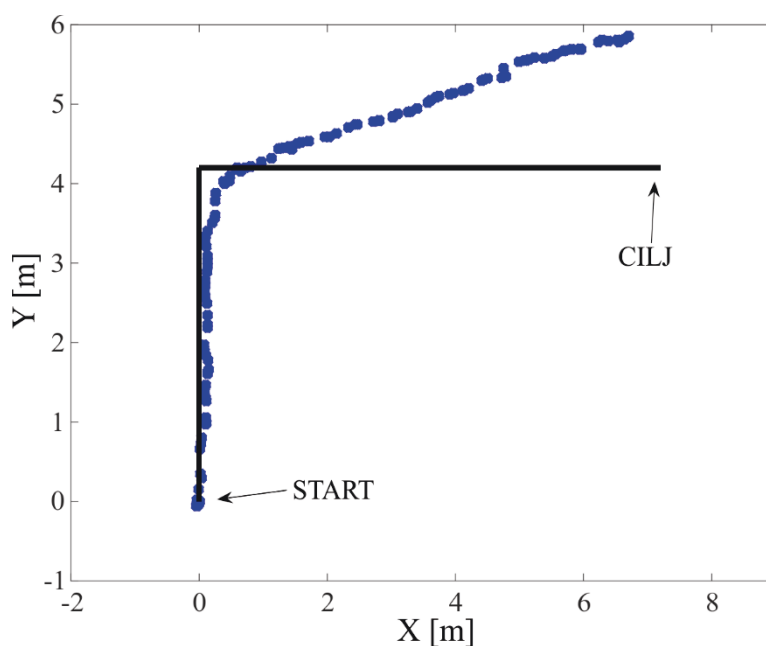


Slika 3.31: Določanje relativnega položaja pešca z vizualno odometrijo

Na sliki 3.31 vidimo, da prvi del poti vizualna odometrija zelo natančno določa relativne premike pešca med hojo, saj so položaji pešca določeni na nekaj centimetrov natančno. Vertikalno premikanje telesa (in s tem kamere) med hojo zanemarljivo vpliva na natančnost vizualne odometrije. V drugem delu poti se pokažejo slabosti navigacije na slepo, saj trajektorija dobljena z vizualno odometrijo začne odstopati od

referenčne poti. Napake se lahko pričnejo pojavljati pri slabi osvetlitvi, monotoni teksturi podlage, zameglitvi slike itd. Že v primeru majhnih napak pri vsakem inkrementalnem premiku imamo po določenem času znatno (akumulirano) napako pri določanju položaja. Na sliki 3.31 vidimo, da končen položaj pešca (določen z VO) odstopa od dejanskega položaja za 0,9 m, kar predstavlja 7,5 % celotne poti.

Pri preizkusu delovanja algoritma vizualne odometrije smo pri enem izmed eksperimentov dobili rezultat, ki je prikazan na sliki 3.32. Ta rezultat je eden izmed slabših, ki smo jih dobili samo z uporabo vizualne odometrije in je zato zelo primeren za razlago možnih napak pri vizualni odometriji. Kot lahko opazimo na sliki 3.32, trajektorija določena z vizualno odometrijo na začetku zelo natančno sledi referenčni črti. Vendar pa v nekem trenutku, zaradi napačno določenega togega premika iz trenutnega v ključni koordinatni sistem, trajektorija prične odstopati od referenčne poti.

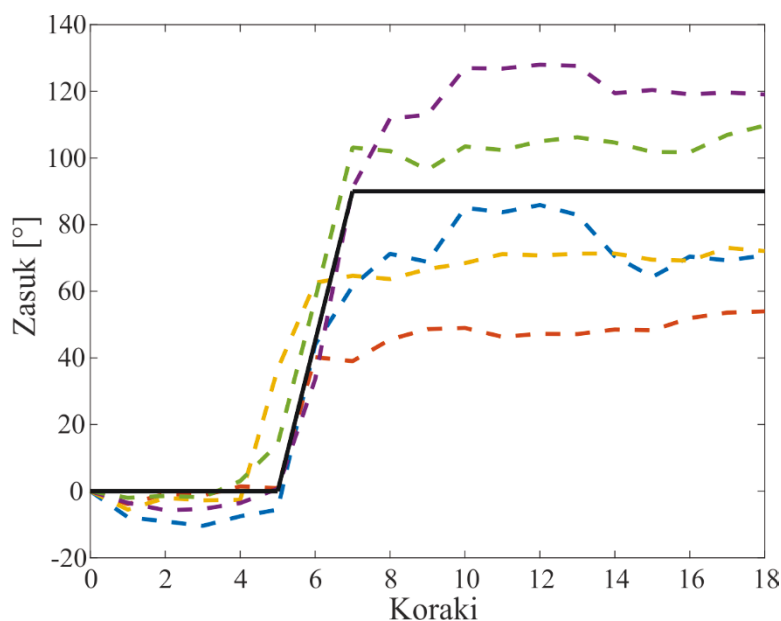


Slika 3.32: Težave vizualne odometrije pri določanju zasuka

Problem nastopi predvsem zaradi nenatančno določenega zasuka (pri zavoju pešca na desno), kar nadalje povzroči, da imamo s časom vedno večjo napako pri določanju položaja. Do te situacije pride zaradi monotone teksture podlage, na kateri smo delali eksperiment. Tako je algoritem KLT pri zasuku pešca določil veliko napačnih sledi in posledično algoritem RANSAC ni uspel določiti pravega modela togega premika.

Ker je bil ta napačno določeni togi premik hkrati upoštevan tudi pri posodobitvi transformacije iz ključnega v svetovni K.S., so bili vsi nadaljnji položaji napačno določeni. Vendar kot lahko vidimo na sliki 3.32, vizualna odometrija nadalje spet pravilno deluje, saj je trajektorija od mesta napake do cilja skoraj povsem ravna tako kot referenčna pot. Torej, če bi v trenutku, ko je prišlo do napačne transformacije ustrezno popravili zasuk, bi dobljena trajektorija zelo lepo sledila referenčni črti. Rešitev za opisani problem se skriva v fuziji vizualne odometrije z inercialnim navigacijskim sistemom, kot je opisano v nadaljevanju.

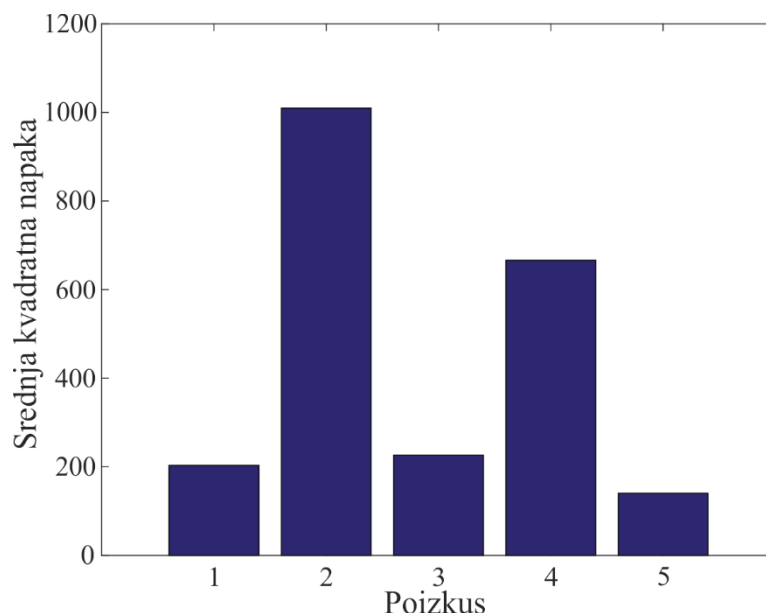
Na sliki 3.33 lahko vidimo spreminjanje zasuka pešca (določenega z VO glede na začetno orientacijo) med hojo po referenčni poti prikazani na sliki 3.31. Eksperiment smo petkrat ponovili. Če opazujemo le nekaj začetnih in nekaj končnih korakov (kjer dejanski zasuk 0° oz. 90° označuje črna črta), vidimo, da se večino (večjih) napak zasuka pojavi zatem, ko pešec naredi zavoj v desno. Po zavoju, ko pešec hodi naravnost, pa so napake manj pogoste (le modra in vijolična krivulja kažeta, da je prišlo do napake tudi pri hoji naravnost).



Slika 3.33: Spreminjanje zasuka (določenega z VO) pri hoji po poti, ki je prikazana na sliki 3.32 (eksperiment je bil petkrat ponovljen)

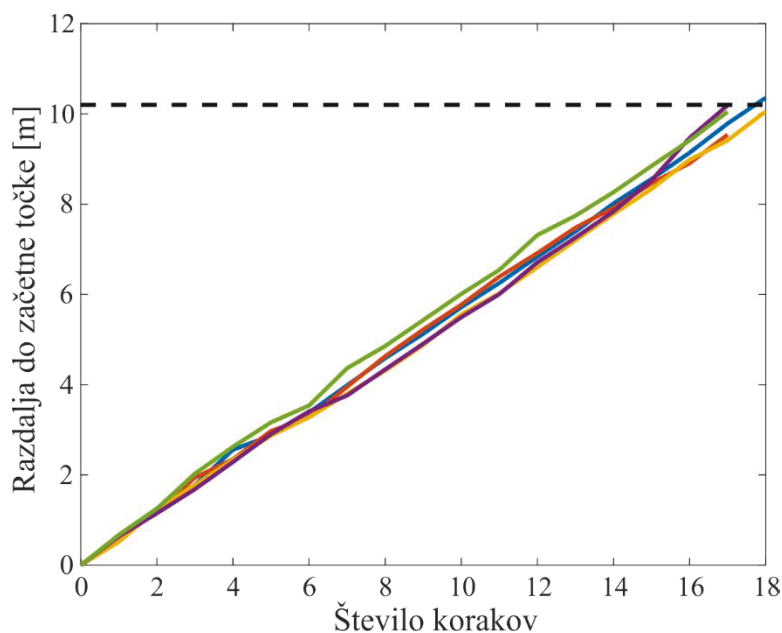
Pri upoštevanju prvih pet in zadnjih osem korakov (na sliki 3.33) lahko izračunamo srednjo kvadratno napako zasuka za vseh pet testov (slika 3.34). Na sliki 3.34 vidimo, da dobimo največjo srednjo kvadratno napako pri drugem (rdeča krivulja na sliki 3.33)

in četrtem poizkusu (vijolična krivulja na sliki 3.33). Namreč v teh dveh primerih vizualna odometrija zelo slabo določi zasuk pešca med zavojem, kar močno vpliva na vse nadaljnje meritve.

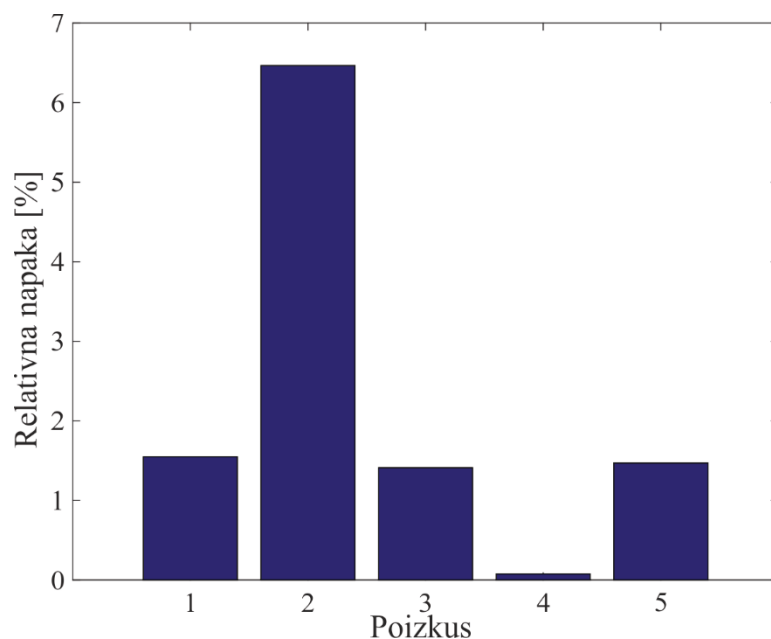


Slika 3.34: Srednja kvadratna napaka pri določanju zasuka z vizualno odometrijo

Da bi lahko nekoliko bolj natančno določili napako pri izračunu dolžine poti z uporabo vizualne odometrije, smo izvedli test, pri katerem je pešec petkrat prehodil ravno pot dolžine 10,2 m. Na sliki 3.35 lahko vidimo (skoraj linearno) naraščanje razdalje do začetne točke v odvisnosti od števila korakov (za vseh pet ponovitev testa). Opazimo lahko, da pri vseh poizkusih razdalja skoraj povsem enako (linearno) narašča za $\sim 0,6$ m na korak. Pri predpostavki, da je pešec celotno pot delal enako dolge korake, bi lahko v primeru nelinearnega naraščanja razdalje ugotovili, ali prihaja do napak znotraj vizualne odometrije. Natančnih napak pri izračunu dolžine inkrementalnih premikov ne moremo podati, saj nimamo zelo natančnega referenčnega sistema, lahko pa ocenimo skupno napako glede na končno razdaljo, ki jo vrne vizualna odometrija. Na sliki 3.36, kjer je prikazana relativna napaka pri izračunu prehojene razdalje z vizualno odometrijo, lahko vidimo, da je ta nekoliko večja le pri drugem poizkusu, in sicer 6,5 %. V ostalih štirih primerih pa je skupna napaka zelo majhna, in sicer manj kot 2 % dolžine poti.



Slika 3.35: Razdalja do začetne točke pri hoji po ravni črti dolžine 10,2 m (eksperiment je bil petkrat ponovljen)



Slika 3.36: Relativna napaka pri izračunu prehojene razdalje z vizualno odometrijo

Zaključimo lahko, da vizualna odometrija predstavlja učinkovit način ocene gibanja robota, opremljenega s cenovno ugodnim senzorjem kot je kamera, ali celo slepega človeka, ki ima na sebi pritrjeno napravo, kot je pametni telefon. Kljub temu, da algoritmi, ki vključujejo obdelavo slik, sodijo med računsko bolj zahtevne, lahko sodobni pametni telefoni poganjajo že tudi te, nekoliko bolj kompleksne algoritme,

kot je vizualna odometrija. Pokazali smo, da vizualna odometrija lahko teče na pametnem telefonu v realnem času in hkrati natančno določa inkrementalne premike.

4 Inercialni navigacijski sistem

PDR (Pedestrian Dead Reckoning) je tehnika, ki se uporablja v inercialnih navigacijskih sistemih (INS) za namen določanja premikov pešca glede na začetno točko. Pri tem je posamezen premik podan s številom napravljenih korakov in smermi, v katerih so bili ti koraki narejeni. Navigacija »na slepo« (ang. dead reckoning), ki temelji na inercialnih senzorjih, je vse bolj prisotna kot dopolnilni sistem tudi pri različnih lokalizacijskih sistemih vozil in robotov. Prednosti tega pristopa so nizka cena in majhne mere senzorjev ter znatni prispevek k izboljšanju natančnosti lokalizacije. Inercialni navigacijski sistem temelji na treh senzorjih, in sicer pospeškometru, žiroskopu in magnetometru. Z obdelavo signalov iz teh senzorjev je mogoče implementirati digitalni kompas za namen določanja orientacije in pedometer za namen izračuna prehojene poti. Ti dve komponenti pa predstavljata osnovo inercialnega navigacijskega sistema. Pedometer je sestavljen iz števca korakov in algoritma za oceno dolžine korakov. Z uporabo pedometra in digitalnega kompasa lahko ob vsakem novem koraku izračunamo trenutni položaj (x_k, y_k) in zasuk θ_k kot:

$$\begin{bmatrix} \theta_k \\ x_k \\ y_k \end{bmatrix} = \begin{bmatrix} \theta_{k-1} + \Delta\theta_k \\ x_{k-1} + l_k \cos\theta_k \\ y_{k-1} + l_k \sin\theta_k \end{bmatrix}, \quad (4.1)$$

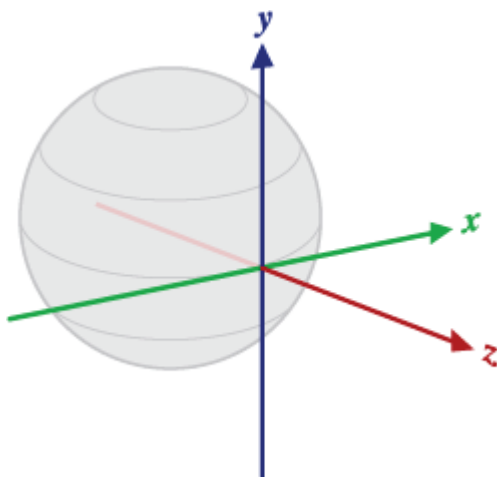
kjer je $\Delta\theta_k$ sprememba kota med korakom v trenutku $k - 1$ in k ter l_k je trenutna dolžina koraka.

Pri robotskih vozilih, kjer uporaba pedometra ni možna, se za namen izračuna dolžine prepotovane poti uporablja dvojna integracija pospeška. Vendar v tem primeru zaradi odstopanja meritev (ang. bias), ki povzročijo napako nagiba (»tilt error«) in šuma, ki ga vsebuje signal pospeškometra, napaka zelo hitro narašča in jo moramo kompenzirati tako, da upoštevamo še neko drugo lokalizacijsko tehniko, kot je npr. vizualna odometrija. V primeru, da je na voljo globalna lokalizacijska tehnika, kot je

npr. lokalizacija z uporabo WiFi ali Bluetooth oddajnikov pa je najbolj primerno uporabiti to za namen omejevanja naraščanja skupne napaka, ki je posledica navigacije »na slepo« [168].

4.1 Digitalni kompas

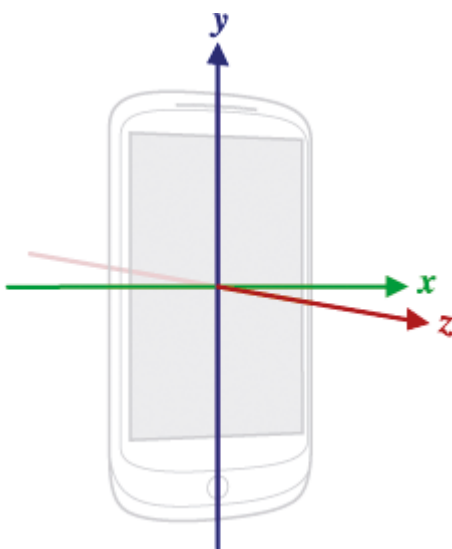
Ena izmed ključnih komponent inercialnega navigacijskega sistema je digitalni kompas, katerega naloga je čim bolj natančno določiti absolutni zasuk naprave (in s tem uporabnika) glede na svetovni koordinatni sistem (slika 4.1). Za delovanje digitalnega kompasa sta v osnovi potrebna dva senzorja, in sicer pospeškometer in magnetometer. Naloga pospeškometra je, da določi smer gravitacijskega pospeška (ta je vzporedna z osjo Z svetovnega K.S.), naloga magnetometra pa, da določi smer magnetnega severa. Glede na ti dve smeri je z vektorskim produktom mogoče izračunati še smer osi X (slika 4.1).



Slika 4.1: Svetovni koordinatni sistem, ki ga uporablja rotacijski vektor [169]

Za bolj natančno in stabilno delovanje digitalnega kompasa pa je potreben še žiroskop, ki natančno meri kotno hitrost (relativni zasuk) in je hkrati tudi veliko bolj odziven kot magnetometer (ima višjo frekvenco vzorčenja). Naloga žiroskopa je eliminirati napačne zasuke, ki jih vrne magnetometer ob prisotnosti magnetnih motenj. Vsi trije senzorji merijo svoje vrednosti glede na koordinatni sistem naprave (slika 4.2).

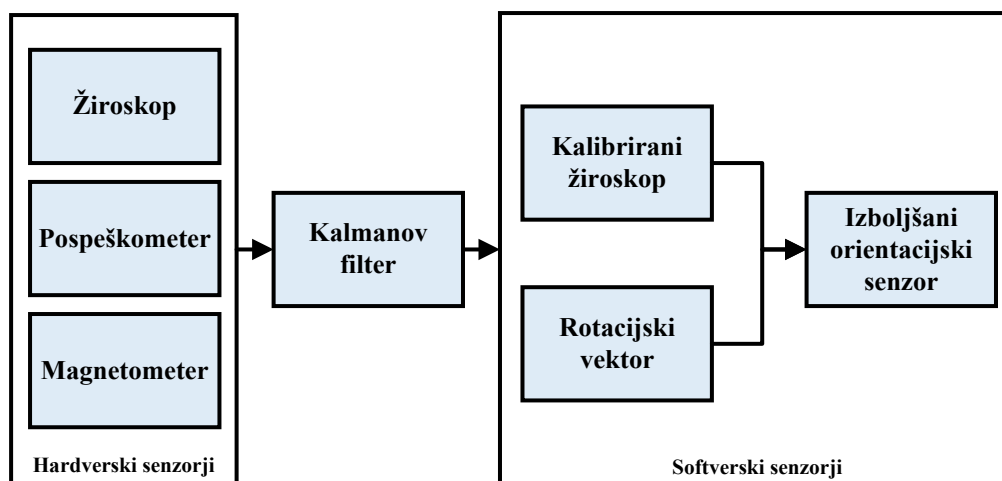
Inercialni senzorji so redko uporabljeni samostojno predvsem zaradi šuma, lezenja (ang. drift) in odstopanja (ang. bias), ki se pojavljajo pri meritvah senzorjev. Napake senzorjev je v veliki meri mogoče eliminirati s fuzijo vseh treh senzorjev. Tako operacijski sistem Android že vključuje več virtualnih senzorjev oz. programske (softverske) implementiranih senzorjev, ki združujejo meritve iz dejanskih (hardverskih) senzorjev. Eden izmed njih je »Rotacijski vektor« (Android Rotation vector), ki preko Kalmanovega filtra [170] združuje pospeškometer, žiroskop in magnetometer v en orientacijski senzor, ki v obliki rotacijskega vektorja (oz. kvaterniona) vrača absolutno orientacijo pametnega telefona glede na svetovni K.S. (slika 4.1).



Slika 4.2: Koordinatni sistem pametnega telefona [169]

Kako učinkovito deluje privzeta fuzija senzorjev, ki je vključena v OS Android pa je odvisno tudi od integriranega vezja MEMS (Microelectromechanical systems), ki je vgrajeno v pametni telefon. Pri preizkusu digitalnega kompasa, ki je temeljil le na Rotacijskem vektorju, smo ugotovili, da je v privzeti fuziji premalo upoštevan žiroskop, saj so na meritve zasuka zelo vplivale magnetne motnje v okolici. To se je opazilo predvsem pri močnejši magnetni motnji, ki se razprostira na večjem področju testne poti znotraj laboratorija. V tem primeru je orientacija počasi lezla proti napačni vrednosti, ki jo je vračal magnetometer. Zato smo se odločili združiti meritve Rotacijskega vektorja in kalibriranega žiroskopa, ki ima nizko stopnjo lezenja. Kalibriran žiroskop vrača zelo natančne zasuke in je hkrati tudi zelo odziven. Težava

pa je v tem, da vrača le relativne zasuke. Namen naše fuzije je bil povečati vpliv žiroskopa, da bi na ta način dobili še bolj natančen virtualni orientacijski senzor. Shema uporabljene fuzije je prikazana na sliki 4.3.



Slika 4.3: Fuzija podatkov iz Rotacijskega vektorja in kalibriranega žiroskopa

Ker žiroskop meri spremembo kota (kotno hitrost), mora biti izhod iz senzorja integriran, da bi dobili relativni zasuk. Nato iz žiroskopa dobljeni kot zapišemo v obliki kvaterniona, pri čemer je rezultat iz rotacijskega senzorja že v obliki kvaterniona. Kvaternion je določen z osjo vrtenja in kotom zasuka okoli te osi. Najbolj enostaven način za združitev dveh kvaternionov (interpolacija) je metoda Quaternion SLERP (Spherical Linear Interpolation) [171], pri kateri z utežmi določimo prispevek posameznega kvaterniona. V našem primeru smo želeli povečati vpliv žiroskopa, vpliv Rotacijskega vektorja pa ohraniti le do te mere, da izniči morebitno lezenje žiroskopa. Tako smo žiroskopu dodelili utež, ki je blizu vrednosti ena (npr. 0,998). Rotacijski vektor se uporabi tudi pri določanju začetnega absolutnega zasuka telefona glede na svetovni koordinatni sistem. Na začetku sta torej oba kvaterniona inicializirana na enako vrednost. To pomeni, da je njun skalarni produkt enak ena. Če se eden izmed vektorjev hipno spremeni, potem se spremeni tudi skalarni produkt (se zmanjša). V primeru, da skalarni produkt pade pod določen prag (npr. 0,8), potem zasuk določa le žiroskop. To se zgodi v primeru magnetne motnje, ko magnetometer vrača napačen zasuk naprave. V primeru tresenja lahko napačne zasuke vrača tudi žiroskop. Zato je v primeru, ko je skalarni produkt obeh vektorjev dalj časa (parameter, ki ga nastavimo)

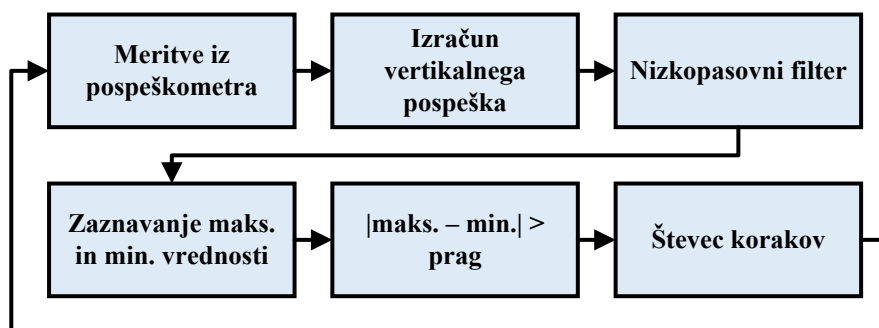
manjši od določenega praga, potrebno oba zasuka ponovno inicializirati na vrednost Rotacijskega vektorja.

Rezultat fuzije, ki je izražen s kvaternionom, lahko pretvorimo v rotacijsko matriko ter v Eulerjeve kote [172]. Za namen digitalnega kompasa je pomemben le zasuk okoli osi Z (svetovnega K.S.), t.i. kot sukanja (ang. yaw). Za pretvorbo iz kvaternionov v rotacijsko matriko in Eulerjeve kote so bile uporabljene metode, ki so privzeto vključene v Android SDK.

4.2 Pedometer

Pri INS za pešce se lahko pri določanju dolžine prepotovane poti izognemo dvojemu integriranju pospeška z uporabo števca korakov. Na ta način lahko ob znani dolžini korakov bolj natančno izračunamo dolžino prepotovane poti, saj akumulirana napaka (zaradi netočne dolžine koraka) ne narašča tako hitro kot pri dvojnemu integralu.

Pedometer je števec korakov, ki v večini primerov temelji zgolj na pospeškometru, lahko pa za svoje delovanje uporablja tudi druge senzorje z namenom večje zanesljivosti detektiranja korakov. V naši implementaciji se tako poleg pospeškometra uporabljata še žiroskop in magnetometer, ki omogočata, da izračunamo pospešek le v vertikalni smeri ne glede na orientacijo pametnega telefona. Signal vertikalnega pospeška (v smeri osi Z svetovnega K.S.) namreč najboljše opisuje gibanje med hojo oz. vsebuje največ informacij, iz katerih je možno natančno določiti, ali je uporabnik naredil nov korak. Slika 4.4 na diagramu prikazuje ključne elemente algoritma za detekcijo koraka.



Slika 4.4: Shema pedometra

Vertikalni pospešek dobimo tako, da upoštevamo rotacijsko matriko R_{WS} (velikosti 3×3), ki jo določa popravljeni Rotacijski vektor. Slednjega smo opisali pri digitalnem kompasu, kjer se uporablja pri določanju orientacije naprave. Ker ta matrika določa preslikavo med svetovnim K.S. (W), ki je prikazan na sliki 4.1 in K.S. telefona (S), ki je prikazan na sliki 4.2, lahko izračunamo pospešek v smeri Z osi svetovnega koordinatnega sistema (slika 4.2) kot:

$$a_z = [R_{WS}(3,1) \ R_{WS}(3,2) \ R_{WS}(3,3)][a_x \ a_y \ a_z]^T, \quad (4.2)$$

kjer so a_x , a_y in a_z pospeški v smeri posameznih osi K.S. telefona. Oblika signala vertikalnega pospeška je nekoliko odvisna od položaja pametnega telefona (ali ga uporabnik drži v roki ali je pritrjen na telo itd.) in od načina hoje uporabnika. Vendar ne glede na te vplive želimo, da je oblika signala za nadaljnjo obdelavo (detekcijo lokalnih ekstremov) čim bolj podobna sinusnemu signalu. Poleg tega, da senzor že sam vrača signal, ki vsebuje določen šum, se na signalu opazijo tudi hitra nihanja naprave med hojo (slika 4.5 zgoraj). Ker se želimo teh visokofrekvenčnih motenj znebiti, smo uporabili nizkopasovni filter z mejno (ang. cut-off) frekvenco 2 Hz in na ta način dobili signal, ki je primeren za detekcijo lokalnih maksimumov in minimumov (slika 4.5 spodaj).

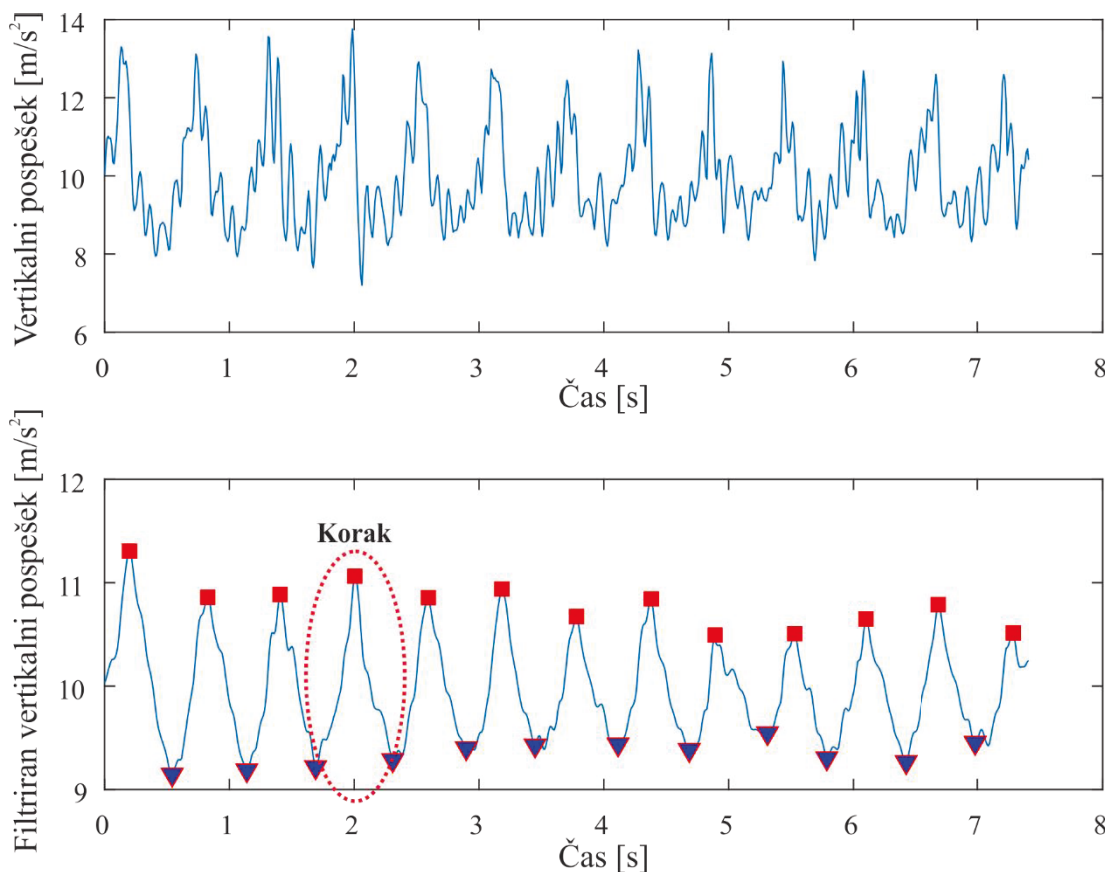
Kot lahko opazimo na sliki 4.5 spodaj, filtrirani signal lokalno skoraj povsem monotonno narašča in pada. Torej lahko lokalne ekstreme dobimo tako, da preverjamo, kdaj je prišlo do spremembe predznaka odvoda signala. Maksimalne vrednosti zaznavamo tako, da preverjamo, ali je signal prešel iz področja monotonega naraščanja na področje monotonega padanja. Minimalne vrednosti pa zaznavamo tako, da preverjamo, ali je signal prešel iz področja monotonega padanja na področje monotonega naraščanja. Če je posamezen ekstrem preblizu prejšnjemu, ga izpustimo.

Posamezne korake zaznavamo na območju, kjer signal narašča od minimalne do maksimalne vrednosti. Pri tem se preverja, ali je absolutna vrednost razlike med maksimalno in minimalno vrednostjo večja od izbranega praga:

$$|a_{maks} - a_{min}| > prag. \quad (4.3)$$

V primeru, da ta pogoj velja, algoritem zabeleži nov korak. V literaturi so opisani tudi druge metode za detekcijo korakov, kot je npr. metoda ZUPT (Zero Velocity Update) [173] in metode, ki temeljijo na frekvenčni analizi signala pospeškometra. V našem primeru informacije pridobljene s frekvenčno analizo (FFT - Fast Fourier

Transformation) – t.i. število korakov na sekundo ne pripomorejo k zanesljivosti detekcije korakov. Ker je v primeru naše uporabe pameten telefon ves čas pritrjen na telo, je detekcija korakov zanesljiva in sprotne frekvenčne analize predstavlja le dodatno računsko obremenitev.



Slika 4.5: Zgoraj je prikazan neobdelan vertikalni pospešek in spodaj filtriran vertikalni pospešek

4.2.1 Ocenjevanje dolžine koraka

Da bi lahko uporabili pedometer kot del inercialnega navigacijskega sistema, je potrebno poznati tudi dolžino posameznega koraka. Najbolj enostaven način pri določanju dolžine koraka je ta, da privzamemo, da je dolžina koraka ves čas enaka. Leppakoski in drugi [168] so predlagali, da se dolžina koraka spreminja linearno glede na povprečni interval koraka (večja kot je frekvenca korakov, daljši je korak). V našem primeru smo uporabili pristop z modelom, ki opisuje dolžino koraka glede na

maksimalno in minimalno vrednost vertikalnega pospeška pri trenutno detektiranem koraku [172]:

$$L = K \sqrt[4]{a_{maks} - a_{min}}, \quad (4.4)$$

kjer sta a_{maks} in a_{min} maksimalni oz. minimalni vertikalni pospešek. K je konstantna vrednost, ki je različna za vsakega uporabnika pedometra. To konstanto lahko izračunamo z upoštevanjem vizualne odometrije ob zagonu aplikacije. Torej, če so izpolnjeni pogoji, da pedometer lahko zaznava korake in iz vizualne odometrije dobimo dolžino posameznega koraka, ki je znotraj pričakovanih mej (npr. med 0,3 m in 1 m), potem lahko izračunamo konstanto K za vsak korak po enačbi:

$$K = L / \sqrt[4]{a_{maks} - a_{min}}, \quad (4.5)$$

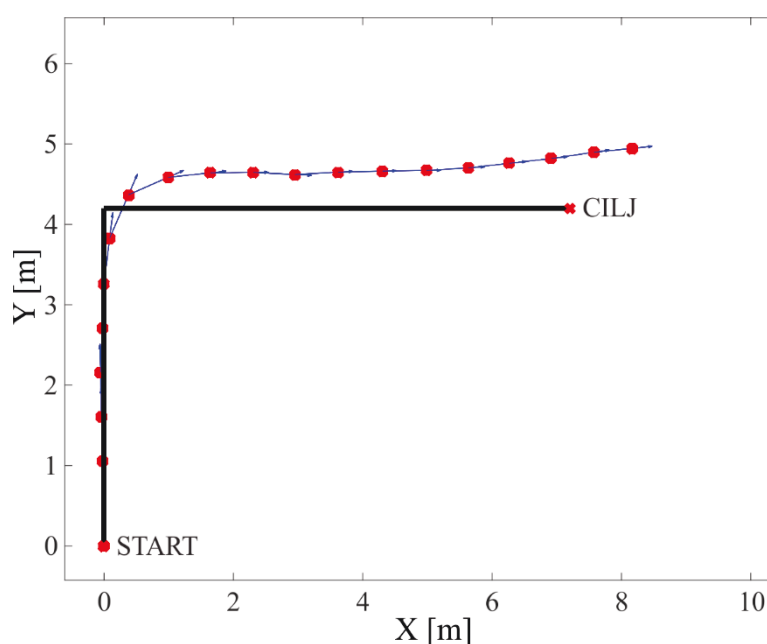
kjer je L dolžina koraka določena z vizualno odometrijo kot: $L_k = \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}$. Pri tem sta (x_k, y_k) in (x_{k-1}, y_{k-1}) položaja pametnega telefona v 2D svetovnem K.S. (opisanem pri vizualni odometriji) ob času k in $k - 1$. Po npr. petih korakih lahko izračunamo povprečno vrednost konstante K , ki jo nato uporabljamo pri izračunu dolžine koraka po enačbi (4.4). Ta postopek nastavljanja parametra K se lahko med uporabo aplikacije večkrat ponovi in se na ta način popravi vrednost konstante na novo povprečno vrednost.

4.3 Rezultati inercialnega navigacijskega sistema

Za namen preizkusa inercialnega navigacijskega sistema, ki temelji na tehniki PDR, smo na pešca pritrdili pametni telefon, na katerem sta se izvajala algoritma za detekcijo korakov (pedometer) in izboljšani digitalni kompas. Med testom je pešec hodil po poti, ki je s črno črto prikazana na sliki 4.6. Z uporabo modela (4.1) sta se ob vsakem novem koraku na pametnem telefonu izračunala nov položaj in zasuk pešca (glede na začetni položaj in orientacijo). Rezultat relativne lokalizacije lahko vidimo na sliki 4.6, kjer rdeče pike označujejo položaj, modre puščice pa orientacijo.

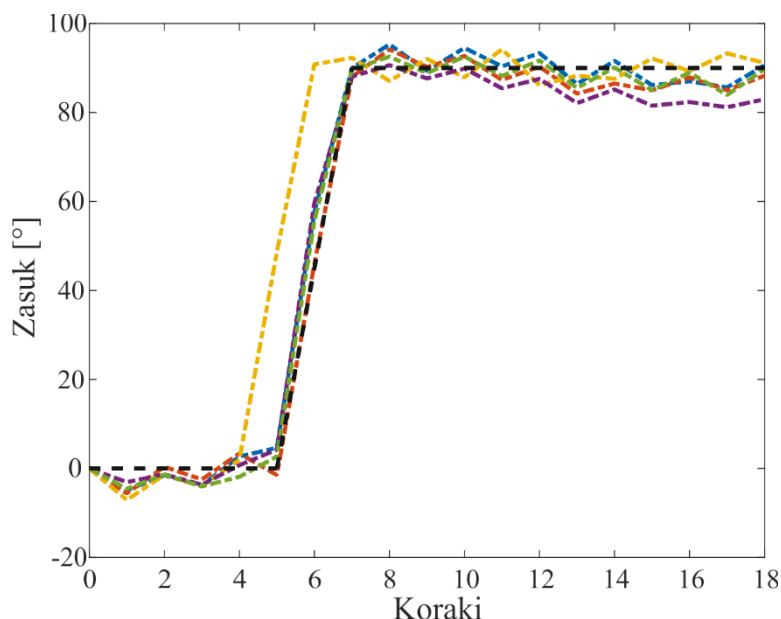
Kot lahko vidimo na sliki 4.6, prvih nekaj korakov rdeče pike (položaji pešca) povsem ležijo na referenčni poti, nato pa se zaradi akumulacije napak (predvsem zaradi netočne ocene dolžine koraka) vse bolj odmikajo od dejanskega položaja. Namreč, ob začetku testa (prvih pet korakov) so dolžine korakov zelo natančno določene z upoštevanjem vizualne odometrije (preko katere se izvaja kalibracija

modela 4.4), nato pa z modelom (4.4) ob vsaki novi oceni dolžine koraka le povečamo napako položaja. Ob koncu poti je izračunani položaj oddaljen od dejanskega položaja za 0,8 m, kar pomeni, da imamo 7 % napake glede na celotno dolžino poti (11,4 m). Prispevek digitalnega kompasa h končni napaki položaja je manjši kot prispevek pedometra, saj meri absolutno orientacijo, kar pomeni, da se napaka pri določanju zasuka ne povečuje s časom. Vpliv nenatančno določenih zasukov (na položaj) se prične povečevati le v primeru magnetne motnje, ko določanje zasukov temelji le na žiroskopu, ki pa je, kot vemo, podvržen lezenju.



Slika 4.6: Rezultat določanja položaja pešca z uporabo inercialnega navigacijskega sistema

Na sliki 4.7 lahko vidimo spreminjanje zasuka pešca (glede na začetno orientacijo) med hojo po referenčni poti prikazani na sliki 4.6. Eksperiment smo petkrat ponovili. Če opazujemo le nekaj začetnih in nekaj končnih korakov (kjer dejanski zasuk 0° oz. 90° označuje črtkana črta), vidimo, da izboljšani digitalni kompas relativno natančno in ponovljivo določa zasuk pešca. Na zadnjem delu poti je prisotna močna magnetna motnja (ta običajen kompas, ki temelji le na magnetometru in pospeškometru, obrne v povsem napačno stran), ki jo je izboljšani digitalni kompas med preizkušanjem uspešno zaznal in posledično zmanjšal vpliv Rotacijskega vektorja (to je mehki senzor) ter povečal vpliv žiroskopa pri določanju zasukov.

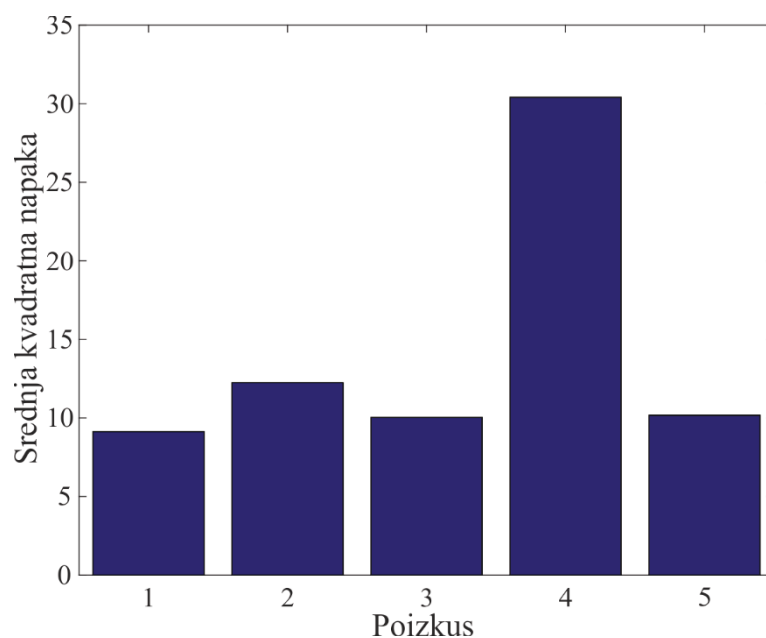


Slika 4.7: Spreminjanje zasuka (določenega z INS) pri hoji po poti, ki je prikazana na sliki 4.6 (eksperiment je bil petkrat ponovljen)

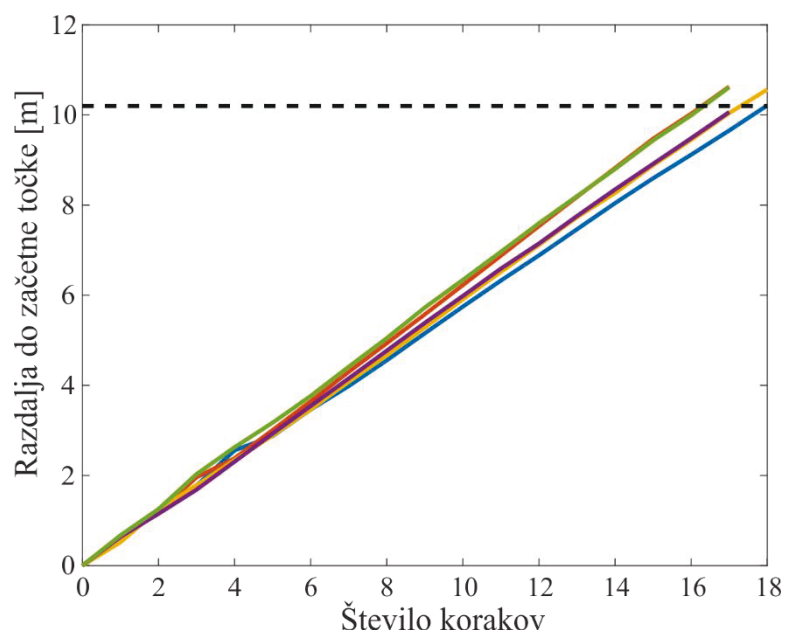
Pri upoštevanju prvih pet in zadnjih osem korakov (na sliki 4.7) lahko izračunamo srednjo kvadratno napako zasuka za vseh pet testov. Rezultat izračuna je prikazan na sliki 4.8. Vidimo lahko, da je le pri četrtem preizkusu digitalni kompas napravil nekoliko večjo napako (zaradi magnetne motnje in lezenja žiroskopa), kar je mogoče opaziti tudi na sliki 4.7 (vijolična krivulja je ob zaključku poti odmaknjena za okoli 8° od pravega zasuka).

Da bi lahko nekoliko bolj natančno določili napako pri izračunu dolžine poti z uporabo pedometra, smo izvedli test, pri katerem je pešec petkrat prehodil ravno pot dolžine 10,2 m. Na sliki 4.9 lahko vidimo naraščanje razdalje do začetne točke v odvisnosti od števila korakov (za vseh pet ponovitev testa). Med testom smo dodatno (ročno) šteli korake, da bi videli, ali pedometer zazna vse napravljene korake. Ugotovili smo, da se štetje korakov izvaja povsem pravilno, saj se pametni telefon zaradi fiksne pritrditve na telo trese manj, kot če bi ga držali v roki in posledično napačnih detekcij korakov skorajda ni. Na sliki 4.9 vidimo, da pri dveh poizkusih razdalja narašča nekoliko hitreje kot pri ostalih treh, saj je pri teh dveh poizkusih pešec naredil en korak manj. To pomeni, da je pri dveh poizkusih v povprečju delal nekoliko daljše korake (0,6 m) kot pri ostalih treh (0,57 m). Natančnih napak pri oceni dolžine korakov ne moremo podati, saj nimamo zelo natančnega referenčnega sistema (v nadaljevanju sledi primerjava z vizualno odometrijo), lahko pa ocenimo skupno

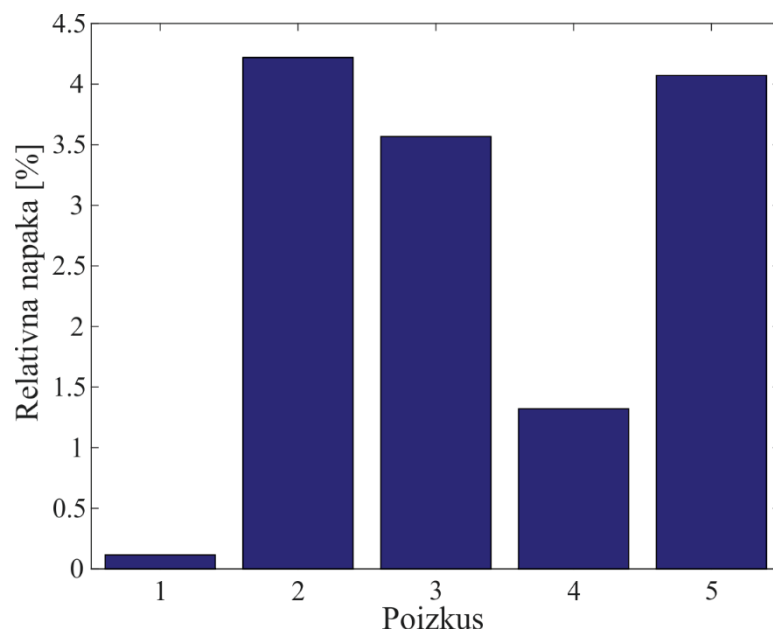
napako glede na končno razdaljo, ki jo vrne inercialni navigacijski sistem. Na sliki 4.10 je prikazana relativna napaka pri izračunu prehojene razdalje s pedometrom. Napaka pri nobenem poizkusu ni bila večja od 5 %, kar je zadovoljiv rezultat.



Slika 4.8: Srednja kvadratna napaka pri določanju zasuka z digitalnim kompasom



Slika 4.9: Razdalja do začetne točke pri hoji po ravni črti dolžine 10,2 m (eksperiment je bil petkrat ponovljen)



Slika 4.10: Relativna napaka pri izračunu prehojene razdalje s pedometrom

5 Fuzija vizualne odometrije in inercialnega navigacijskega sistema

Fuzija je združevanje meritev (podatkov) iz več različnih senzorjev ali tudi metod z namenom, da bi dosegli bolj natančne in zanesljive informacije kot v primeru individualne uporabe senzorjev in metod. V tej študiji je bil cilj združiti informacije pridobljene z vizualno odometrijo in inercialnim navigacijskim sistemom z uporabo razširjenega Kalmanovega filtra (Extended Kalman Filter – EKF) [168, 174]. Uporabljena je bila tehnika, ki temelji na tesno sklopljenem Kalmanovem filtru (ang. tightly coupled Kalman filter) [175], kar pomeni, da se vizualna odometrija izvaja v odvisnosti od inercialnega navigacijskega sistema. Izhod iz razširjenega Kalmanovega filtra se preko povratne zanke upošteva pri nadaljnjih izračunih vizualne odometrije. Inercialni navigacijski sistem (kompas in pedometer) se uporablja kot napoved stanj v fazi predikcije, vizualna odometrija (transformacija iz trenutnega K.S. v svetovni K.S.) pa kot popravek v fazi korekcije. Ker je Kalmanov filter računsko učinkovit, je primeren za implementacijo na pametnem telefonu. Razširjeni Kalmanov filter se na pametnem telefonu izvaja na svoji niti, tako da se izvajanje vizualne odometrije ne upočasni ob fuziji informacij.

Fuzijo vizualne odometrije in inercialnega navigacijskega sistema smo izvedli z namenom, da bi dosegli bolj natančno in robustno določanje položaja uporabnika pametnega telefona. Vizualna odometrija je še posebej pomembna za določanje translacij (za oceno dolžine prepotovane poti), saj model za oceno dolžine korakov prav tako temelji na vizualni odometriji (za oceno konstante K).

Pri vizualni odometriji se je izkazalo, da se težave pojavljajo predvsem v primeru hitrih zasukov, ko se sledi značilnih točk izgubijo zaradi zameglitve slike. V tem primeru se digitalni kompas izkaže za zelo koristnega pri popravljanju nenatančnih

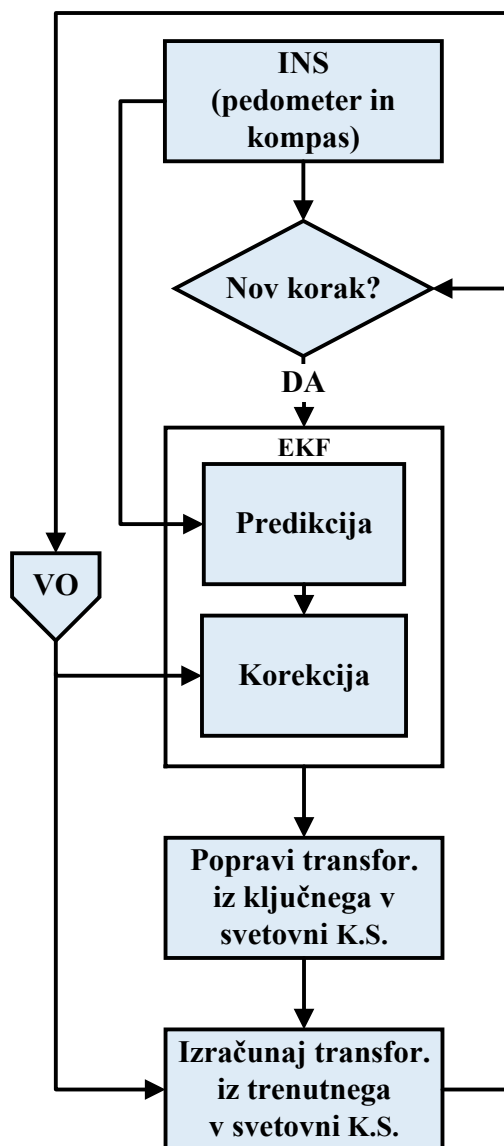
zasukov pridobljenih z vizualno odometrijo. V primeru, ko je kamera obrnjena proti podlagi, ki ima zelo monotono teksturo, sledilnik značilnih točk težko določa sledi, saj je značilnih točk malo. To pomeni, da je tudi sledi malo in od teh je veliko tudi napačno določenih (npr. pri gibanju naravnost se pojavi sled, ki ima smer levo ali desno). Zato v takih primerih želimo, da se z upoštevanjem inercialnega navigacijskega sistema izvaja predikcija položaja in zasuka pametnega telefona, kar pripomore k temu, da napačni rezultati vizualne odometrije manj vplivajo na končno oceno položaja uporabnika. Inercialni navigacijski sistem ni tako natančen pri računanju dolžine prepotovane poti kot vizualna odometrija, vendar pa lahko zadovoljivo kompenzira napako, ki bi se drugače zelo povečala v primeru odpovedi vizualne odometrije. S fuzijo zasukov, ki jih določata vizualna odometrija in digitalni kompas, pa je mogoče še dodatno zmanjšati vpliv lezenja žiroskopa ter magnetnih motenj. Tako vizualna odometrija kot inercialni navigacijski sistem sodita v kategorijo navigacij »na slepo« (ang. dead reckoning), kar pomeni, da napaka položaja s časom raste (napaka se akumulira). S fuzijo obeh sistemov napake položaja ne moremo povsem eliminirati, lahko pa zmanjšamo hitrost naraščanja napake. Na drugi strani nam digitalni kompas vrača absolutno orientacijo (saj fuzija upošteva magnetometer), kar pomeni, da nam napaka zasuka ne narašča s časom.

Ključni koraki fuzije so prikazani na sliki 5.1, kjer je mogoče opaziti, da se združevanje informacij izvede, ko pedometer sporoči, da je bil zaznan nov korak. Kot prikazuje shema, se izhod inercialnega navigacijskega sistema uporabi v fazi predikcije, izhod vizualne odometrije pa v fazi korekcije razširjenega Kalmanovega filtra. Kot rezultat fuzije dobimo popravljeno transformacijo iz ključnega v svetovni K.S. Ta se upošteva pri izračunu novih transformacij iz trenutnega v svetovni K.S. zatem, ko se v algoritmu VO dodajo nove sledi. Dodajanje novih sledi se izvede po vsakem zaznanem koraku (ali ko število aktivnih sledi pade pod določen prag).

Ocenjevanje položaja uporabnika pametnega telefona, ki temelji na tehniki PDR, se začne izvajati glede na začetne koordinate x_0 in y_0 ter začetni zasuk θ_0 . Nov zasuk θ_k in položaj v 2D prostoru (x_k, y_k) se nato izračunavata z nelinearnim modelom stanj (state-space model):

$$\begin{bmatrix} \theta_k \\ x_k \\ y_k \end{bmatrix} = \begin{bmatrix} \theta_{k-1} + \Delta\theta_k \\ x_{k-1} + l_k \cos\theta_k \\ y_{k-1} + l_k \sin\theta_k \end{bmatrix}, \quad (5.1)$$

kjer je $\Delta\theta_k$ sprememba kota med korakom v trenutku $k - 1$ in k ter l_k je trenutna dolžina koraka. Ker je model stanja nelinearen, navaden Kalmanov filter ni primeren in zato je bil uporabljen razširjeni Kalmanov filter [175].



Slika 5.1: Shema fuzije vizualne odometrije in inercialnega navigacijskega sistema z razširjenim Kalmanovim filtrom

Elementi vektorja stanja $\mathbf{x}_k = [x_1 \ x_2 \ x_3]^T$ so naslednji: $x_1 = \text{zasuk}$, $x_2 = x \text{ koordinata}$, $x_3 = y \text{ koordinata}$. Filtriranje oz. fuzija z razširjenim Kalmanovim filtrom se prične glede na začetno oceno stanja $\hat{\mathbf{x}}_0$ in začetno kovarianco $\mathbf{P}_0$. Začetna ocena stanja se izračuna ob prvem koraku z VO glede na trenutno

transformacijo iz trenutnega v svetovni K.S. Model gibanja v fazi predikcije razširjenega Kalmanovega filtra napoveduje, kje naj bi bil uporabnik v trenutku k .

Tako se vektor stanj v fazi predikcije izračuna kot:

$$\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}_{k-1} + \begin{bmatrix} \Delta\theta_k \\ l_k \cos \hat{x}_{1_{k-1}} \\ l_k \sin \hat{x}_{1_{k-1}} \end{bmatrix}, \quad (5.2)$$

kjer $\hat{\mathbf{x}}_{k-1}$ predstavlja a posteriori oceno stanja, ki se izračuna po korekciji z meritvijo v trenutku $k - 1$. $\hat{\mathbf{x}}_k^-$ predstavlja a priori oceno stanja za k -ti časovni trenutek ter $\hat{x}_{1_{k-1}}$ predstavlja predhodno a posteriori oceno zasuka. Spremembo kota $\Delta\theta_k$ določa razlika med trenutno vrednostjo iz kompasa in predhodno a posteriori oceno zasuka $\hat{x}_{1_{k-1}}$. Dolžina koraka l_k pa se izračuna z modelom (4.4), ki je opisan v sklopu pedometra.

Matrika prehajanja stanj $\mathbf{F}_k$, ki je potrebna za izračun nove a priori kovariance, je pridobljena s parcialnim odvodom enačbe (5.2):

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & 0 \\ -l_k \sin \hat{x}_{1_k}^- & 1 & 0 \\ l_k \cos \hat{x}_{1_k}^- & 0 & 1 \end{bmatrix}. \quad (5.3)$$

Kovariančna matrika šuma procesa $\mathbf{Q}_k$ se izračuna ob vsakem novem koraku kot:

$$\mathbf{Q}_k = \text{diag} \left(\begin{bmatrix} \sigma_{\Delta\theta}^2 \\ \cos^2(\hat{x}_{1_k}^-) \sigma_l^2 \\ \sin^2(\hat{x}_{1_k}^-) \sigma_l^2 \end{bmatrix} \right), \quad (5.4)$$

kjer je $\sigma_{\Delta\theta}^2$ varianca meritve zasuka z digitalnim kompasom in σ_l^2 varianca ocene dolžine korakov s pedometrom. V predikcijskem koraku se izračuna a priori kovariančna matrika $\mathbf{P}_k^-$ kot:

$$\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k, \quad (5.5)$$

kjer je $\mathbf{P}_{k-1}$ a posteriori kovarianca iz predhodnega časovnega koraka.

Vektor meritev $\mathbf{z}_k = [\theta_{VO_k} \ x_{VO_k} \ y_{VO_k}]^T$ sestoji iz kota sukanja θ_{VO_k} (ang. yaw) ter koordinat x_{VO_k} in y_{VO_k} , ki jih določa vizualna odometrija preko transformacije v 2D prostoru iz trenutnega v svetovni koordinatni sistem. Merilna (izhodna) matrika $\mathbf{H}$ je enaka enotski matriki velikosti 3×3 : $\mathbf{H} = \mathbf{I}_{3 \times 3}$. Merilna posodobitev vektorja stanj $\hat{\mathbf{x}}_k$ in izračun kovariančne matrike $\mathbf{P}_k$ se izvedeta na podlagi naslednjih enačb:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_k^-), \quad (5.6)$$

$$\mathbf{P}_k = (\mathbf{I}_{3 \times 3} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^-, \quad (5.7)$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1}, \quad (5.8)$$

kjer je $\mathbf{R}$ kovariančna matrika merilnega šuma (ki vsebuje variance napak ocene kota in položaja z vizualno odometrijo) in $\mathbf{I}_{3 \times 3}$ enotska matrika. $\mathbf{K}_k$ predstavlja ojačanje Kalmanovega filtra. Kovariančna matrika $\mathbf{R}$ je diagonalna matrika, ki je določena z eksperimentalno ocenjenimi variancami napak zasuka in položaja. Po vsakem koraku lahko na podlagi trenutnega vektorja stanj $\hat{\mathbf{x}}_k$ izračunamo popravljeno transformacijo iz ključnega K.S. v svetovni K.S. Ta se potem upošteva pri vizualni odometriji do naslednjega koraka za izračun transformacij iz trenutnega K.S. v svetovni K.S.

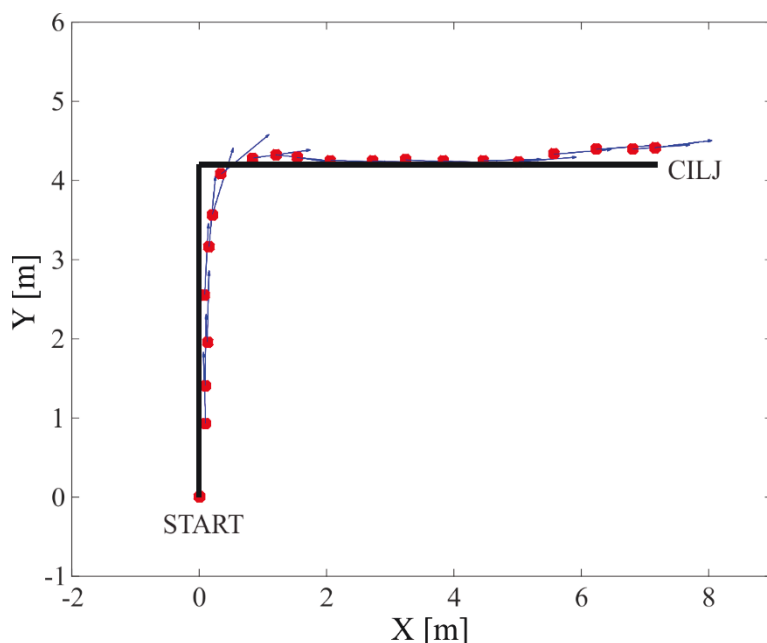
5.1 Rezultati fuzije vizualne odometrije in inercialnega navigacijskega sistema

Pri prejšnjih eksperimentih smo videli, da tako vizualna odometrija kot tudi inercialni navigacijski sistem lahko delujeta samostojno in v določenih primerih ponujata povsem zadovoljive rezultate pri določanju relativnega položaja in zasuka pešca (uporabnika pametnega telefona) glede na začetni položaj in orientacijo. Ker imata oba lokalizacijska pristopa svoje prednosti in slabosti (ki smo jih omenili pri opisu njunih preizkusov), smo se odločili, da ju z namenom povečanja natančnosti in robustnosti določanja položaja ter zasuka združimo preko razširjenega Kalmanovega filtra. Cilj fuzije je bil združiti prednosti vizualne odometrije, ki omogoča natančno določanje inkrementalnih premikov, ter prednosti digitalnega kompasa, ki omogoča natančno določanje zasukov.

Pri implementaciji fuzijskega algoritma (ki temelji na razširjenem Kalmanovem filtru) na pametnem telefonu smo temu z namenom, da omogočimo sočasno izvajanje vizualne odometrije, inercialnega navigacijskega sistema ter njune fuzije, dodelili svojo nit (ang. thread), ki se lahko (v primeru večjedrnega procesorja) izvaja na svojem procesorskem jedru. V našem primeru smo za namen testiranja uporabljali pametna telefona, ki imata vgrajen štiri oz. osemjedrni procesor.

Na sliki 5.2 lahko vidimo rezultat fuzije vizualne odometrije in inercialnega navigacijskega sistema preko razširjenega Kalmanovega filtra. Če dobljeno trajektorijo poti pešca primerjamo z referenčno potjo, ki je označena s črnimi črtami (slika 5.2), lahko ugotovimo, da so odstopanja zelo majhna, in sicer le nekaj

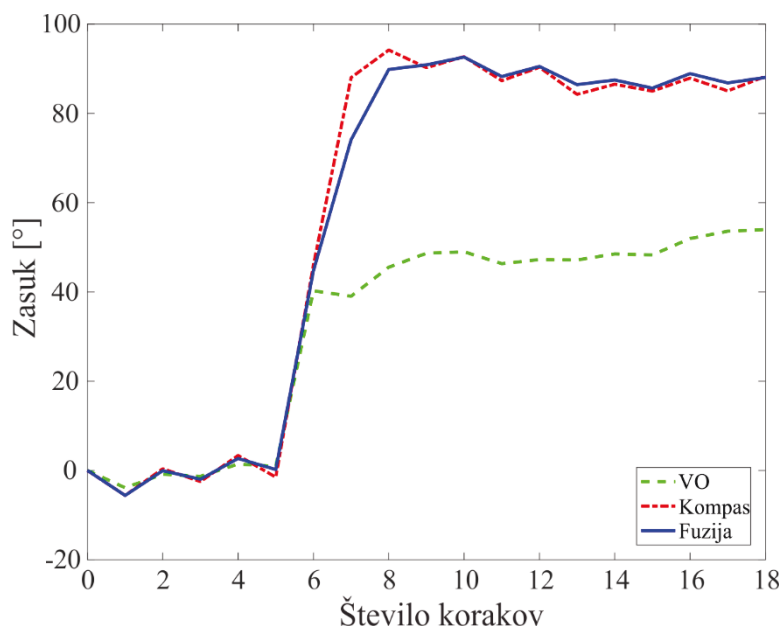
centimetrov. Pri tem moramo upoštevati dejstvo, da tudi pešec ne more povsem natančno slediti referenčni črti, ki je označena na tleh.



Slika 5.2: Določanje trajektorije gibanja pešca s fuzijo vizualne odometrije in inercialnega navigacijskega sistema

S fuzijo vizualne odometrije in inercialnega navigacijskega sistema v veliki meri izboljšamo natančnost določanja zasukov, pri čemer ima digitalni kompas ključno vlogo. Na sliki 5.3 lahko vidimo, da bi bil zasuk pri uporabi le vizualne odometrije (zelena krivulja) od šestega koraka dalje (na poti, ki je prikazana na sliki 5.2) povsem napačno določen.

S fuzijo z razširjenim Kalmanovim filtrom dosežemo sprotno popravljanje zasuka, ki ga upoštevamo tudi pri nadaljnjih izračunih položaja z vizualno odometrijo. To pomeni, da s predikcijo zasuka, določenega z uporabo digitalnega kompasa, izločimo (oz. zmanjšamo njihov prispevek) napačno določene zasuke vizualne odometrije, ki bi v nasprotnem primeru drastično vplivali na natančnost relativne lokalizacije. Na sliki 5.3 vidimo, da se modra krivulja, ki prikazuje zasuke pešca, določene s fuzijo vizualne odometrije in digitalnega kompasa, dobro ujema z referenčnim zasukom. Ta je prvih pet korakov enak 0° in zadnjih deset korakov 90° , vendar moramo upoštevati, da se pešec med hojo ne uspe povsem držati referenčnega zasuka (še posebej ne med zavojem).

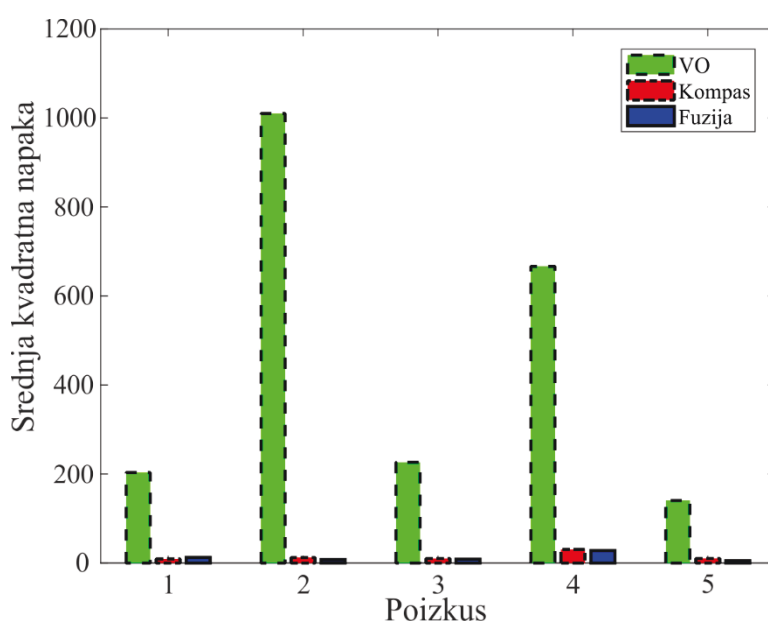


Slika 5.3: Spreminjanje zasuka pešca glede na njegovo začetno orientacijo med hojo po referenčni poti, ki je prikazana na sliki 5.2

Da bi lahko bolj natančno določili napako pri oceni zasukov s fuzijo vizualne odometrije in inercialnega navigacijskega sistema, smo eksperiment, pri katerem je pešec sledil referenčni poti, prikazani na sliki 5.2, petkrat ponovili. Za primerjavo smo hkrati spremljali tudi napake zasukov pri uporabi le vizualne odometrije oz. digitalnega kompasa. Na sliki 5.4, kjer je prikazana srednja kvadratna napaka zasukov (določenih z vizualno odometrijo, inercialnim navigacijskim sistemom in fuzijo slednjih dveh metod), lahko vidimo, da pri uporabi le vizualne odometrije dobimo zelo velike napake, saj so že v primeru enega napačno določenega zasuka vsi nadaljnji zasuki napačni. Posledično pa tudi napaka položaja le še raste s številom korakov. Ker digitalni kompas (ponovljivo) vrača natančne zasuke, dobimo pri fuziji z razširjenim Kalmanovim filtrom zelo majhne napake zasukov (ki se časovno ne akumulirajo) za vseh pet poizkusov.

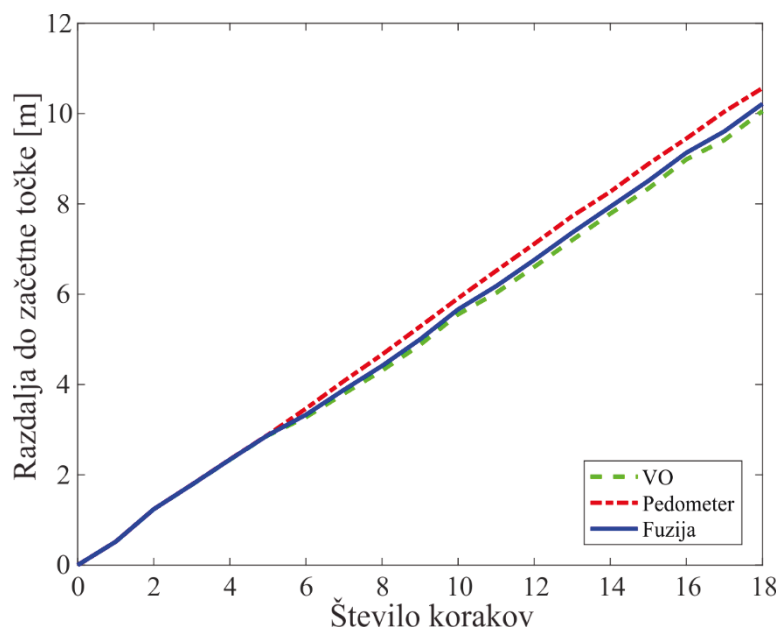
Z namenom, da bi ugotovili, kako fuzija vizualne odometrije in inercialnega navigacijskega sistema vpliva na natančnost določanja inkrementalnih premikov, smo izvedli test, pri katerem je pešec petkrat prehodil ravno pot v dolžini 10,2 m. Po vsakem koraku se je na pametnem telefonu izračunala razdalja do začetne točke z uporabo vizualne odometrije, inercialnega navigacijskega sistema in fuzije obeh metod (slika 5.5). Kot lahko opazimo na sliki 5.5 (ki prikazuje rezultate enega od petih poizkusov), se prvih pet korakov vse tri krivulje povsem prekrivajo, saj se na začetku

pri pedometru izvaja kalibracija modela za oceno dolžine korakov. Po končani kalibraciji vidimo, da razdalja do začetne točke narašča skoraj linearno in približno enako hitro s številom korakov tako v primeru uporabe vizualne odometrije kot pedometra. S fuzijo obeh metod dobimo v tem primeru zelo natančno oceno dolžine prepotovane poti, saj je, kot vidimo na sliki 5.5, praktično enaka 10,2 m (zaradi zaokroževanja iz 10,23 m). Sicer je glavni namen fuzije ublažiti vpliv večje napake, ki bi se potencialno lahko pojavila pri računanju inkrementalnih premikov z vizualno odometrijo.

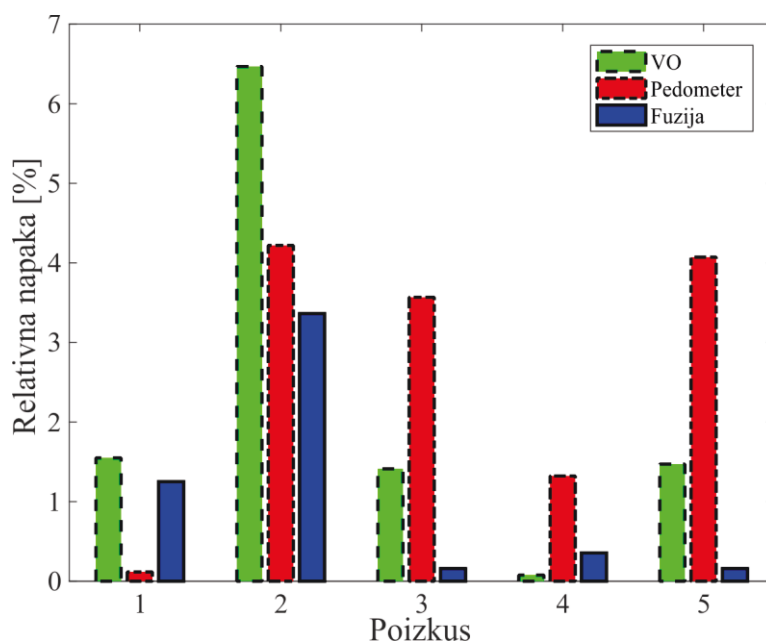


Slika 5.4: Srednja kvadratna napaka pri določanju zasuca z vizualno odometrijo, inercialnim navigacijskim sistemom in fuzijo slednjih dveh algoritmov

Za vseh pet ponovitev eksperimenta in vse pristope računanja razdalje do začetne točke smo izračunali relativno napako, ki je prikazana na sliki 5.6. Relativno napako smo izračunali glede na odstopanje izračunane končne razdalje od dejanske dolžine poti (10,2 m). Pomen fuzije vizualne odometrije in pedometra se pokaže predvsem pri drugem poizkusu (slika 5.6), kjer vizualna odometrija napravi največjo napako (skupaj več kot 6 % poti), ki pa se pri uporabi fuzije zmanjša skoraj za polovico (na 3,5 %). Fuzija obeh relativnih lokalizacijskih metod je zelo pomembna za dolgoročno zmanjšanje hitrosti naraščanja napake, saj pedometer, čeprav ni zelo natančen pri določanju dolžine prehojene poti, lahko v veliki meri zmanjša napako položaja v primeru, ko vizualna odometrija odpove (slabi svetlobni pogoji, monotona podlaga).



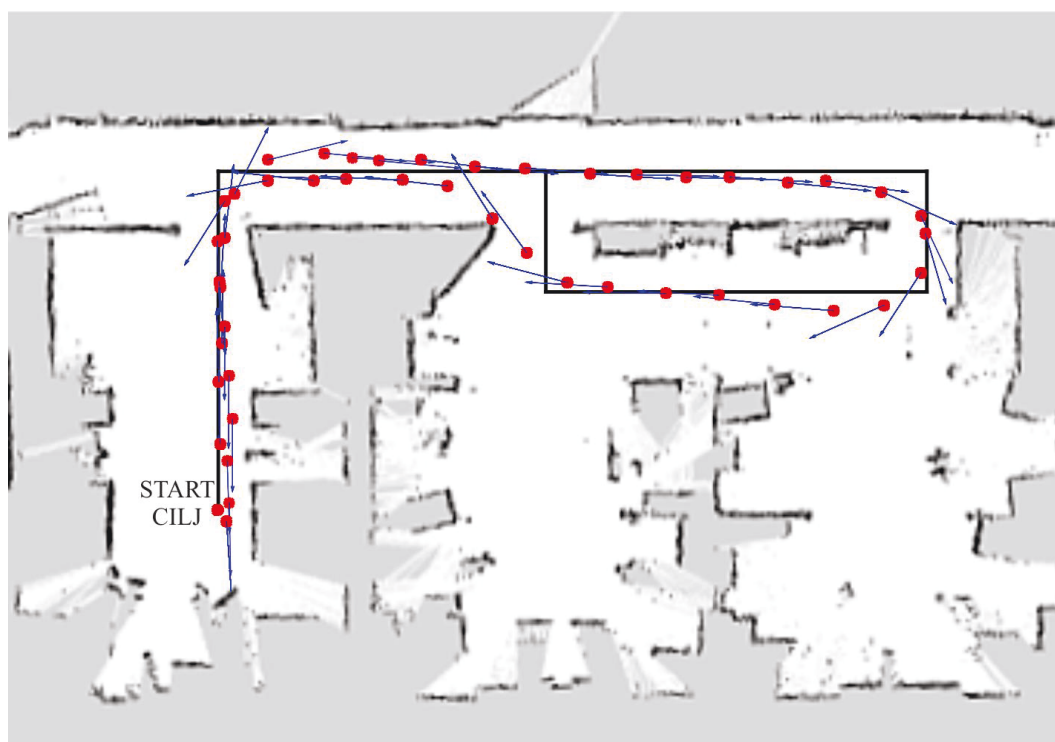
Slika 5.5: Spreminjanje razdalje do začetne točke med hojo po ravni črti dolžine 10,2 m



Slika 5.6: Relativna napaka pri izračunu prehojene razdalje z vizualno odometrijo, inercialnim navigacijskim sistemom in fuzijo slednjih dveh algoritmov

Z namenom, da bi pokazali zmogljivost relativnega lokalizacijskega sistema, ki temelji na fuziji vizualne odometrije in inercialnega navigacijskega sistema, smo naredili eksperiment, pri katerem je pešec prehodil sklenjeno pot (od starta do cilja v smeri urinega kazalca) dolžine 27 m, ki je prikazana na sliki 5.7. Fuzijski algoritem se je izvajal v realnem času na pametnem telefonu in sproti pošiljal položaj ter orientacijo

pešca preko WiFi omrežja v okolje Matlab, kjer smo položaje prikazali na zemljevidu. Testna pot po laboratoriju je bila idealna za preizkus natančnosti in zanesljivosti fuzijskega algoritma, saj je pešec na poti naredil šest zavojev, na večjem delu poti (ki ima obliko pravokotnika) pa je bila prisotna tudi močna magnetna motnja, ki je predstavljala velik izziv za digitalni kompas. Na sliki 5.7 vidimo, da je fuzijski algoritem deloval dobro in izločil vpliv motenj po celotni poti, saj so vse napake ocenjenih položajev manjše od 30 cm.



Slika 5.7: Določanje položajev in zasukov pešca pri hoji po sklenjeni poti s fuzijo vizualne odometrije in inercialnega navigacijskega sistema

6 Lokalizacija z uporabo radijskih signalov

Ker sodobni mobilni aparati omogočajo sprejem različnih radijskih signalov, so se temu primerno uveljavili različni pristopi lokalizacije tako z uporabo mobilnega GSM omrežja, WLAN omrežja, Bluetooth tehnologije, FM signalov, NFC povezave itd. Za namen lokalizacije v notranjem okolju so se najbolj izkazali algoritmi, ki temeljijo na merjenju jakosti WiFi ali Bluetooth signalov. S stališča potrebne infrastrukture ima WiFi omrežje prednost pred Bluetooth omrežjem, saj je v večini zgradb že prisotno. Vendar, ker je gostota WiFi dostopnih točk običajno zelo redka, je na osnovi obstoječih omrežij težko doseči visoko natančnost lokalizacije. Zato se v zadnjem času vse bolj uveljavlja omrežje Bluetooth, ki ima številne prednosti pred omrežjem WiFi [176]: manjša poraba energije tako pri sprejemnikih (pametnih telefonih) kot oddajnikih (baterije lahko napajajo oddajnike več mesecev ali celo let), nizka cena oddajnikov (v prostor lahko namestimo večje število oddajnikov, kar omogoča večjo natančnost lokalizacije), večja robustnost (signali vsebujejo manj šuma), bolj kompaktna oblika, sprejemniki lahko z višjo frekvenco osvežujejo parametre omrežja. Na mobilnih napravah (Android, iOS) je mogoče z uporabo vgrajenih komunikacijskih modulov prebrati parametre omrežja kot sta RSSI (ang. Received Signal Strength Indicator) ter ID oddajnika.

Lokacijo mobilnih naprav je torej mogoče določiti z merjenjem jakosti signalov (padec jakosti je sorazmeren oddaljenosti od postaje), ki jih oddajajo bazne postaje. Najbolj preprost način določanja položaja, ki je hkrati tudi najmanj natančen, temelji na določanju bližine (ang. proximity based method) sprejemnika (do oddajnika) na podlagi jakosti signala [177]. Območje jakosti signala je v tem primeru razdeljeno na tri dele, ki povedo, ali je sprejemnik blizu (»IMMEDIATE«, močan signal), daleč (»FAR«, šibek signal) ali srednje daleč (»NEAR«, srednja jakost signala) od

oddajnika. V primeru, ko imamo na voljo jakosti signalov iz več različnih oddajnikov (iz vsaj treh oddajnikov), lahko dosežemo večjo natančnost lokalizacije z uporabo dveh uveljavljenih pristopov: trilateracija (ki zahteva izgradnjo modelov padanja jakosti signalov) [178] in metoda »prstnih odtisov« (ang. fingerprinting based method) [179].

Pri pristopu s trilateracijo [180] je položaj sprejemnika (uporabnika mobilne naprave) določen preko modelov, ki povedo, kakšne so oddaljenosti od oddajnikov pri izmerjenih jakostih signalov (upoštevata se informacija o padanju jakosti signala z oddaljevanjem od oddajnika). Za vsak oddajnik dobimo krožnico, na kateri naj bi se nahajal sprejemnik (uporabnik pametnega telefona). Z uporabo geometrijske tehnike – trilateracije, se nato poišče le eno točko, v kateri naj bi se nahajal uporabnik mobilne naprave. Za določitev absolutnega položaja (s trilateracijo) moramo predhodno poznati tudi položaj oddajnikov.

Pristop s »prstnimi odtisi« [180] je napram pristopu izgradnje modela jakosti signala veliko bolj enostaven za implementacijo ter hkrati tudi manj občutljiv na šume brezžičnih signalov. Posledično lahko s to metodo dosežemo tudi večjo natančnost pozicioniranja v prostoru [179]. Sistem, ki temelji na uporabi »prstnih odtisov«, je običajno sestavljen iz dveh faz, in sicer faze učenja (»offline«) in faze določanja položaja (»online«). V fazi učenja je cilj zgraditi podatkovno bazo, ki vsebuje meritve jakosti WiFi ali Bluetooth signalov (vseh dostopnih točk oz. oddajnikov) za vse referenčne položaje v prostoru. V fazi ugotavljanja položaja se trenutno izmerjene vrednosti RSSI primerjajo z meritvami v bazi (iskanje najbližjega sosedu) in glede na to, pri kateri referenčni točki se vrednosti RSSI najbolj ujemajo, se določi trenutni položaj.

Lokalizacija z uporabo radijskih (WiFi, Bluetooth) signalov se zdi na prvi pogled enostavna za implementacijo, kar v resnici tudi je, če so naše zahteve glede natančnosti manj striktne (natčnost na nekaj metrov). V primeru, ko želimo samo z uporabo jakosti radijskih signalov doseči visoko natančnost lokalizacije (da so napake manjše od 1 m), pa so potrebni večji napor v smislu iskanja pravega algoritma, ki bo to omogočal. Največji problem povzroča nihanje jakosti signalov zaradi odbojev signalov od ovir (posledica je razširjanje signalov po več poteh, ang. multipath

propagation) [177] in absorpcij signalov (zaradi statičnih in dinamičnih ovir jakost signala slabi).

6.1 Bluetooth

Bluetooth [181] je brezžična komunikacijska tehnologija, ki elektronskim napravam (računalnik, pametni telefon, miška, tipkovnica, slušalke, zvočniki, žarnica itd.) omogoča izmenjavo informacij (podatkov) na kratke razdalje. Bluetooth tako kot WiFi deluje v ISM (ang. industrial, scientific and medical) frekvenčnem pasu 2,4 GHz. Tehnologijo Bluetooth, ki je bila sprva standardizirana kot IEEE 802.15.1, je iznašlo podjetje Ericsson leta 1994. Za specificiranje novih verzij skrbi »Bluetooth Special Interest Group« (SIG), ki združuje več kot 25.000 podjetij iz vsega sveta.

Pri tehnologiji Bluetooth je za prenos informacij na voljo 79 kanalov, kjer prvi kanal uporablja frekvenco 2402 Mhz in zadnji kanal 2480 Mhz. Za preprečitev interference z drugimi RF signali se pri Bluetoothu uporablja metoda adaptivnega frekvenčnega skakanja (ang. adaptive frequency hopping) [182], ki identificira fiksne izvore interference in nato onemogoči problematične frekvenčne kanale za neko časovno periodo. Torej gre pri tej metodi za hitro preklapljanje nosilnega signala (ang. carrier wave) med različnimi frekvenčnimi kanali z uporabo psevdonaključne sekvence. Podatki, ki se pošiljajo v paketkih, se tako enkrat prenašajo po enem, drugič po drugem kanalu.

Za izmenjavo informacij morajo Bluetooth naprave najprej vzpostaviti medsebojno povezavo. Ena naprava (»master«) se lahko hkrati poveže s sedmimi drugimi napravami (»slave«) ter z njimi simultano komunicira (dvosmerno). Maksimalna razdalja, pri kateri lahko Bluetooth naprave še vzpostavijo povezavo, je odvisna od njihove izhodne moči ter drugih faktorjev, kot so npr. odboji signalov zaradi prisotnosti ovir (stene, pregrade, ljudje itd.). Bluetooth naprave delimo v tri razrede glede na njihovo maksimalno izhodno moč oz. maksimalni doseg signalov (tabela 6.1).

Prenos podatkov z uporabo Bluetooth povezave je možen le v primeru, ko sta dve napravi predhodno seznanjeni (ang. paired) in povezani. Za primere (kot je npr. lokalizacija z uporabo jakosti Bluetooth signalov ali branje meritev z brezžičnega senzorja), kjer potrebujemo le enosmerno komunikacijo (oddajanje informacij v okolico), je običajna Bluetooth povezava manj primerna.

Tabela 6.1: Bluetooth razredi

Razred	Maks. izhodna moč	Doseg
1	20 dBm	do 100 m
2	4 dBm	do 10 m
3	0 dBm	do 1 m

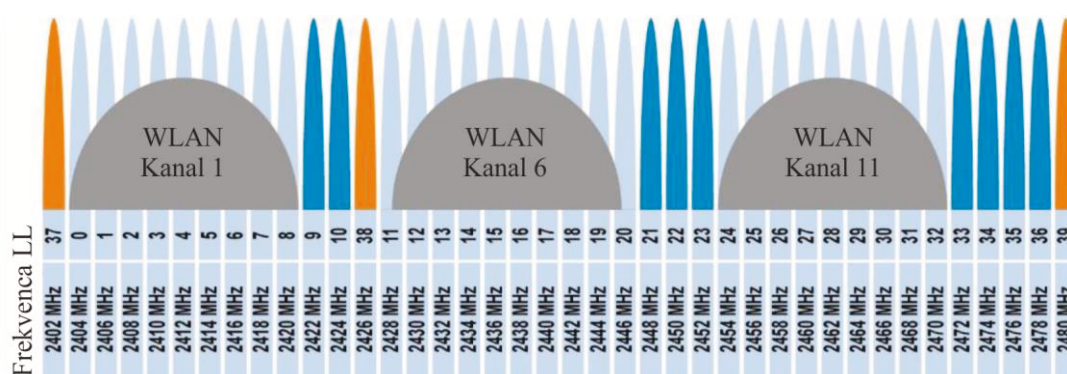
Zaradi potreb po brezžični komunikaciji, ki ne zahteva seznanjanja elektronskih naprav ter hkrati potroši zelo malo energije, je prišlo do razvoja novega Bluetooth standarda (verzija 4.0), in sicer Bluetooth Low Energy (BLE) [183].

V zadnjem času so predvsem na področju zdravja, športa ter interneta stvari (internet of things - IoT) vse bolj popularni brezžični senzorji (to so npr. pedometer, merilnik srčnega utripa, brezžični termometer), ki neprestano oddajajo določene informacije (meritve), brez da bi se na njih morali povezati. Da bi brezžični senzorji imeli dolgo avtonomijo in bili hkrati čim manjši (baterija mora biti majhna), morajo biti vse komponente kar se da varčne. Za njihovo brezžično komunikacijo z ostalimi napravami se je kot najbolj primeren izkazal Bluetooth Low Energy, ki je poznan tudi pod imenom Bluetooth Smart (za naprave, ki le oddajajo) oz. Bluetooth Smart Ready (za gostitelje oz. »host« naprave). Gostiteljske naprave (PC, pametni telefon), ki imajo vgrajen Bluetooth verzije 4.0 ali novejši, delujejo v dvojnem načinu (»dual mode«), kar pomeni, da so lahko hkrati povezani s klasično Bluetooth napravo (npr. Bluetooth slušalkami) in BLE napravo (npr. zapestnica z merilnikom srčnega utripa). BLE naprave, ki se uporabljajo le za oddajanje informacij (npr. brezžični senzorji), pa delujejo v enojnem načinu (»single mode«, ki omogoča le BLE), kar jim omogoča nizko porabo energije.

6.1.1 Bluetooth Low Energy

Bluetooth Low Energy (BLE) je bil predstavljen leta 2010 kot del protokola Bluetooth ver. 4.0 [183]. BLE je bil zasnovan z namenom enostavne komunikacije (za prenos majhne količine podatkov) med elektronskimi napravami, ki imajo omejen vir energije (uporabljajo baterijsko napajanje). V primerjavi s klasičnim Bluetooth-om, BLE uporablja manjše število kanalov za prenos informacij, in sicer namesto 79 jih uporablja le 40, vendar so ti dvakrat širši (širina posameznega kanala je 2 MHz). Pri

BLE se uporablja modulatorska shema GFSK (Gaussian Frequency Shift Keying) z modulatorskim indeksom 0,5, ki omogoča nižjo stopnjo bitnih napak ter večji doseg signalov. BLE v teoriji omogoča prenos podatkov do hitrosti 1 Mbit/s, vendar so v praksi te hitrosti precej nižje (manjše od 0,3 Mbit/s). Za oglaševanje (ang. advertising, neprekinjeno oddajanje oglasnih paketkov) se uporabljajo kanali 37, 38 in 39, saj se na ta način v večji meri izognemo interferenci z najbolj pogosto uporabljenimi WiFi kanali 1, 6 in 11 (glej sliko 6.1). Vsi ostali BLE kanali pa se uporabljajo za prenos podatkov pri vzpostavljeni povezavi med dvema napravama.



Slika 6.1: BLE frekvenčni kanali

Tako kot klasične Bluetooth naprave, tudi BLE naprave lahko delujejo pri različno nastavljenih izhodnih močeh, glede na zahtevan doseg signalov [183]. Nizka izhodna moč (med -30 dBm in -12 dBm) se lahko uporablja v primeru, ko je razdalja med oddajnikom in sprejemnikom majhna (manj kot 15 m), visoka moč (med -8 dBm in 4 dBm) pa je izbrana, ko je zahtevan doseg signalov večji od 20 m (pri nastavljeni moči 4 dBm je doseg signalov okoli 70 m, če med oddajnikom in sprejemnikom ni ovir). BLE podpira maksimalno izhodno moč 10 dBm (10 mW), pri kateri naj bi bil (teoretično) doseg signalov okoli 300 m v primeru, da ni ovir med oddajnikom in sprejemnikom. S povečevanjem izhodne moči imajo BLE naprave tudi večjo porabo energije, kar posledično pomeni krajšo avtonomijo pri baterijskem napajanju. BLE ima v primerjavi s klasičnim Bluetooth-om pri enaki izhodni moči večji doseg (v tabeli 6.2 so podatki za BLE oddajnik podjetja Kontakt.io [184]), saj uporablja bolj učinkovito modulatorsko tehniko (GFSK).

Tabela 6.2: Možne izhodne moči BLE oddajnika in doseg signalov [184]

TX Power level	Maks. izhodna moč	RSSI pri oddal. 1 m	Doseg [m]
0	-30 dBm	-115 dBm	2
1	-20 dBm	-84 dBm	4
2	-16 dBm	-81 dBm	10
3	-12 dBm	-77 dBm	20
4	-8 dBm	-72 dBm	30
5	-4 dBm	-69 dBm	40
6	0 dBm	-65 dBm	60
7	4 dBm	-59 dBm	70

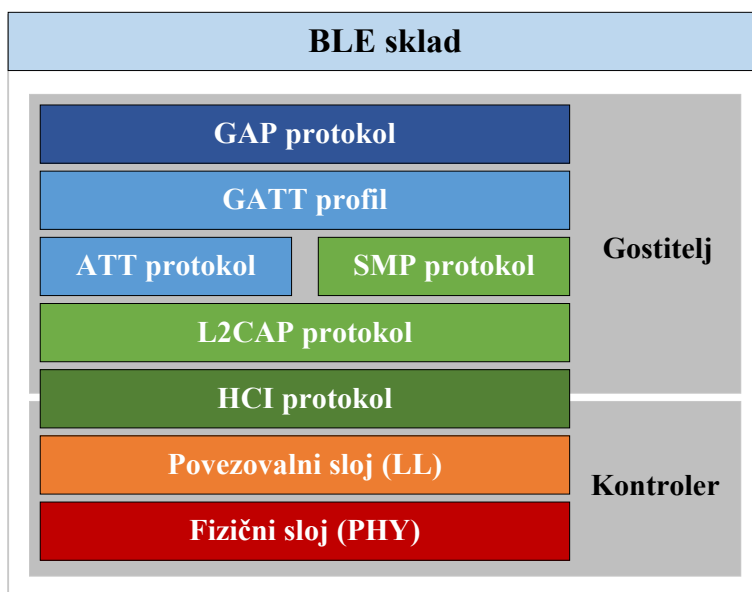
Bistvena prednost BLE v primerjavi s klasičnim Bluetooth-om je poleg nizke porabe energije tudi ta, da lahko naprave, ki imajo vgrajen BLE, le oddajajo (ang. *broadcasting*) informacije v okolico, brez da bi bilo potrebno z njimi vzpostaviti povezavo (kot je to potrebno pri klasični Bluetooth povezavi). Pri uporabi BLE gre torej v večini primerov za enosmerno komunikacijo, kjer sprejemnik le čaka na nove pakete iz oddajnika. Povezavo med dvema BLE napravama je potrebno vzpostaviti le v primeru, ko želimo na t.i. »slave« napravi (BLE oddajniku) spremeniti določene parametre (jo na novo konfigurirati), naložiti nov firmware ali prenesti večjo količino podatkov (npr. meritve iz več senzorjev). BLE naprava, ki jo konfiguriramo, ima vlogo »slave« in je lahko hkrati povezana le z eno »master« BLE napravo. »Master« BLE naprava pa je lahko hkrati povezana na več »slave« BLE naprav. Zaradi varnosti je dostop do BLE oddajnikov pogosto zaščiten z geslom.

BLE ima v primerjavi s klasičnim Bluetooth-om drugače zasnovan sklad (ang. *stack*) [185], zaradi česar tudi nista medsebojno kompatibilna. Na sliki 6.2 lahko vidimo, da je BLE sklad v osnovi sestavljen iz dveh delov, in sicer gostitelja (»host«) ter kontrolerja (»controller«) (kar je enako tudi pri klasičnem Bluetoothu). Kontroler predstavlja fizični sloj (»physical layer«) in povezovalni sloj (»link layer«). Kontroler je implementiran s strojno opremo, ki običajno temelji na SOC (System-on-chip) rešitvi. Gostitelj sestoji iz zgornjih slojev BLE sklada, v katerih se izvajajo naslednji programsko rešeni servisi:

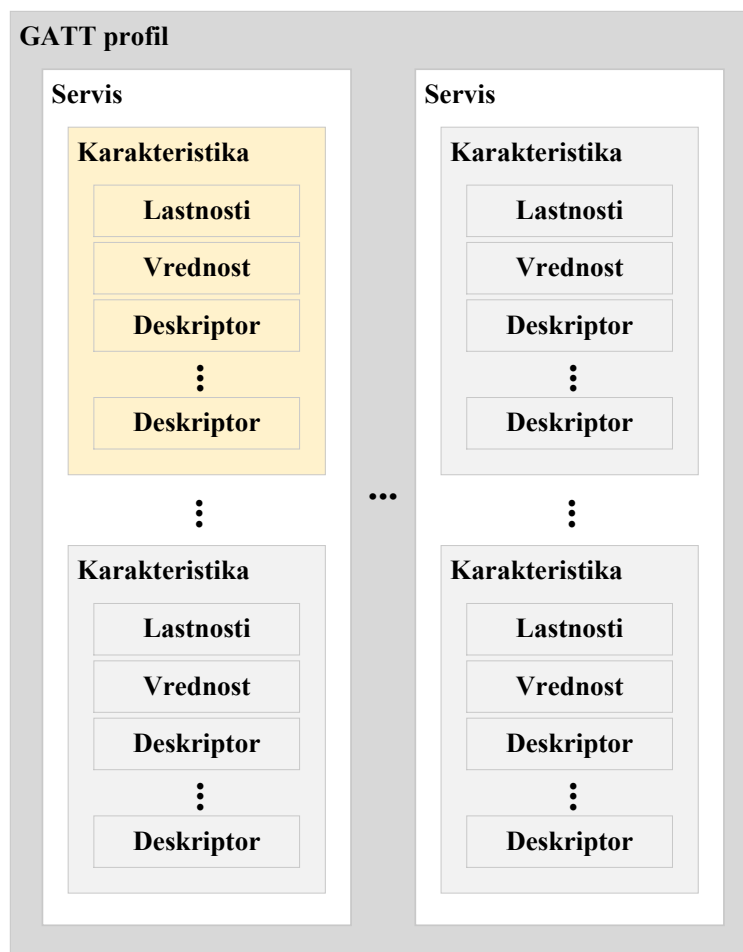
- »Logical link control and adaption protocol« (L2CAP): L2CAP sloj skrbi za multipleksiranje podatkov med zgornjim delom (gostiteljem) in spodnjim delom BLE sklada (kontrolerjem). Za samo komunikacijo med kontrolerjem in gostiteljem pa skrbi standardiziran HCI (host controller interface) protokol.
- »Attribute protocol« (ATT): ATT protokol se uporablja za odkrivanje servisov ter za pošiljanje atributov (servisi in pripadajoče karakteristike znotraj GATT profilov) med dvema BLE napravama, ki medsebojno komunicirata. Posamezen atribut lahko opišemo kot podatkovno strukturo, ki jo uporablja GATT profil za pošiljanje in prejemanje podatkov. ATT definira štiri polja podatkov: »handle« (naslov atributa), »type« (UUID oznake povedo, kaj vrednosti predstavljajo), »value« (dejanska vrednost atributa) in »permissions« (določa, ali lahko atribut beremo ali zapišemo).
- »Security manager protocol« (SMP): SMP protokol skrbi za izmenjavo varnostnega ključa in enkripcijo pri vzpostavljanju povezave. Povezava se vzpostavi po več zaporednih izmenjavah podatkov.
- »Generic attribute profile« (GATT): GATT profil je pomemben del BLE sklada, saj določa osnovo profila BLE naprave [186]. Sestavljen je iz več servisov (npr. če ima naprava več senzorjev, potrebujemo poleg primarnih servisov še za vsak senzor po en servis) in pripadajočih karakteristik (slika 6.3), ki imajo svoje oznake UUID (Universally Unique Identifier). Polna oznaka UUID ima lahko največ 128 bitov, vendar se običajno uporabljajo krajše oblike s 16 ali 32 biti. Servise in pripadajoče karakteristike, ki opisujejo določen GATT profil, skupaj imenujemo tudi atributi. Servisi so opisani s karakteristikami, njihovo vsebino (ki je sestavljena iz lastnosti in pripadajočih vrednosti ter deskriptorjev, s katerimi so podane še dodatne informacije o izbrani karakteristiki, npr. merska enota) in obliko ter načinom dostopa in varnosti. Servisi GATT profila se delijo v dve skupini, in sicer primarni in sekundarni servisi. Primarni servisi (so fiksni) skrbijo za osnovne funkcije (serijska števila, stanje baterije itd.), sekundarni pa za ostale funkcije, ki jih definira načrtovalec naprave. GATT profil posamezne BLE naprave mora obvezno implementirati dve vlogi, in sicer vlogo odjemalca (GATT client) in strežnika (GATT server), saj drugače ta naprava ne more biti vidna ostalim

BLE napravam. GATT strežnik sprejema zahteve, ukaze in potrditve preko ATT profila, ki jih pošilja GATT odjemalec. GATT strežnik odgovarja na zahteve GATT odjemalca s tem, da mu pošilja obvestila (asinhrono), ko je pripravljen (npr. ko so na voljo nove meritve s senzorja). GATT omogoča odjemalcem branje in pisanje atributov, ki jih ponuja server BLE naprave. Preko UUID-ja lahko odjemalec poišče točno določen servis na strežniku ali točno določeno karakteristiko posameznega servisa (in jo prebere ali spremeni). Odjemalec je običajno »master« BLE naprava, strežnik pa »slave« BLE naprava. BLE profili, ki so standardizirani s strani organizacije Bluetooth SIG, predpisujejo obliko podatkov pri komunikaciji z aplikacijskim slojem (to je najvišji sloj, ki se nahaja nad gostiteljskim delom BLE sklada). Standardni profili so oblikovani glede na želene funkcionalnosti (npr. profil za srčni utrip, profil za določanje bližine itd. [183]). Za opis standardnih GATT profilov se uporabljajo vrednosti atributov, ki so preddefinirane za vsak standarden profil posebej.

- »Generic access protocol« (GAP): GAP protokol definira postopke za seznanjanje (ang. pairing) in povezovanje BLE naprav. Postopki so generični, tako da lahko aplikacijski sloj implementira različne BLE načine delovanja (način oddajanja, opazovalni način, centralni način in periferni način).



Slika 6.2: BLE sklad

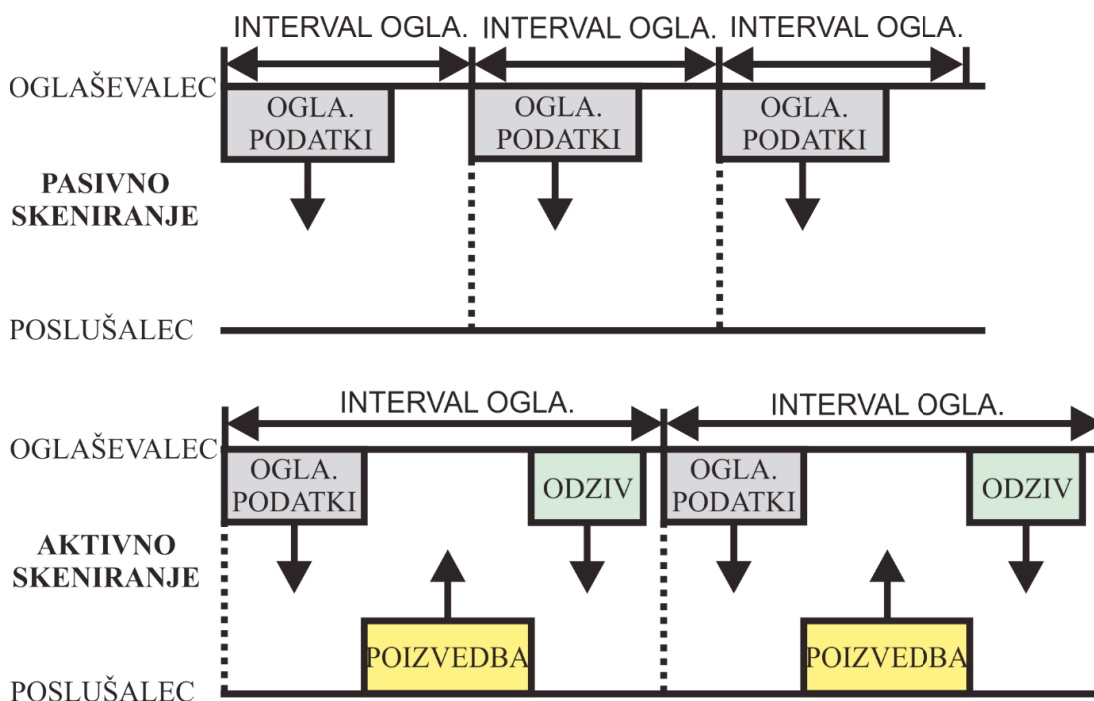


Slika 6.3: Hierarhija GATT profila

BLE naprava lahko deluje na štiri različne načine: način oddajanja (»broadcast«), centralni način, periferni način in opazovalni način. Način oddajanja se uporablja v primeru, ko naprava le oddaja informacije (npr. brezžični senzor), opazovalni način pa se uporablja, ko naprava le sprejema informacije. Pri uporabi teh dveh načinov delovanja napravi medsebojno nista seznanjeni. Centralni način delovanja in periferni način delovanja pa omogočata vzpostaviti BLE povezavo, ki je najbolj podobna klasični Bluetooth povezavi, saj se v tem primeru vzpostavi dvosmerna komunikacija med »master« BLE napravo (ki podpira centralni način delovanja) in eno ali več »slave« BLE napravami (ki podpirajo periferni način delovanja). Kontroler, ki je del BLE sklada, omogoča, da naprava lahko deluje tako v »master« kot v »slave« načinu (nekateri BLE naprave ne podpirajo obeh načinov).

BLE specifikacije definirajo dva različna pristopa za odkrivanje oz. iskanje povezljivih BLE naprav, in sicer sta to aktivno in pasivno skeniranje (slika 6.4) [177].

Pri pasivnem skeniranju centralna naprava pasivno posluša na treh kanalih (37, 38 in 39), predvidenih za oglaševanje (advertising) in zajema PDU (protocol data unit) paketke, ki jih pošiljajo druge BLE naprave (oglaševalci, ki so v načinu oddajanja). Pri aktivnem skeniranju centralna BLE naprava (tako kot pri pasivnem skeniranju) najprej čaka na PDU paketke drugih naprav. Ko te paketke sprejme, iz njih prebere informacije o tem, katere načine delovanja podpirajo BLE naprave, ki so poslale te paketke. Če PDU paketek iz določene BLE naprave pove, da je ta naprava povezljiva ali da jo je možno dodatno skenirati, potem lahko centralna naprava pošlje nov (»scan request«) paketek oz. poizvedbo, ki od te naprave zahteva več informacij (odziv).



Slika 6.4: Pasivno (zgoraj) in aktivno (spodaj) skeniranje

Na začetku vsakega oglaševanja (ki se periodično ponavlja) oglaševalec pošlje paketek, ki pove, za kakšen tip oglaševanja gre. Po BLE specifikacijah obstajajo štirje tipi oglaševalskih paketkov:

- **ADV_NONCONN_IND** (non-connectable advertisement): Ta tip uporabljajo naprave, ki samo pošiljajo oglaševalske paketke z določenimi podatki in ne pričakujejo odzivov s strani poslušalcev.

- **ADV_DISCOVER_IND** (discoverable advertisement): Ta tip paketkov vsebuje podatke in pove, da je možno z dodatno poizvedbo (request) s strani poslušalca pridobiti dodatne informacije.
- **ADV_IND** (advertisement): Ta tip paketkov pove, da je na voljo več podatkov, ki jih je možno pridobiti z dodatno poizvedbo ter, da je možno s to napravo vzpostaviti povezavo.
- **ADV_DIRECT_IND** (directed advertisement): Ta tip paketkov se uporablja, ko naprava (ki izvaja oglaševanje) želi vzpostaviti povezavo s točno določeno »master« BLE napravo, ki skenira oglaševalske pakete. S tem paketkom se torej pošlje zahteva za povezavo na točno določeno BLE napravo.

BLE naprava, ki posluša oz. skenira oglaševalske pakete, se lahko različno odzove glede na tip teh paketkov: 1. lahko ignorira te pakete, 2. lahko prebere podatke iz teh paketkov ter jih posreduje v gostiteljski del sklada, 3. lahko se odzove na prejete pakete in oglaševalski napravi pošlje dodatno poizvedbo ali z njo vzpostavi povezavo (pošlje zahtevo za povezavo).

Kmalu po prihodu novega standarda Bluetooth 4.0, je bilo kaj hitro jasno, da je bil BLE zasnovan tako, da je njegov potencial mogoče izkoristiti tudi za namen lokalizacije v notranjem okolju [187]. Številna podjetja so razvila kompaktne BLE oddajnike (ang. beacons) z baterijskim napajanjem, ki neprestano v okolico pošiljajo pakete z določenimi informacijami (glede na konfiguracijo oddajnika).

6.1.2 BLE oddajniki

BLE oddajniki (beacons) so kompaktne naprave (slika 6.5), ki prvenstveno niso bile razvite za namen določanja položaja uporabnikov v notranjem okolju, saj je bil njihov glavni namen uporabe oglaševanje (»fizični splet« [188], spremljanje navad kupcev) in podajanje kontekstnih informacij z določanjem bližine uporabnika (mobilne naprave). Z dodanimi senzorji lahko ti oddajniki uporabniku pošiljajo tudi druge informacije, npr. temperaturo in vlago v prostoru. Pri BLE oddajnikih se uporablja način oddajanja (»broadcast«), pri katerem se periodično (z izbranim intervalom) v okolico pošiljajo paketi (običajno tipa »ADV_IND«) z določeno obliko in vsebino. Ti oddajniki običajno uporabljajo baterijsko napajanje (pogosto se uporablja baterija

v obliki kovanca, ki zavzame zelo malo prostora), saj zaradi BLE tehnologije porabijo zelo malo energije (npr. s 1000 mAh baterijo lahko delujejo več let). Optimalna poraba energije je dosežena z nizko stopnjo obratovalnega cikla (ang. duty cycle), kar pomeni, da naprava večino časa spi, saj se zbudi le v trenutku, ko mora poslati nov paketek. Interval oddajanja paketkov je mogoče nastaviti na vrednosti od 20 ms do 10 s.



Slika 6.5: BLE oddajniki

Podatki, ki se pošiljajo z oglaševalskimi paketki, imajo definirano strukturo glede na BLE specifikacije [183]. Na sliki 6.6 so prikazane vse komponente BLE paketa in njihove velikosti v bajtih (bytes). Maksimalna velikost paketa je lahko 47 bajtov, pri čemer je velikost vseh (uporabnih) podatkov (»broadcast data«) oglaševalskega paketa omejena na 31 bajtov. Zaradi teh omejitev se uporabljajo posebni protokoli BLE oddajnikov, ki natančno predpisujejo obliko podatkov. Struktura paketa z elementi preambula (»preamble«), dostopni naslov (»access address«), koristni podatki (»packet payload«) in »CRC« (ciklično redundančna kontrola ali ang. cyclic redundancy check) je skupna vsem BLE paketkom (oglaševalski paketki imajo v osnovi enako strukturo kot paketki, ki se uporabljajo pri prenosu podatkov, ko je povezava vzpostavljena).

Preambula (1 B)	Dostopni naslov (4 B)	Paket koristnih podatkov (2-39 B)		CRC (3 B)
		Glava (2 B)	Koristni podatki (B)	
		Oglaševalski naslov (6 B)	Oglaševalski podatki (0-31 B)	

Slika 6.6: Podatki BLE oglaševalskega paketa [189]

Preambula je določena z osmimi biti, ki so enaki: 01010101 ali 10101010. Naslov za dostop (ki ni naslov BLE naprave) zavzame 32 bitov in je lahko dveh tipov: oglaševalski naslov za dostop ali podatkovni naslov za dostop. Prvi se uporablja pri oddajanju oglaševalskih paketkov, skeniranju ali pri oddajanju pobude za povezavo in je vedno enak 0x8E89BED6. Drugi naslov se uporablja pri že vzpostavljeni BLE povezavi in spremembah na različnih nivojih povezave.

Paket koristnih podatkov, ki lahko zavzame največ 39 bajtov, je sestavljen iz glave in koristnih podatkov. Vsebina glave, ki zavzame 16 bitov ali 2 bajta, je odvisna od vrste paketa (oglaševalski, podatkovni). Pri oglaševalskem paketu glava določa njegov tip. Koristni podatki (payload) so podatki, ki se dejansko prenašajo med BLE napravami. Lahko so to oglaševalski podatki, odziv na skeniranje ali podatki za izmenjavo informacij pri vzpostavljeni povezavi. Pri oglaševalskih paketkih se koristni podatki razdelijo na MAC naslov naprave (broadcast address, slika 6.6) in same oglaševalske podatke (broadcast data).

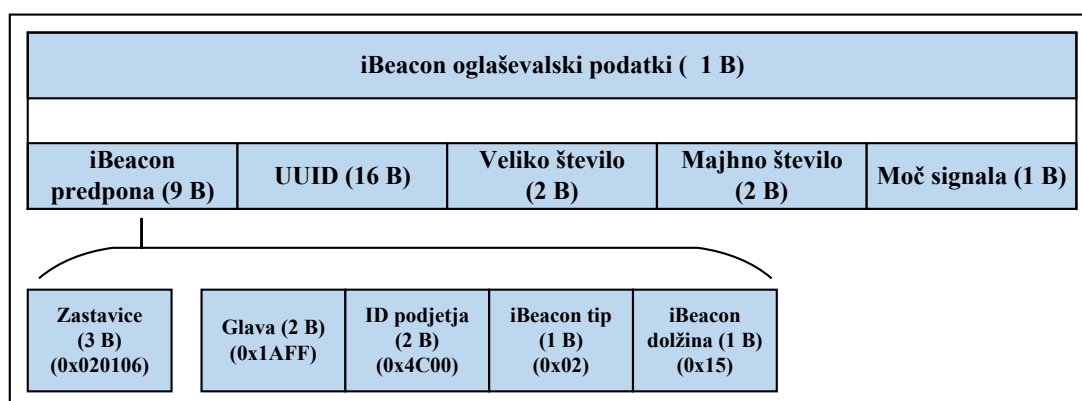
CRC je bolj kompleksen protokol od parnosti in omogoča odkrivanje napak pri prenosu. Namesto enega bita uporablja več bitov, navadno 16 bitov. Pri BLE strukturi paketa je uporabljenih 24 bitov (3 bajti) za ciklično redundančno kontrolo.

Z namenom, da bi BLE oddajniki z oddajanjem oglaševalskih paketkov (omejene velikosti) poslušalcem (BLE sprejemnikom) sporočali čim več uporabnih informacij (jakost signala, izhodna moč itd.), je bilo specificiranih več protokolov, ki določajo obliko in vsebino oglaševalskih paketkov. Pri teh protokolih so koristni podatki (payload) razdeljeni na več segmentov v katerih so podatki zapisani s heksadecimalnimi vrednostmi. Najbolj popularna in razširjena sta protokola »iBeacon« [190] (specificiran leta 2013 s strani podjetja Apple) in »Eddystone« [191] (specificiran leta 2015 s strani podjetja Google). Oba protokola sta kompatibilna tako z Android kot tudi iOS mobilnimi napravami.

Protokol iBeacon je bil prvotno razvit za uporabo na pametnih telefonih in drugih iOS napravah podjetja Apple, ki naj bi v okolico periodično oddajali določene informacije za naprave, ki poslušajo in pričakujejo te informacije. Uporaba iBeacon protokola se je zelo razširila s prihodom kompaktnih iBeacon oddajnikov (slika 6.5), katerih namen je uporaba v notranjem okolju za določanje bližine (proximity) mobilnih naprav in sprožanje določenih akcij (fizični splet, kontekstne informacije).

V zadnjem času iBeacon oddajniki postajajo vse bolj popularni tudi za namen lokalizacije v notranjem okolju in s tem povezane navigacije po muzejih, trgovski centrih, letališčih, bolnišnicah itd.

iBeacon protokol določa vsebino in obliko (31 bajtov) oglaševalskih podatkov (broadcast data), kot je prikazano na sliki 6.7. iBeacon predpona (ang. prefix), ki je velikosti 9 bajtov, vsebuje fiksne podatke ([02 01 06 1A FF 4C 00 02 15]), ki povedo poslušalcem, da oddajnik uporablja iBeacon protokol. UUID je identifikator, ki zavzame 16 bajtov in ga lahko poljubno spreminjamo. Uporablja se kot ID, ki pove, kateremu podjetju (znamki) ali aplikaciji pripadajo oddajniki. Veliko število (»Major number«) je prav tako identifikator velikosti 2 bajta, ki se uporablja za označevanje večje skupine oddajnikov (npr. vsi BLE oddajniki v enem trgovskem centru, ki imajo istega lastnika, imajo enako »major« oznako). Majhno število (»Minor number«) je identifikator, po katerem se ločijo oddajniki znotraj ene skupine. Zadnji bajt znotraj iBeacon podatkov je rezerviran za vrednost »TX Power« (zapisano z dvojiškim komplementom), ki pove, kakšna je moč BLE signala na razdalji 1 m od oddajnika. Proizvajalci BLE oddajnikov te običajno kalibrirajo tako, da je moč signalov (na razdalji 1 m od oddajnika) »TX Power« pri izbranem nivoju moči (»TX Power level«, tabela 6.2) pri vseh oddajnikih enaka.

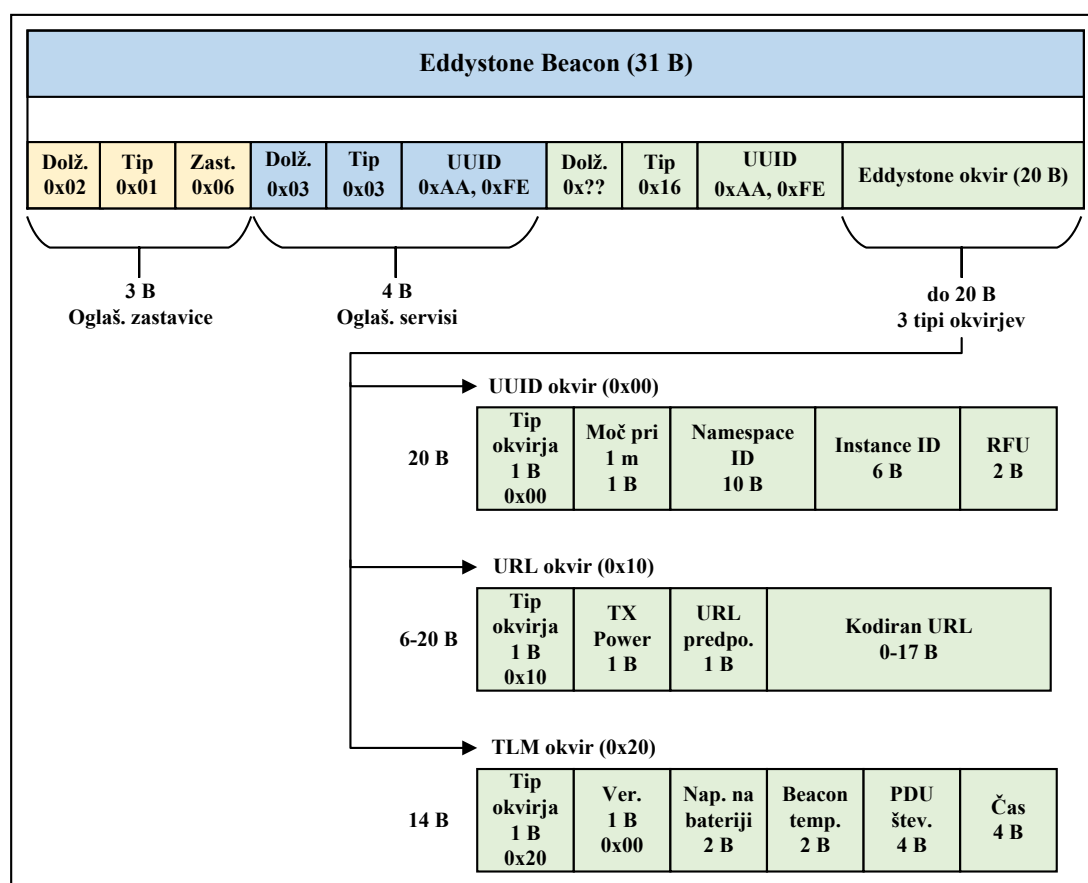


Slika 6.7: Podatki oglaševalskega paketka pri »iBeacon« oddajnikih

Pri BLE oddajnikih iBeacon je tako kot pri vseh ostalih BLE oddajnikih možno nastaviti moč signala (TX Power) in pa interval pošiljanja oglaševalskih paketkov (priporočena vrednost je 100 ms).

Protokol Eddystone [192, 193], ki ga je razvilo podjetje Google, se od protokola iBeacon ne razlikuje veliko, saj Eddystone oglaševalski paketki vsebujejo enake informacije kot iBeacon paketki, le oblika (format) je drugačna. Prednost Eddystone protokola je predvsem odprtokodna licenca.

Za Eddystone BLE oddajnike je značilno, da lahko zaporedno pošiljajo tri oglaševalske paketke [192], ki se razlikujejo po zadnjem delu (okvirju) paketa - »Eddystone Frame« (predstavlja zadnjih 20 bajtov paketa). Te tri dele imenujemo UUID, URL in TLM okvirji (slika 6.8). Začetnih 11 bajtov znotraj uporabnih podatkov (payload) je pri vseh treh Eddystone paketkih enakih, saj predstavljajo predpono, ki poslušalcem pove, kateri protokol je uporabljen (tj. Eddystone).



Slika 6.8: Podatki treh oglaševalskih paketkov pri »Eddystone« oddajnikih

UUID okvir (slika 6.8), ki zavzame 20 bajtov, je sestavljen iz petih delov: tip okvirja, »TX power« (v tem primeru je jakost signala kalibrirana pri razdalji 0 m od oddajnika), »Namespace ID« (ima enak namen kot UUID oznaka pri iBeacon

paketku), »Instance ID« (ima enak namen kot »Major« in »Minor« vrednosti pri iBeacon paketku) in RFU (Reserved for future use).

URL okvir pri drugem tipu Eddystone paketka (slika 6.8) zavzame od 6 do 20 bajtov in je sestavljen iz naslednjih delov: tip okvirja, »TX power« (jakost signala pri 0 m), URL predpona (<http://www>, <http://> itd.) in pa kodiran URL (url, ki lahko zavzame največ 17 bajtov). URL okvir se uporablja kot QR koda, ki uporabniku pametnega telefona pošlje obvestilo in povezavo do poljubnega spletnega mesta (»fizični splet«).

Tretji tip Eddystone paketka vsebuje okvir TLM (slika 6.8), ki zavzame 14 bajtov in je namenjen oddajanju telemetrije Eddystone oddajnika. Telemetrija lahko vključuje: napetost na bateriji, temperaturo oddajnika, stanje števca (ki šteje število oddanih paketkov), čas od vklopa naprej. Ker ta tip paketka ne vsebuje nobenih informacij o ID-ju oddajnika, ga lahko povežemo z drugima dvema tipoma (UUID in URL) Eddystone paketkov le preko MAC naslova oddajnika.

6.2 Izgradnja modelov jakosti signalov

6.2.1 Meritve jakosti signalov z uporabo pametnega telefona

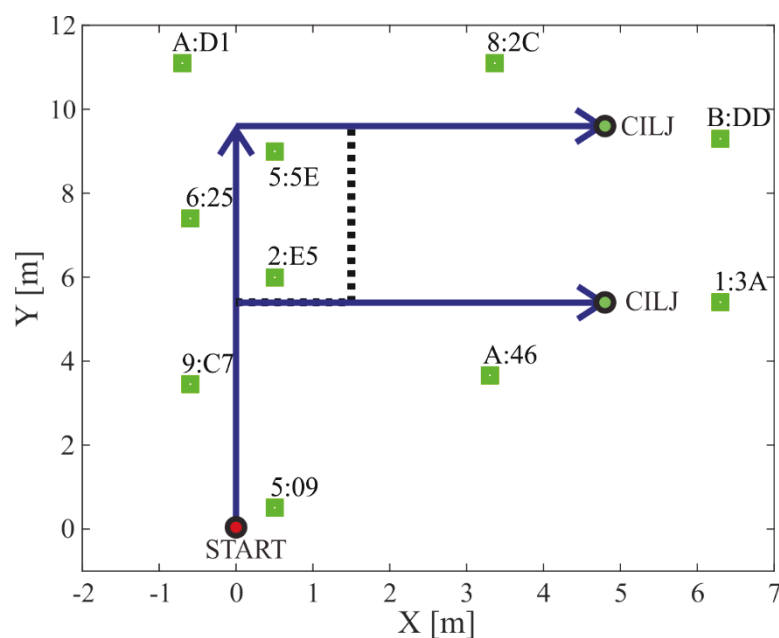
V laboratoriju velikosti približno 85 m² smo za namen preizkusa algoritmov lokalizacije, ki temeljijo na merjenju jakosti Bluetooth signalov, namestili 10 oddajnikov podjetja Kontakt.io. Oddajnike smo razporedili po prostoru, kot je zapisano v tabeli 6.3 in prikazano na sliki 6.9. Da bi prisotnost človeškega telesa v prostoru čim manj vplivala na meritve jakosti signalov, smo oddajnike namestili na višino 2 m od tal.

Vsi Bluetooth oddajniki v prostoru so bili nastavljeni na največjo moč oddajanja signalov, tj. 4 dBm. Pri tej nastavljeni moči naj bi bila jakost Bluetooth signalov na oddaljenosti 1 m od oddajnika (to določi proizvajalec s kalibracijo) enaka $TXPOWER = -59$ dBm [194], doseg Bluetooth signalov pa naj bi bil 70 m v odprtem prostoru (brez ovir) [195]. Pri največji moči oddajanja signalov naj bi ena baterija (3 V CR2477 s kapaciteto 1000 mAh) oddajnikov zadostovala za 15 mesecev delovanja [195]. Interval oddajanja signalov je bil nastavljen na 100 ms.

Tabela 6.3: Položaj Bluetooth oddajnikov v prostoru

Oznaka oddajnika	MAC Bluetooth oddajnika	Položaj
iBeacon 1	E3:10:72:E5:75:09	(0,50 0,50)
iBeacon 2	C4:A6:FC:77:89:C7	(-0,60 3,45)
iBeacon 3	F3:F4:4B:70:65:5E	(0,50 9,00)
iBeacon 4	DD:C2:72:C2:11:3A	(6,30 5,40)
iBeacon 5	E4:28:E3:81:A8:2C	(3,37 11,10)
iBeacon 6	D9:50:3B:F6:AA:46	(3,30 3,65)
iBeacon 7	DD:57:4F:2E:3B:DD	(6,30 9,30)
iBeacon 8	C6:63:3D:43:02:E5	(0,50 6,00)
iBeacon 9	C5:FA:6D:A0:46:25	(-0,60 7,40)
iBeacon 10	DA:57:30:EC:6A:D1	(-0,70 11,10)

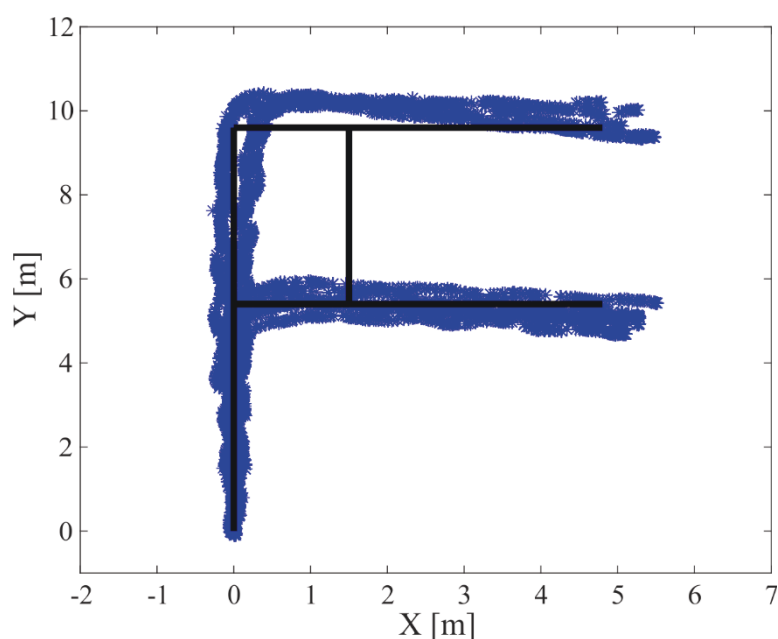
Oddajnike smo poskušali razporediti čim bolj enakomerno po celotnem testnem poligonu (slika 6.9), pri čemer so bile prisotne določene omejitve zaradi postavitve sten, vrat, stebrov in oken.



Slika 6.9: Položaj Bluetooth oddajnikov v prostoru

Meritve jakosti signalov iz vseh desetih oddajnikov smo pridobili tako, da smo desetkrat prehodili poti, ki sta prikazani na sliki 6.9 (od točke START do točke CILJ)

in pri tem s pametnim telefonom zajemali jakosti signalov. Pri tem smo sledili trenutni položaj pametnega telefona z uporabo algoritma, ki preko razširjenega Kalmanovega filtra združuje vizualno odometrijo in inercialni navigacijski sistem [196]. Vsi položaji, ki so bili določeni na ta način in v katerih so bile opravljene meritve jakosti signalov, so prikazani na sliki 6.10. Na tej sliki lahko opazimo, da točke nekoliko odstopajo od načrtane poti (črte so bile označene na tleh), in sicer za okoli 0,5 m. Del teh napak lahko pripišemo lokalizacijskemu algoritmu [196], ki določa položaj relativno glede na začetno točko, del pa tudi nenatančni hoji pešca po črti.



Slika 6.10: Vsi položaji, v katerih so bile zajete meritve jakosti signalov

6.2.2 Prileganje modela na meritve jakosti signalov

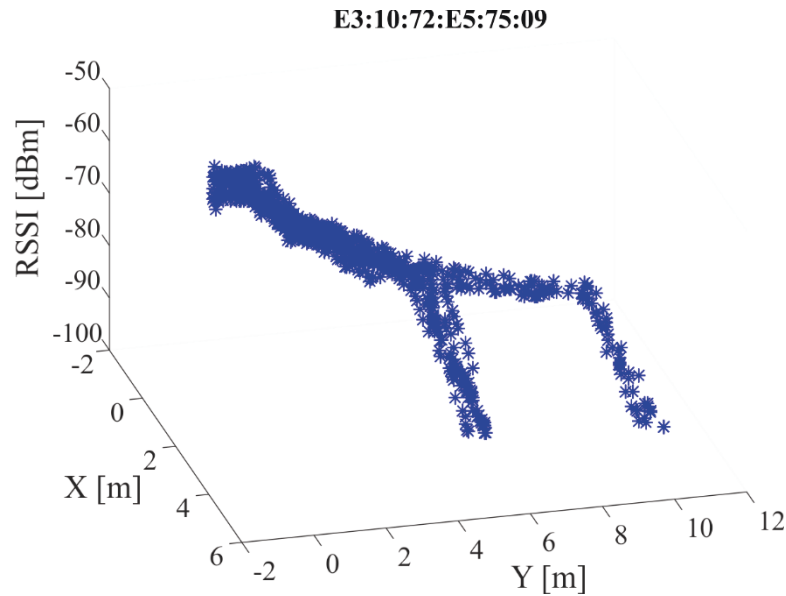
Vse meritve jakosti signalov smo razvrstili glede na MAC naslove oddajnikov in tako dobili deset množic. Na sliki 6.11, kjer so prikazane meritve jakosti signalov za oddajnik z MAC naslovom E3:10:72:E5:75:09, lahko opazimo, kako jakost signalov pada z oddaljevanjem od oddajnika s položajem (0,5; 0,5).

Padanje jakosti signalov z razdaljo (od oddajnikov) lahko opišemo z matematičnim modelom, ki ima obliko splošne nelinearne enačbe in ima za razdalje $d \geq 1$ m tri parametre:

$$\begin{aligned} \text{if } \text{ratio} < 1 : d &= \text{ratio}^{10} \\ \text{else} : d &= K_1 \cdot \text{ratio}^{K_2} + K_3, \end{aligned} \quad (6.1)$$

kjer je:

$$\text{ratio} = \frac{RSSI}{TXPOWER} = \frac{RSSI}{-59}.$$



Slika 6.11: Meritve jakosti signalov (iz oddajnika z MAC naslovom E3:10:72:E5:75:09) po poteh, ki so označene na sliki 6.9

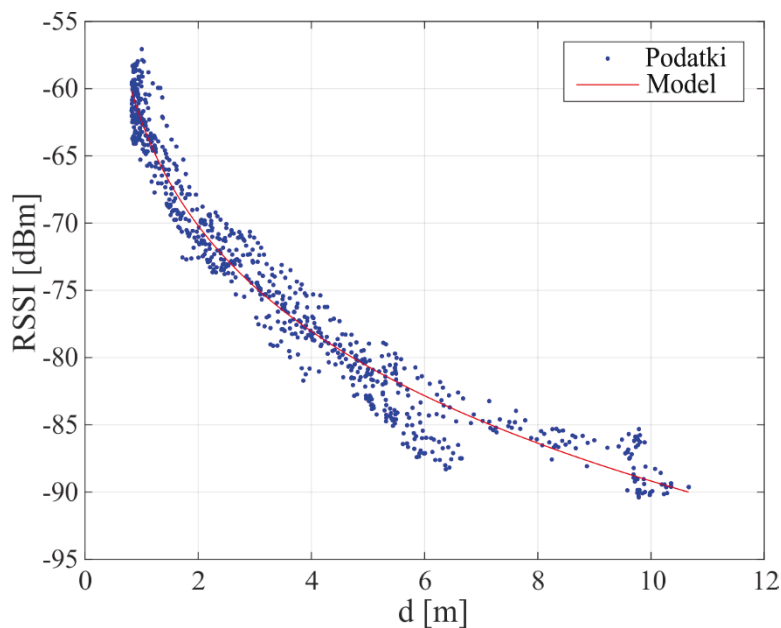
$TXPOWER$ predstavlja jakost signala pri oddaljenosti $d = 1$ m od oddajnika. Če upoštevamo, da je $TXPOWER = -59$ dBm (pri nastavljeni moči oddajanja 4 dBm) lahko zapišemo:

$$RSSI = f_{RSSI}(d) = \begin{cases} -59 \cdot \left(\frac{d-K_3}{K_1}\right)^{1/K_2}, & d \geq 1 \\ -59 \cdot d^{1/10}, & d < 1 \end{cases} \quad (6.2)$$

Na podlagi meritev jakosti signalov iz vseh 10 oddajnikov, razporejenih po prostoru, smo (v okolju Matlab) zgradili 10 matematičnih modelov, ki opisujejo padanje jakosti signalov z oddaljenostjo od oddajnika (6.2). Na sliki 6.12 je prikazano prileganje modela na meritve jakosti signalov iz oddajnika z MAC naslovom E3:10:72:E5:75:09. Kot lahko opazimo na sliki 6.12, meritve jakosti signalov vsebujejo šum, ki je posledica nihanja jakosti signalov (zaradi odbojev in absorpcij) in tudi ne povsem točnega določanja položaja sprejemnika pri zajemu meritev. Iz funkcije jakosti signalov (6.2) lahko opazimo, da so parametri K_1 , K_2 in K_3 prisotni le

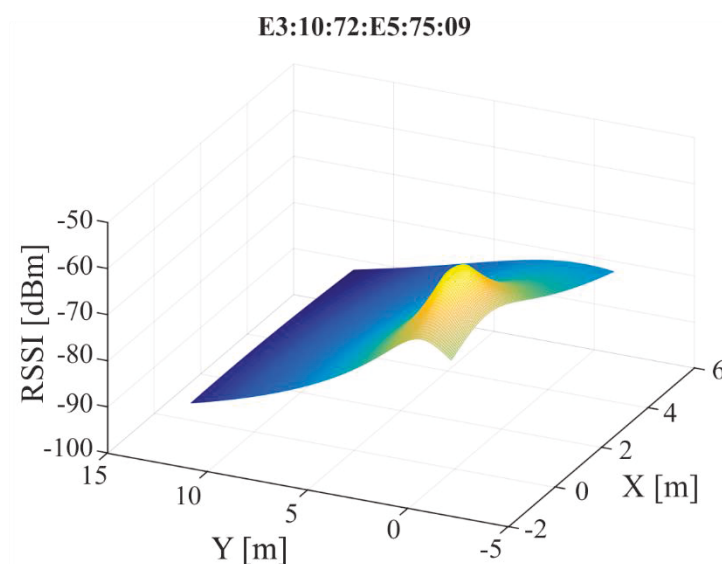
v primeru, ko je razdalja od oddajnika večja ali enaka 1 m. Zato smo pri iskanju teh parametrov upoštevali le meritve, ki so bile pridobljene na razdalji vsaj 1 m od oddajnika. Prileganje modela (6.2) smo izvedli z omejeno nelinearno optimizacijo (uporabljena je bila metoda omejenega koraka, ki je opisana v poglavju 6.3.2), s katero smo dobili parametre K_1 , K_2 in K_3 . Kot kriterij pri optimizaciji je bila uporabljena metoda najmanjših absolutnih residuov, pri kateri se išče minimum vsote $S = \sum_{i=1}^m |r_i|$. Residui so določeni kot $r_i = y_i - f_{RSSI}(d_i)$ (za $i = 1, 2, \dots, m$), kjer je y_i meritev jakosti signala na oddaljenosti d_i , m pa je število vseh meritev jakosti signalov iz enega oddajnika. Za doseg boljših rezultatov pri iskanju parametrov K_1 , K_2 in K_3 smo uporabili omejeno nelinearno optimizacijo, kar pomeni, da smo za vsak parameter K predpisali interval, na katerem naj bi se nahajala rešitev: $0 < K_1 \leq 8$, $1 \leq K_2 \leq 8$, $-9 \leq K_3 \leq 1$. Za začetne vrednosti parametrov smo izbrali vrednosti, ki jih uporablja proizvajalec oddajnikov v modelu za opis padanja jakosti signala z razdaljo: $K_1 = 0,90$; $K_2 = 7,70$; $K_3 = 0,11$.

Na sliki 6.12 vidimo, da se model dobro prilega meritvam jakosti signalov za oddajnik z MAC naslovom E3:10:72:E5:75:09. Meritve za ta oddajnik vsebujejo zelo malo šuma, saj je bilo večina meritev (do razdalje $d \approx 9$ m) zajetih tako, da ni bilo nobene ovire (stene) med oddajnikom in sprejemnikom.



Slika 6.12: Prileganje matematičnega modela (6.2) na meritve jakosti Bluetooth signalov oddajnika z MAC naslovom E3:10:72:E5:75:09

Z uporabo dobljenega modela za oddajnik z MAC naslovom E3:10:72:E5:75:09 smo generirali 3D prikaz širjenja signala po prostoru, ki ga lahko vidimo na sliki 6.13. Opaziti je mogoče, da signal na razdalji 10 m pade na približno -90 dBm (kar pomeni, da signal pada približno 3 dBm/m). Spreminjanje jakosti signala (po prostoru), kot je prikazano na sliki 6.13, je idealno, saj vrednosti padajo v vseh smereh enako in monotonno z razdaljo od oddajnika, kar pa v realnosti ne drži, če so v prostoru prisotne ovire (stene, pregrade). Bolj idealen model bi dobili s prileganjem ploskve na prostorsko razporejene meritve jakosti signalov. Vendar bi v tem primeru potrebovali več meritev jakosti signalov, ki bi bile zajete bolj enakomerno po prostoru okoli oddajnika (kar je v praksi zelo težavno ali celo nemogoče).

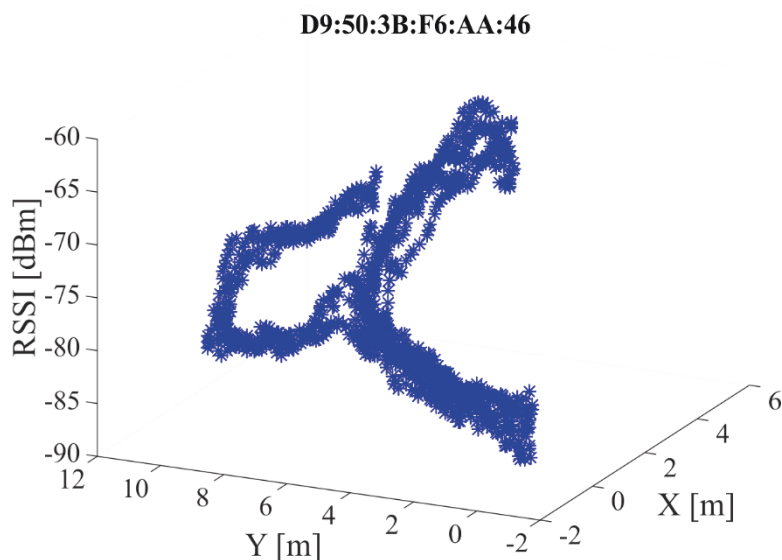


Slika 6.13: Širjenje Bluetooth signala (oddajnika z MAC naslovom E3:10:72:E5:75:09) po prostoru

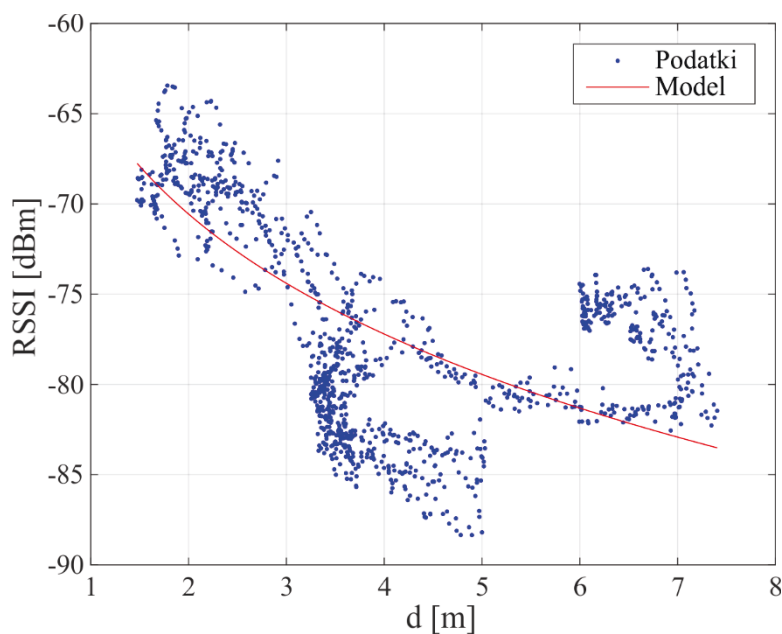
Na sliki 6.14 so prikazane meritve jakosti signalov za oddajnik z MAC naslovom D9:50:3B:F6:AA:46 in središčem v točki (3,30; 3,65). Pri zajemu meritev jakosti signalov za ta oddajnik po poteh, prikazanih na sliki 6.10, je bilo prisotnih veliko ovir (stene, stebri, lesene pregrade) med oddajnikom in sprejemnikom. Posledično lahko vidimo na sliki 6.15 veliko razpršenost jakosti signalov pri enaki oddaljenosti od oddajnika.

Na sliki 6.15 vidimo, da se model (6.2) v tem primeru ne more prilegati vsem meritvam jakosti signalov. To pomeni, da so jakosti signalov v enem delu prostora

dobro opisane z modelom, v drugem delu (npr. za steno) pa zelo slabo. Cilj je doseči kompromis v smislu, da bo model pri neki oddaljenosti od oddajnika približno enako zanesljiv za vse dele prostora (po celotni krožnici) ne glede na ovire.

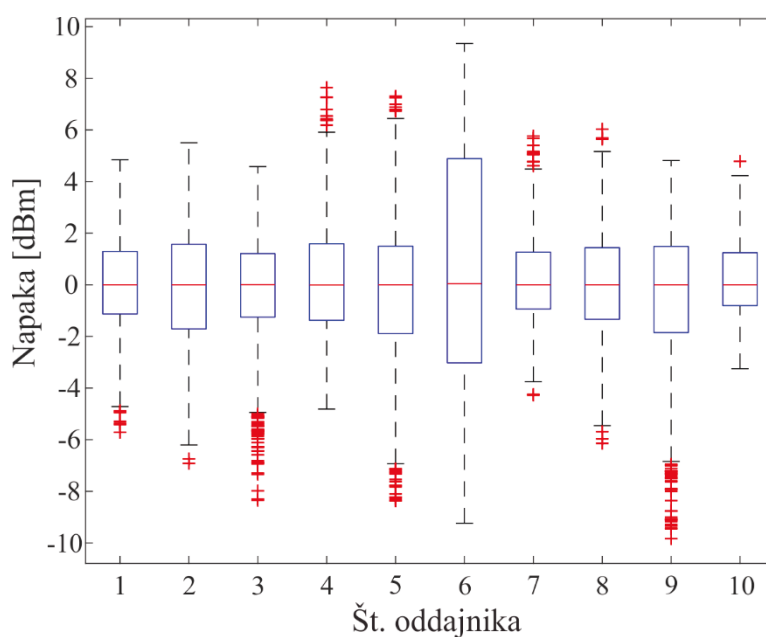


Slika 6.14: Meritve jakosti signalov (iz oddajnika z MAC naslovom D9:50:3B:F6:AA:46) po poteh, ki so označene na sliki 6.9



Slika 6.15: Prileganje matematičnega modela (6.2) na meritve jakosti Bluetooth signalov oddajnika z MAC naslovom D9:50:3B:F6:AA:46

Na sliki 6.16 je prikazan graf »škatla z brki« (ang. box plot), ki prikazuje razpon napak za vse modele jakosti signalov (katerih parametri so bili določeni z uporabo omejene nelinearne optimizacije), ki jih opisuje enačba (6.2). Modri okvirji na sliki 6.16 zajemajo 50 % vseh napak pri posameznem modelu. Z rdečimi križci pa so označene zelo odstopajoče napake (ang. outliers), ki so večje od $Q3 + 1,5 * IQR$ ali manjše od $Q1 - 1,5 * IQR$ (med ti dve meji se uvrsti 99,3 % vseh napak), kjer $Q1$ označuje prvi kvartil (25. percentila), $Q3$ (75. percentila) tretji kvartil. Vrednost IQR (ang. interquartile range) je določena kot: $IQR = Q3 - Q1$.



Slika 6.16: Razpon napak pri posameznih modelih padanja jakosti signalov

Glede na razpore napak (za vse modele jakosti signalov), ki so prikazani na sliki 6.16, lahko ugotovimo, da vsi modeli, z izjemo modela, ki pripada oddajniku št. 6 (z MAC naslovom D9:50:3B:F6:AA:46), približno enako dobro opisujejo meritve jakosti signalov iz posameznih oddajnikov. Največji razpon napak dobimo pri modelu, ki pripada oddajniku št. 6: $[-9,23 \ 9,35]$ dBm.

6.2.3 Mehki identifikacijski modeli

Z namenom, da bi dosegli boljše prilaganje modelov na izmerjene meritve (pri vseh oddaljenostih od posameznega oddajnika), smo izbrali bolj sofisticiran pristop k modeliranju, in sicer algoritem »SUHICLUST« (ang. SUpervised, HIERarchical

CLUSTtering) [197, 198], ki omogoča identifikacijo nelinearnih sistemov (opis sistema z mehkiimi identifikacijskimi modeli). V tem primeru je celoten model sestavljen iz več lokalnih linearnih podmodelov. Algoritem SUHICLUST združuje prednosti nadzorovanega učenja (ang. supervised) s tem, da vključuje hierarhični algoritem (kjer upošteva napake modelov pri nadaljnji delitvi podatkov), ki temelji na hevristični gradnji dreves in prednosti mehkega rojenja (nenadzorovano učenje). Prispevke posameznih linearnih modelov določajo uteži (pripadnostne funkcije), pridobljene pri postopku mehkega rojenja. Pri algoritmu je potrebno izbrati maksimalno število lokalnih modelov (to je enako številu rojev) ali maksimalno dovoljeno globalno napako, ki posledično določata maksimalno število iteracij. Pri nelinearnem modeliranju je ključna naloga delitev problemske domene (npr. inkrementalno ortogonalna delitev vhodnega prostora na hiperkvadre, aksialno neortogonalna delitev). Vhodni prostor je tako potrebno razdeliti na pripadnostne funkcije, ki hkrati določajo tudi prispevke posameznih lokalnih modelov. Parametre lokalnih linearnih modelov je mogoče določiti z lokalno ali globalno metodo najmanjših kvadratov. Kompleksnost modela inkrementalno narašča in algoritem se zaključi v primeru, ko je napaka dovolj majhna oz. je že bilo ustvarjenih maksimalno število lokalnih linearnih modelov.

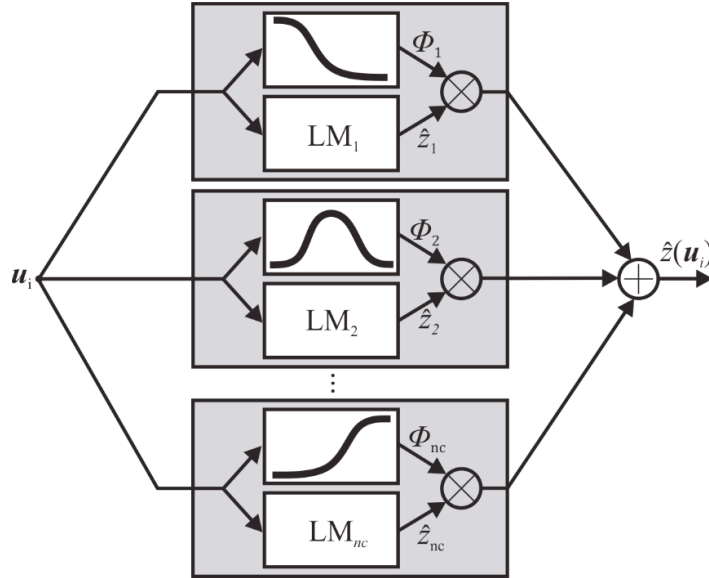
Pri algoritmu SUHICLUST je struktura modela enaka obliki Takagi-Sugeno (T-S) mehkega modela. Takagi-Sugeno (T-S) mehki modeli se uporabljajo kot nelinearni aproksimator za nelinearne statične funkcije ali nelinearne dinamične aproksimacije [199, 200, 201]. Za pripadnostne funkcije so bile izbrane normalizirane Gaussove funkcije.

Izhod $\hat{z}(\mathbf{u}_i)$ (T-S) mehkega modela z nu vhodi $\mathbf{u}_i$ je določen kot interpolacija nc izhodov lokalnih modelov (LM) $\hat{z}_k(\mathbf{u}_i) = \hat{\boldsymbol{\theta}}_k^T \mathbf{u}_i + \hat{\theta}_{k0}$ (slika 6.17), kjer gre $k = 1, \dots, nc$ in $i = 1, \dots, N$ ($\hat{\boldsymbol{\theta}}_k = [\hat{\theta}_{k,1}, \dots, \hat{\theta}_{k,nu}]^T$ so koeficienti linearnega lokalnega modela in N je število vseh meritev vhodov in izhodov):

$$\hat{z}(\mathbf{u}_i) = \sum_{k=1}^{nc} \hat{z}_k(\mathbf{u}_i) \Phi_k(\mathbf{u}_i), \quad (6.3)$$

kjer je $\mathbf{u}_i = [u_{i,1}, \dots, u_{i,nu}]^T$.

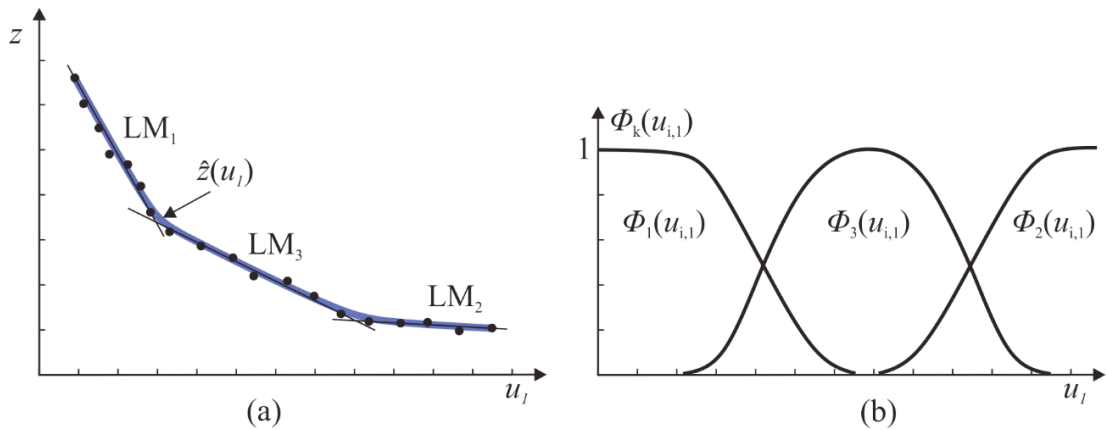
S Φ_k so označene normalizirane pripadnostne funkcije (ali tudi utežne funkcije), ki določajo območja in prispevke linearnih lokalnih modelov h končnemu globalnemu modelu oz. njegovemu izhodu.



Slika 6.17: Mehki model sestavljen iz več lokalnih linearnih modelov

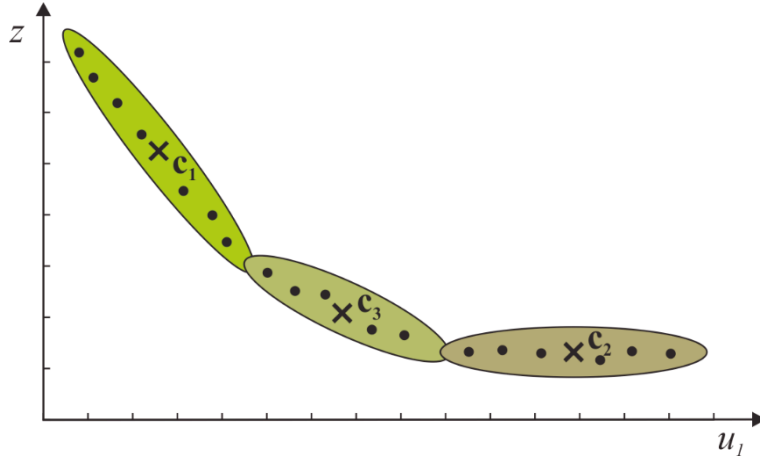
Utežne funkcije, ki določajo prispevke lokalnih modelov (slika 6.18 (a)), lahko zavzamejo vrednosti med 0 in 1 (slika 6.18 (b)). Za celoten vhodni prostor velja, da je vsota prispevkov vseh lokalnih modelov enaka ena:

$$\sum_{k=1}^{nc} \Phi_k(\mathbf{u}_i) = 1, \Phi_k(\mathbf{u}_i) > 0. \quad (6.4)$$



Slika 6.18: Primer mehkega modela (a), ki je sestavljen iz treh linearnih modelov. Njihov prispevek določajo utežne funkcije Φ_k (b).

Algoritem SUHICLUST pri postopku mehkega rojenja uporablja algoritem Gustafson–Kessel (GK) [202]. Posamezen roj (slika 6.19) je definiran s položajem središča roja $\mathbf{c}_k = [c_{k,1}, c_{k,2}, \dots, c_{k,nu}, c_{k,z}]^T$ in mehko kovariančno matriko $\mathbf{P}_k$.



Slika 6.19: Primer rojenja s tremi roji

Mehka kovariančna matrika k -tega mehkega roja $\mathbf{P}_k \in \mathbb{R}^{(nu+1) \times (nu+1)}$ je definirana kot:

$$\mathbf{P}_k = \sum_{i=1}^N \Phi_k^2(\mathbf{d}_i) (\mathbf{d}_i - \mathbf{c}_k)(\mathbf{d}_i - \mathbf{c}_k)^T, \quad (6.5)$$

kjer je $\mathbf{d}_i = [\mathbf{u}_i^T z_i]^T$ ($i = 1, \dots, N$) podatkovni vektor dolžine $(nu + 1)$ in Φ_k normalizirana pripadnost podatkovnega vektorja $\mathbf{d}_i$ h k -temu roju. Mehka kovariančna matrika določa smer in variabilnost podatkov v I/O (vhodno-izhodnem) prostoru. Z uporabo singularne dekompozicije - SVD (ang. Singular Value Decomposition) je mogoče mehko kovariančno matriko razstaviti kot:

$$\mathbf{P}_k = \mathbf{G}_k \mathbf{\Lambda}_k \mathbf{G}_k^T, \quad (6.6)$$

kjer je $\mathbf{G}_k \in \mathbb{R}^{(nu+1) \times (nu+1)}$ matrika lastnih vektorjev $\mathbf{G}_k = [\mathbf{g}_{k,1}, \mathbf{g}_{k,2}, \dots, \mathbf{g}_{k,nu+1}]$ in $\mathbf{\Lambda}_k \in \mathbb{R}^{(nu+1) \times (nu+1)}$ matrika lastnih vrednosti $\mathbf{\Lambda}_k = \text{diag}(\lambda_{k,l})$ ($l = 1, \dots, nu + 1$). Pri kreiranju rojev je potrebno določiti maksimalno število rojev in izbrati njihova središča tako, da so najmanjše normalizirane lastne vrednosti vseh rojev manjše od določenega praga $\frac{\lambda_{k,nu+1}}{\text{sled}(\mathbf{\Lambda}_k)} < \Delta$ (npr. $\Delta = 0,05$). Pri vsakem nadaljnjem rojenju so upoštevani podatki, ki pripadajo najslabšemu lokalnemu modelu.

Denimo, da imamo podatkovno množico predstavljeno v obliki matrike $\tilde{\mathbf{D}}$ velikosti $(N \times nu + 1)$:

$$\tilde{\mathbf{D}} = \begin{bmatrix} \tilde{u}_{1,1} & \cdots & \tilde{u}_{1,nu} & \tilde{z}_1 \\ \tilde{u}_{2,1} & \cdots & \tilde{u}_{2,nu} & \tilde{z}_2 \\ \vdots & \ddots & \vdots & \vdots \\ \tilde{u}_{N,1} & \cdots & \tilde{u}_{N,nu} & \tilde{z}_N \end{bmatrix}. \quad (6.7)$$

Podatkovno množico lahko zapišemo tudi kot: $\tilde{\mathbf{d}}_i = [\tilde{\mathbf{u}}_i^T \tilde{z}_i]^T$ ($i = 1, \dots, N$), kjer so $\tilde{\mathbf{u}}_i = [\tilde{u}_{i,1}, \dots, \tilde{u}_{i,nu}]^T$ vhodne meritve in $\tilde{z}_i$ meritev izhoda. Matriko s podatki $\tilde{\mathbf{D}}$ nato preoblikujemo (podatke centriramo in normaliziramo zaradi različnih velikostnih razredov magnitud vhodnih in izhodnih meritev) v normalizirano podatkovno matriko $\mathbf{D}$:

$$\mathbf{D} = [\mathbf{U}, \mathbf{z}] = \begin{bmatrix} u_{1,1} & \cdots & u_{1,nu} & z_1 \\ u_{2,1} & \cdots & u_{2,nu} & z_2 \\ \vdots & \ddots & \vdots & \vdots \\ u_{N,1} & \cdots & u_{N,nu} & z_N \end{bmatrix}, \quad (6.8)$$

kjer je:

$$u_{i,j} = \frac{\tilde{u}_{i,j} - m_{u_j}}{\tilde{\rho}_{u_j}}, \quad i = 1, \dots, N \text{ in } j = 1, \dots, nu, \quad (6.9)$$

$$z_i = \frac{\tilde{z}_i - m_z}{\tilde{\rho}_z} \quad (6.10)$$

in

$$m_{u_j} = \frac{1}{N} \sum_{i=1}^N \tilde{u}_{i,j}, \quad (6.11)$$

$$m_z = \frac{1}{N} \sum_{i=1}^N \tilde{z}_i \quad (6.12)$$

ter

$$\tilde{\rho}_{u_j} = \max_i (|\tilde{u}_{i,j} - m_{u_j}|), \quad (6.13)$$

$$\tilde{\rho}_z = \max_i (|\tilde{z}_i - m_z|). \quad (6.14)$$

Pri algoritmu SUHICLUST v prvem koraku izračunamo kovariančno matriko $\mathbf{P}_0$ podatkov zapisanih v matriki $\mathbf{D}$ kot:

$$\mathbf{P}_0 = \frac{1}{N-1} \mathbf{D}^T \mathbf{D}. \quad (6.15)$$

Lastne vrednosti kovariančne matrike $\mathbf{P}_0$ predstavljajo variance podatkovne matrike $\mathbf{D}$ v smeri pripadajočih lastnih vektorjev. Podatki se najbolj raztezajo v smeri lastnega vektorja, ki pripada največji lastni vrednosti. Ta določa varianco σ_0^2 okoli središča podatkov v smeri glavnega lastnega vektorja $\mathbf{g}_0$. Začetna centra rojev $\mathbf{v}_{11}$ in $\mathbf{v}_{12}$ za podatkovno množico $\mathbf{D}$ sta določena kot:

$$\mathbf{v}_{11} = -\boldsymbol{\epsilon}_0, \quad (6.16)$$

$$\mathbf{v}_{12} = \boldsymbol{\epsilon}_0, \quad (6.17)$$

$$\boldsymbol{\epsilon}_0 = \sigma_0 \mathbf{g}_0, \quad (6.18)$$

kjer je $\mathbf{g}_0$ glavni lastni vektor in σ_0 pripadajoča standardna deviacija. Začetna centra rojev $\mathbf{v}_{11}$ in $\mathbf{v}_{12}$ ter podatkovna matrika $\mathbf{D}$ so vhodi v GK algoritem za mehko rojenje. Ta vrne dva nova roja s pripadajočima centroma $\mathbf{c}_1$ in $\mathbf{c}_2$ ter mehkim kovariančnima matrikama $\mathbf{P}_{11}$ in $\mathbf{P}_{12}$.

V naslednjem koraku se izračunajo parametri lokalnih linearnih modelov z metodo najmanjših kvadratov. Razdaljo $l_{i,k}$ med i -to podatkovno točko in centrom k -tega roja $\mathbf{c}_k = [c_{k,1}, c_{k,2}, \dots, c_{k,nu}, c_{k,z}]^T$ je mogoče določiti z upoštevanjem mehke kovariančne matrike $\mathbf{P}_k$, ki skalira in rotira osi:

$$l_{i,k}^2 = (\mathbf{d}_i - \mathbf{c}_k)^T \mathbf{P}_k^{-1} (\mathbf{d}_i - \mathbf{c}_k), \quad (6.19)$$

kjer je

$$\mathbf{d}_i = [\mathbf{u}_i^T \ z_i]^T. \quad (6.20)$$

Mehka kovariančna matrika $\mathbf{P}_k$ je simetrična matrika z dimenzijami $(nu + 1) \times (nu + 1)$:

$$\mathbf{P}_k = \begin{bmatrix} \sigma_{1,1}^2 & \sigma_{1,2}^2 & \cdots & \sigma_{1,nu+1}^2 \\ \sigma_{2,1}^2 & \sigma_{2,2}^2 & \cdots & \sigma_{2,nu+1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{nu+1,1}^2 & \sigma_{nu+1,2}^2 & \cdots & \sigma_{nu+1,nu+1}^2 \end{bmatrix}. \quad (6.21)$$

Inverz matrike $\mathbf{P}_k$ je mogoče izračunati z uporabo Moore-Penrose psevdo-inverza, ki je primeren tudi, če je matrika $\mathbf{P}_k$ slabo pogojena (singularna).

Pripadnostne funkcije μ_k imajo obliko Gaussovih funkcij in jih lahko zapišemo kot:

$$\mu_k(\mathbf{d}_i) = e^{-\gamma l_{i,k}^2}, \quad (6.22)$$

kjer je $\gamma \geq 1$ faktor, ki določa gladkost Gaussovih funkcij $\mu_k(\mathbf{d}_i) > 0$ in posledično tudi gladkost normaliziranih pripadnostnih funkcij $\Phi_k(\mathbf{d}_i)$ (normalizirane Gaussove funkcije):

$$\Phi_k(\mathbf{d}_i) = \frac{\mu_k(\mathbf{d}_i)}{\sum_{p=1}^{nc} \mu_p(\mathbf{d}_i)}. \quad (6.23)$$

Pripadnostne funkcije je potrebno normalizirati zato, da je vsota prispevkov posameznih modelov (pri določenem vходу) k izhodu mehkega modela enaka ena.

V postopku rojenja se na osnovi podatkovne množice $\mathbf{D}$ kreira nc novih podatkovnih množic $\mathbf{D}_k$, ki pripadajo posameznim rojem. Množica $\mathbf{D}_k$ je definirana kot: $\mathbf{D}_k = \{\mathbf{d}_{k,i}\} (i = 1, \dots, N_k)$, pri čemer mora za vse podatkovne vektorje $\mathbf{d}_{k,i}$ veljati:

$$\Phi_k(\mathbf{d}_{k,i}) > 0. \quad (6.24)$$

Parametri k -tega lokalnega linearnega modela morajo biti določeni tako (z uporabo metode uteženih najmanjših kvadratov), da se ta optimalno prilega izhodni spremenljivki $z_{k,i} = z_i \Phi_k(\mathbf{d}_{k,i})$ oz. da je kriterijska funkcija (6.25) minimalna.

$$J_k = \sum_{i=1}^{N_k} (z_{k,i} - (\hat{\boldsymbol{\theta}}_k^T \mathbf{u}_{k,i} + \hat{\theta}_{k0}) \Phi_k(\mathbf{d}_{k,i}))^2 \quad (k = 1, \dots, nc), \quad (6.25)$$

kjer so $\boldsymbol{\theta}_k$ in θ_{k0} parametri lokalnega linearnega modela, ki opisuje podatke k -tega roja $\mathbf{D}_k$. Torej je aproksimacija izhodne spremenljivke $\hat{z}_{k,i}$ k -tega roja enaka:

$$\hat{z}_{k,i} = \hat{\boldsymbol{\theta}}_k^T \mathbf{u}_{k,i} + \hat{\theta}_{k0}, \quad k = 1, \dots, nc, \quad i = 1, \dots, N_k. \quad (6.26)$$

Kvaliteta lokalnega modela, ki opisuje podatke iz k -tega roja, je ocenjena kot relativna standardna deviacija σ_{q_k} :

$$\sigma_{q_k}^2 = \frac{1}{N_k - 1} \sum_{i=1}^{N_k} \frac{\delta_{k,i}^2}{\sigma_{z_k}^2}, \quad (6.27)$$

kjer je

$$\delta_{k,i} = z_{k,i} - (\hat{\boldsymbol{\theta}}_k^T \mathbf{u}_{k,i} + \hat{\theta}_{k0}) \Phi_k(\mathbf{d}_{k,i}) \quad (6.28)$$

in

$$\sigma_{z_k}^2 = \frac{1}{N_k - 1} \sum_{i=1}^{N_k} (z_{k,i} - m_{z_k})^2, \quad (6.29)$$

$$m_{z_k} = \frac{1}{N_k - 1} \sum_{i=1}^{N_k} z_{k,i}. \quad (6.30)$$

Roj z največjo relativno standardno deviacijo σ_{q_k} (kateremu pripada lokalni model z največjo napako) se v naslednji iteraciji razdeli na dva nova roja, za katera je potrebno izračunati parametre dveh novih linearnih modelov. Začetna centra dveh novih rojev sta določena kot:

$$\mathbf{v}_{k1} = \mathbf{c}_{k-1} + \boldsymbol{\epsilon}_k, \quad (6.31)$$

$$\mathbf{v}_{k2} = \mathbf{c}_{k-1} - \boldsymbol{\epsilon}_k, \quad (6.32)$$

kjer je:

$$\boldsymbol{\epsilon}_k = \sigma_k \mathbf{g}_{k,1}. \quad (6.33)$$

Središča rojev $\mathbf{c}_k$ in mehke kovariančne matrike $\mathbf{P}_k$ je potrebno transformirati nazaj v obliko podatkovne množice $\tilde{\mathbf{D}}$.

Nova središča mehkih rojev določimo kot:

$$\tilde{\mathbf{c}}_k = \mathbf{c}_k + \mathbf{c}_0 \quad (k = 1, \dots, nc) \quad (6.34)$$

in transformirane kovariančne matrike $\tilde{\mathbf{P}}_k$ kot:

$$\tilde{\mathbf{P}}_k = \mathbf{P}_k \cdot \mathbf{T}, \quad (6.35)$$

$$\mathbf{T} = \begin{bmatrix} m_{u_1}^2 & m_{u_1}m_{u_2} & \cdots & m_{u_1}m_{u_{nu}} \\ m_{u_2}m_{u_1} & m_{u_2}^2 & \cdots & m_{u_2}m_{u_{nu}} \\ \vdots & \vdots & \ddots & \vdots \\ m_{u_{nu}}m_{u_1} & m_{u_{nu}}m_{u_2} & \cdots & m_{u_{nu}}^2 \end{bmatrix}, \quad (6.36)$$

kjer je $m_{u_j} = \frac{1}{N} \sum_{i=1}^N \tilde{u}_{i,j}$.

Da bi lahko izračunali izhod mehkega modela, ki je sestavljen iz več lokalnih modelov, je potrebno najprej izračunati *normalizirane* pripadnostne funkcije $\Phi_k(\tilde{\mathbf{u}}_i)$:

$$\Phi_k(\tilde{\mathbf{u}}_i) = \frac{\mu_k(\tilde{\mathbf{u}}_i)}{\sum_{p=1}^{nc} \mu_p(\tilde{\mathbf{u}}_i)}, \quad (6.37)$$

kjer so $\mu_k(\tilde{\mathbf{u}}_i)$ pripadnostne funkcije:

$$\mu_k(\tilde{\mathbf{u}}_i) = e^{-\gamma L_{k,i}^2}. \quad (6.38)$$

$L_{k,i}^2$ so razdalje med podatkovnimi točkami in središči rojev:

$$L_{k,i}^2 = ([\tilde{\mathbf{u}}_i^T \quad \tilde{c}_{kz}]^T - \tilde{\mathbf{c}}_k)^T \tilde{\mathbf{P}}_k^{-1} ([\tilde{\mathbf{u}}_i^T \quad \tilde{c}_{kz}]^T - \tilde{\mathbf{c}}_k), \quad (6.39)$$

kjer je $\tilde{\mathbf{P}}_k$ ($i = 1, \dots, nc$) denormalizirana mehka kovariančna matrika.

Na podlagi znanih parametrov lokalnih modelov $\hat{\boldsymbol{\theta}}_k$ in $\hat{\theta}_{k0}$ (ki jih tokrat izračunamo z uporabo globalne metode uteženih najmanjših kvadratov, pri čemer so upoštevane pripadnostne funkcije) in normaliziranih pripadnostnih funkcij $\Phi_k(\tilde{\mathbf{u}}_i)$ je mogoče izračunati izhod mehkega modela:

$$\hat{z}_i = \sum_{k=1}^{nc} (\hat{\boldsymbol{\theta}}_k^T \tilde{\mathbf{u}}_i + \hat{\theta}_{k0}) \Phi_k(\tilde{\mathbf{u}}_i) \text{ za } i = 1, \dots, N. \quad (6.40)$$

V zadnjem koraku algoritma SUHICLUST je potrebno narediti še kvantitativno validacijo modela. Za ta namen je uporabljena funkcija *NRMSE* (ang. normalized root-mean-square error):

$$NRMSE(\tilde{\mathbf{z}}, \hat{\tilde{\mathbf{z}}}) = \sqrt{\frac{\sum_{i=1}^N (\tilde{z}_i - \hat{\tilde{z}}_i)^2}{\sum_{p=1}^N (\tilde{z}_p - \bar{\tilde{z}})^2}}, \quad (6.41)$$

kjer je $\tilde{\mathbf{z}} = [\tilde{z}_1, \dots, \tilde{z}_N]^T$ merjeni izhod, $\hat{\tilde{\mathbf{z}}} = [\hat{\tilde{z}}_1, \dots, \hat{\tilde{z}}_N]^T$ izhod mehkega modela ter $\bar{\tilde{z}} = \frac{1}{N} \sum_{i=1}^N \tilde{z}_i$.

Če pri izračunani vrednosti $NRMSE(\tilde{\mathbf{z}}, \hat{\tilde{\mathbf{z}}})$ ni izpolnjen pogoj (6.42), je potrebno celoten postopek algoritma SUHICLUST ponoviti.

$$NRMSE(\tilde{\mathbf{z}}, \hat{\tilde{\mathbf{z}}}) < NRMSE_{MAX}. \quad (6.42)$$

V nadaljevanju je prikazana psevdokoda algoritma SUHICLUST [197, 198], ki je razdeljena na 15 ključnih korakov:

Psevdokoda algoritma SUHICLUST:

1. Inicializacija algoritma:
 - Transformacija podatkov $\tilde{\mathbf{D}} \rightarrow \mathbf{D}$ ali $\tilde{\mathbf{d}} \rightarrow \mathbf{d}$
 - Inicializacija parametrov algoritma:
 - o Faktor glajenja γ , ($\gamma = 1$)
 - o Maksimalno število rojev nc_{max}
 - o Toleranca aproksimacijske napake $NRMSE_{MAX}$,
($NRMSE_{MAX} = 0,05$)
2. Izračun kovariančne matrike $\mathbf{P}_0$ (6.15)
3. Določitev začetnih centrov rojev: $\mathbf{v}_{11} = -\sigma_0 \mathbf{g}_0$ in $\mathbf{v}_{12} = \sigma_0 \mathbf{g}_0$
4. Gustafson–Kessel mehko rojenje na celotni množici podatkov $\tilde{\mathbf{D}} \rightarrow$ centra rojev $\mathbf{c}_1$ in $\mathbf{c}_2$
- do** {
5. Izračun vrednosti $\Phi_k(\mathbf{d}_i)$ (6.23), $i = 1, \dots, N$, $k = 1, \dots, nc$
6. Izračun parametrov $\hat{\boldsymbol{\theta}}_k$ in $\hat{\theta}_{k0}$ z lokalno metodo najmanjših kvadratov
7. Izračun vrednosti σ_{qk} (6.27), $k = 1, \dots, nc$
8. Razdelitev roja z največjo lokalno napako in določitev središč dveh novih rojev

$$\mathbf{v}_{k1} = \mathbf{c}_{k-1} + \boldsymbol{\epsilon}_k$$

$$\mathbf{v}_{k2} = \mathbf{c}_{k-1} - \boldsymbol{\epsilon}_k$$

$$\boldsymbol{\epsilon}_k = \sigma_k \mathbf{g}_{k,1}$$
9. GK rojenje z uporabo elementov iz roja, ki ima največjo lokalno napako $\rightarrow$ centra rojev $\mathbf{c}_{k1}$ in $\mathbf{c}_{k2}$
10. Določitev novih začetnih centrov rojev:

$$(\mathbf{v}_1, \dots, \mathbf{v}_k, \dots, \mathbf{v}_{nc_{new}}) = (\mathbf{c}_1, \dots, \mathbf{c}_{k1}, \mathbf{c}_{k2}, \dots, \mathbf{c}_{nc}), nc_{new} = nc + 1$$
11. GK mehko rojenje z uporabo vseh podatkov $\rightarrow$ centri rojev $\mathbf{c}_k$,
 $k = 1, \dots, nc_{new}$
12. Transformacija središč in mehkih kovariančnih matrik glede na množico meritev

$$\mathbf{c}_k \rightarrow \tilde{\mathbf{c}}: \tilde{\mathbf{c}}_k = \mathbf{c}_k + \mathbf{c}_0, k = 1, \dots, nc$$

$$\mathbf{P}_k \rightarrow \tilde{\mathbf{P}}_k: \tilde{\mathbf{P}}_k = \mathbf{P}_k \cdot \mathbf{T}, k = 1, \dots, nc$$
13. Izračun vrednosti $\Phi_k(\tilde{\mathbf{u}}_i)$ (6.38)
14. Izračun parametrov $\hat{\boldsymbol{\theta}}_k$ in $\hat{\theta}_{k0}$ z uporabo globalne metode uteženih najmanjših kvadratov
15. $\hat{\mathbf{z}}_i = \sum_{k=1}^{nc} (\hat{\boldsymbol{\theta}}_k^T \tilde{\mathbf{u}}_i + \hat{\theta}_{k0}) \Phi_k(\tilde{\mathbf{u}}_i)$ za $i = 1, \dots, N$
- }** **while** { $NRMSE(\hat{\mathbf{z}}, \tilde{\mathbf{z}}) < NRMSE_{MAX}$ ali $nc = nc_{max}$ }

Prednosti gradnje mehkega modela z algoritmom SUHICLUST so naslednje:

- Algoritem deluje inkrementalno: v vsaki iteraciji je dodan nov linearni model.

- Delitev: v vsaki iteraciji je lokalni model z največjo napako razdeljen na dva nova podmodela.
- Lokalna uporaba metode najmanjših kvadratov: parametri lokalnih linearnih modelov so ocenjeni z uporabo metode uteženih najmanjših kvadratov.
- Prilagodljiva natančnost: Gladkost interpolacije linearnih modelov je odvisna od mehke kovariančne matrike, ki je dobljena z rojenjem in s tem tudi od velikosti področij, ki jih zavzamejo normalizirane pripadnostne funkcije (manjša kot so področja slabša bo gladkost interpolacije).

Določanje območja zaupanja pri mehki identifikaciji z metodo SUHICLUST

Za novo množico podatkov lahko definiramo interval zaupanja [203], s tem da obravnavamo enako funkcijo $g \in G$ kot pri identifikaciji modela. Dano imamo množico meritev izhodnih vrednosti $\mathbf{Z}^* = \{z_1^*, \dots, z_M^*\}$ in množico vhodov $\mathbf{U}^* = \{\mathbf{u}_1^*, \dots, \mathbf{u}_M^*\}$, pri čemer velja $z_i^* = g(\mathbf{u}_i^*)$, $\mathbf{u}_i^* \in \mathcal{S}$ ($i = 1, \dots, M$) pa je validacijska podatkovna množica.

Da bi lahko določili interval zaupanja pri mehki identifikaciji, je najprej potrebno poiskati spodnjo mehko funkcijo $\underline{f}$ in zgornjo mehko funkcijo $\bar{f}$ tako da velja:

$$\underline{f}(\mathbf{u}_i^*) \leq g(\mathbf{u}_i^*) \leq \bar{f}(\mathbf{u}_i^*), \forall \mathbf{u}_i^* \in \mathcal{S}. \quad (6.43)$$

Funkcija g se torej z določenim zaupanjem nahaja znotraj območja, ki ga definirata zgornja in spodnja mehka funkcija.

Merjene izhodne vrednosti k -tega lokalnega linearnega modela (število lokalnih modelov je nc) so določene kot:

$$\mathbf{z}_k^* = \Psi_k^{*T} \boldsymbol{\theta}_{v_k} + \mathbf{e}_k^* \quad (k = 1, \dots, nc), \quad (6.44)$$

kjer je $\Psi_k^* = [\boldsymbol{\psi}_{k,1}^*, \dots, \boldsymbol{\psi}_{k,M}^*]$ regresijska matrika k -tega lokalnega linearnega modela in $\mathbf{z}_k^* = [z_1^*, \dots, z_M^*]^T$ ter $\boldsymbol{\theta}_{v_k} = [\theta_{k0} \ \boldsymbol{\theta}_k^T]^T$. $\boldsymbol{\psi}_{k,i}^{*T}(\mathbf{u}_i^*) = \Phi_k(\mathbf{u}_i^*)[1 \ \mathbf{u}_i^{*T}]$ predstavlja regresijski vektor. Izhod k -tega lokalnega linearnega modela je pri upoštevanju validacijske podatkovne množice enak:

$$\hat{\mathbf{z}}_k^* = \Psi_k^{*T} \hat{\boldsymbol{\theta}}_{v_k}, \quad (6.45)$$

kjer so $\hat{\boldsymbol{\theta}}_{v_k} = [\hat{\theta}_{k0} \ \hat{\boldsymbol{\theta}}_k^T]^T$ parametri k -tega modela, določeni v fazi mehke identifikacije po metodi najmanjših kvadratov (z uporabo vektorja izhodnih meritev $\mathbf{z}_k = [z_{k,1}, \dots, z_{k,N_k}]^T$ ter regresijske matrike $\Psi_k = [\boldsymbol{\psi}_{k,1}, \dots, \boldsymbol{\psi}_{k,N_k}]$ za k -ti model).

Za določitev intervala zaupanja je potrebno izračunati pričakovano kovarianco residua med izhodom modela in novo množico podatkov v vsaki lokalni domeni:

$$\text{cov}(\mathbf{z}_k^* - \hat{\mathbf{z}}_k^*) = E\{(\mathbf{z}_k^* - \hat{\mathbf{z}}_k^* - E\{\mathbf{z}_k^* - \hat{\mathbf{z}}_k^*\})(\mathbf{z}_k^* - \hat{\mathbf{z}}_k^* - E\{\mathbf{z}_k^* - \hat{\mathbf{z}}_k^*\})^T\}. \quad (6.46)$$

Pri upoštevanju enakih statističnih lastnosti šuma za podatke iz validacijske množice ($E\{\mathbf{e}_k^*\} = 0$) in identifikacijske množice ($E\{\mathbf{e}_k\} = 0$) je pričakovana vrednost napake med merjenim izhodom in ocenjenim izhodom enaka $E\{\mathbf{z}_k^* - \hat{\mathbf{z}}_k^*\} = 0$.

Kovariančno matriko iz enačbe (6.46) lahko torej zapišemo kot:

$$\text{cov}(\mathbf{z}_k^* - \hat{\mathbf{z}}_k^*) = E\{(\mathbf{e}_k^* - \Psi_k^{*T} \tilde{\boldsymbol{\theta}}_{v_k})(\mathbf{e}_k^* - \Psi_k^{*T} \tilde{\boldsymbol{\theta}}_{v_k})^T\}, \quad (6.47)$$

kjer je $\tilde{\boldsymbol{\theta}}_{v_k} = [\tilde{\boldsymbol{\theta}}_{k0} \ \tilde{\boldsymbol{\theta}}_k^T]^T$ in nadalje kot:

$$\text{cov}(\mathbf{z}_k^* - \hat{\mathbf{z}}_k^*) = E\{\mathbf{e}_k^* \mathbf{e}_k^{*T}\} - E\{\Psi_k^{*T} \tilde{\boldsymbol{\theta}}_{v_k} \mathbf{e}_k^{*T}\} - E\{\mathbf{e}_k^* \tilde{\boldsymbol{\theta}}_{v_k}^T \Psi_k^*\} + E\{\Psi_k^{*T} \tilde{\boldsymbol{\theta}}_{v_k} \tilde{\boldsymbol{\theta}}_{v_k}^T \Psi_k^*\}. \quad (6.48)$$

Nato upoštevamo enačbo (za optimizacijo z vsoto najmanjših kvadratov):

$$\tilde{\boldsymbol{\theta}}_{v_k} = (\Psi_k \Psi_k^T)^{-1} \Psi_k \mathbf{e}_k, \quad (6.49)$$

kjer velja $\hat{\boldsymbol{\theta}}_{v_k} = \boldsymbol{\theta}_{v_k} + \tilde{\boldsymbol{\theta}}_{v_k}$. Predpostavimo, da imata oba signala (iz faze identifikacije in validacije) enake statistične lastnosti: $E\{\mathbf{e}_k \mathbf{e}_k^T\} = E\{\mathbf{e}_k^* \mathbf{e}_k^{*T}\} = \hat{\sigma}_k^2$ (tj. varianca $\mathbf{e}_k$) ter sta medsebojno nekorelirana $E\{\mathbf{e}_k \mathbf{e}_k^{*T}\} = E\{\mathbf{e}_k^* \mathbf{e}_k^T\} = 0$. Tako dobimo:

$$\text{cov}(\mathbf{z}_k^* - \hat{\mathbf{z}}_k^*) = \hat{\sigma}_k^2 \mathbf{I} + \hat{\sigma}_k^2 \Psi_k^{*T} (\Psi_k \Psi_k^T)^{-1} \Psi_k^*. \quad (6.50)$$

Na osnovi slednje kovariance lahko definiramo spodnjo in zgornjo mejo intervala zaupanja [203] za lokalni linearni model k :

$$\underline{f}_k(\mathbf{u}_i^*) = \Psi_{k,i}^{*T} \hat{\boldsymbol{\theta}}_{v_k} - t_{\alpha, M-n_u} \hat{\sigma}_k (1 + \Psi_{k,i}^{*T} (\Psi_k \Psi_k^T)^{-1} \Psi_{k,i}^*)^{\frac{1}{2}}, i = 1, \dots, M, \quad (6.51)$$

$$\overline{f}_k(\mathbf{u}_i^*) = \Psi_{k,i}^{*T} \hat{\boldsymbol{\theta}}_{v_k} + t_{\alpha, M-n_u} \hat{\sigma}_k (1 + \Psi_{k,i}^{*T} (\Psi_k \Psi_k^T)^{-1} \Psi_{k,i}^*)^{\frac{1}{2}}, i = 1, \dots, M, \quad (6.52)$$

kjer je $t_{\alpha, M-n_u}$ konstanta določena na podlagi t -porazdelitve pri izbranem intervalu zaupanja $100(1 - 2\alpha) \%$ oz. $100(1 - \alpha)$ percentilu določenim z zgornjo mejo (npr. pri izbranem intervalu zaupanja $100(1 - 0,1) = 90 \%$, kjer je $\alpha = 0,05$, zgornja meja predstavlja $100(1 - 0,05) = 95$ percentil verjetnostne porazdelitve). Število prostostnih stopenj je enako: $M - n_u$ (n_u je dolžina vektorja $\mathbf{u}$). V enačbah za zgornjo

in spodnjo mehko mejo člen $\psi_{k,i}^*{}^T \hat{\theta}_{v_k}$ predstavlja izhod k -tega lokalnega linearnega modela za točko i .

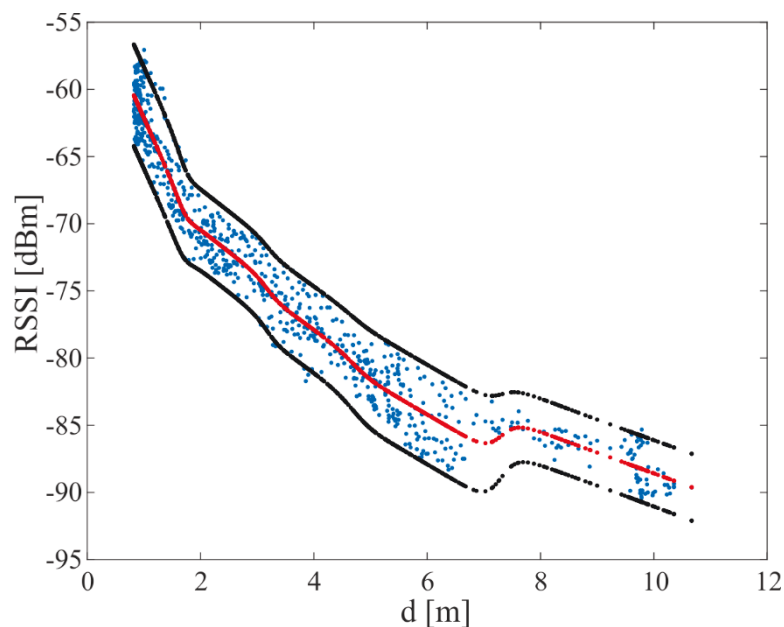
6.2.4 Rezultati gradnje mehkih modelov

Z uporabo algoritma SUHICLUST smo na podlagi meritev jakosti Bluetooth signalov (zajetih po poti, ki je prikazana na sliki 6.9) za vse oddajnike v prostoru zgradili mehke identifikacijske modele, ki opisujejo spreminjanje jakosti signalov v odvisnosti od razdalje med Bluetooth oddajnikom in sprejemnikom. Za vsak posamezen oddajnik so bili parametri (maksimalno število podmodelov, faktor glajenja itd.) algoritma SUHICLUST ustrezno prilagojeni, tako da modeli, kar se da, dobro opisujejo meritve. V vseh primerih gradnje mehkega modela je bil izbran faktor glajenja $\gamma = 2$.

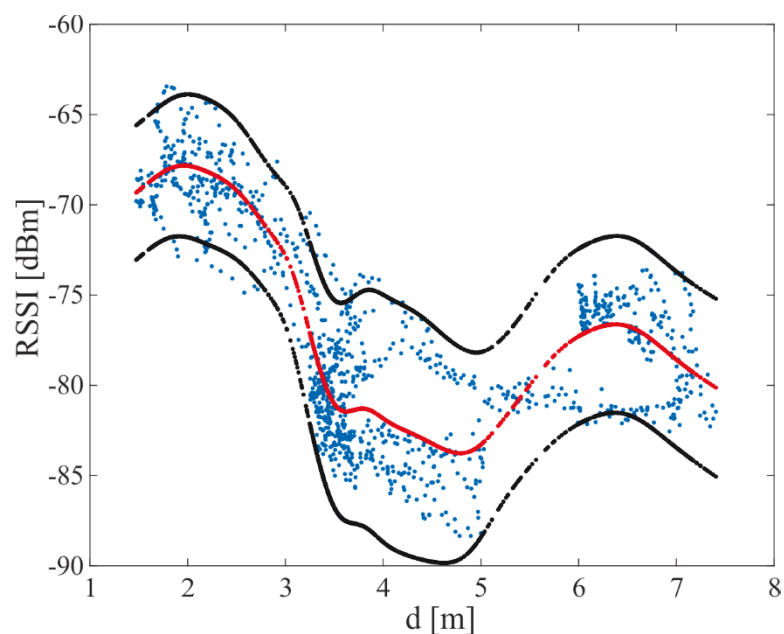
Na sliki 6.20 vidimo primer prileganja mehkega modela (rdeče pike) na meritve jakosti Bluetooth signalov oddajnika z MAC naslovom E3:10:72:E5:75:09. Za ta oddajnik je bil mehki model sestavljen iz petih lokalnih linearnih podmodelov. Na sliki 6.20 so prikazane tudi meje (črne pike) območja s 95 % stopnjo zaupanja. Stopnjo prileganja mehkega modela k zajetim meritvam lahko ovrednotimo s kriterijsko funkcijo RMSE (ang. Root Mean Square Error). Za mehki model, ki je prikazan na sliki 6.20, je RMSE enak 1,67, za model (dobljen z omejeno nelinearno optimizacijo), ki je prikazan na sliki 6.12, pa je RMSE enak 1,84. V obeh primerih je bila uporabljena isto množica meritev jakosti signalov (iz oddajnika z MAC naslovom E3:10:72:E5:75:09). Za ta oddajnik se stopnji prileganja modelov ne razlikujeta prav veliko, saj meritve izražajo pričakovano eksponentno padanje jakosti signala, ki ga je moč dobro opisati tudi z modelom (6.2).

Na sliki 6.15 smo lahko videli, da meritev jakosti signalov iz oddajnika z MAC naslovom D9:50:3B:F6:AA:46 ni mogoče dobro opisati z modelom (6.2). Meritve namreč ne izražajo eksponentnega padanja jakosti signala z razdaljo, hkrati pa je tudi razpon jakosti signalov pri določeni razdalji od oddajnika zelo velik (zaradi prisotnosti ovir). Kot lahko vidimo na sliki 6.21, je z uporabo algoritma SUHICLUST mogoče doseči boljše prileganje modela na celotnem območju razdalje d (oddaljenost med oddajnikom in sprejemnikom). Ker je v tem primeru celoten model sestavljen iz več (natančneje desetih) lokalnih modelov, je mogoče pri določenih razdaljah d , kjer so

meritve zelo razpršene, doseči boljši kompromis v smislu, da model približno enako dobro opisuje jakost signala po celotni krožnici okoli oddajnika.



Slika 6.20: Prileganja mehkega modela (s petimi linearnimi podmodeli) na meritve jakosti Bluetooth signalov oddajnika z MAC naslovom E3:10:72:E5:75:09

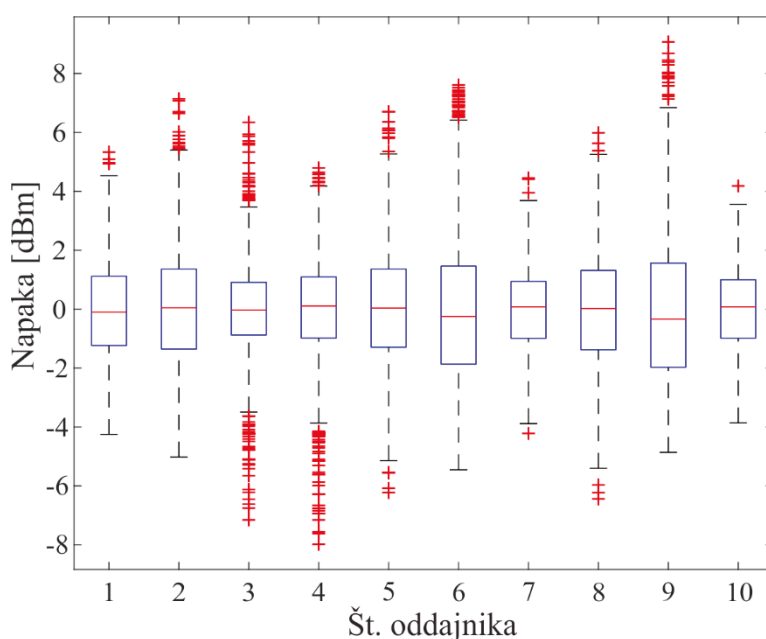


Slika 6.21: Prileganja mehkega modela (z desetimi linearnimi podmodeli) na meritve jakosti Bluetooth signalov oddajnika z MAC naslovom D9:50:3B:F6:AA:46

Če izračunamo stopnjo prileganja obeh modelov (prikazana na sliki 6.15 in sliki 6.21), dobimo za prvi model (opisan z enačbo (6.2)) RMS enak 4,66 in za mehki model

2,57. Torej je v tem primeru zelo velika razlika v kvaliteti modelov, če uporabimo nelinearno optimizacijo in model (6.2) ali model zgrajen z algoritmom SUHICLUST.

Na sliki 6.22 je prikazan graf »škatla z brki« (ang. box plot), ki prikazuje razpon napak za vse identificirane mehke modele. Modri okvirji na sliki 6.22 zajemajo 50 % vseh napak pri posameznem modelu. Z rdečimi križci pa so označene zelo odstopajoče napake (ang. outliers), ki so večje od $Q3 + 1,5 * IQR$ ali manjše od $Q1 - 1,5 * IQR$ (med ti dve meji se uvrsti 99,3 % vseh napak), kjer $Q1$ označuje prvi kvartil (25. percentila), $Q3$ tretji kvartil (75. percentila) in $IQR = Q3 - Q1$.



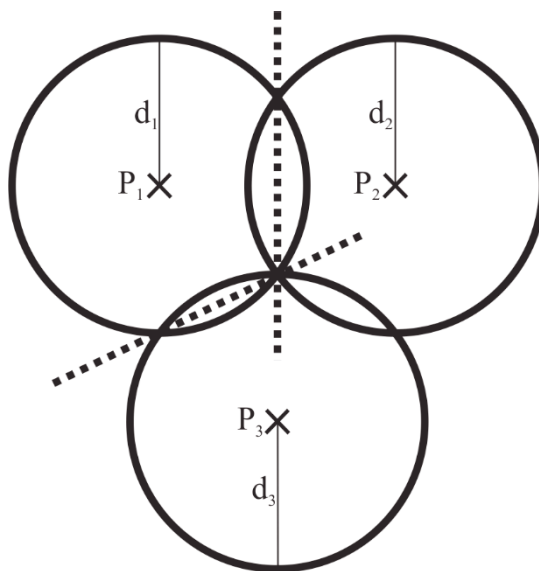
Slika 6.22: Razpon napak za vse identificirane mehke modele

Če primerjamo sliki 6.16 in 6.22, lahko opazimo, da so pri uporabi algoritma SUHICLUST napake modelov manjše kot pri uporabi modela (6.2). Najbolj očitna razlika je ravno pri oddajniku št. 6 (z MAC naslovom D9:50:3B:F6:AA:46), katerega modela sta prikazana na sliki 6.15 (model (6.2)) in 6.21 (mehki model).

6.3 Določanje trenutnega položaja s trilateracijo

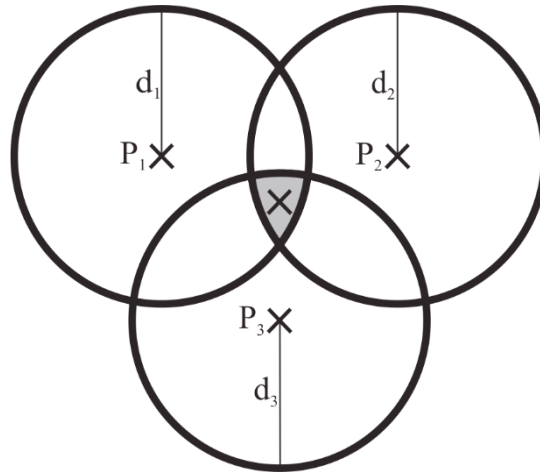
Trilateracija je geometrijska tehnika, ki se je uveljavila pri lokalizaciji z uporabo globalnega navigacijskega satelitskega sistema (GPS, GLONASS, BeiDou). Kot pove že njeno ime, gre pri trilateraciji za vrsto lateracije, pri kateri so za določanje položaja

potrebne vsaj tri bazne postaje (oddajniki), katerih položaj je poznan [204, 205]. Informacijo o oddaljenosti (razdalje d_1 , d_2 , d_3 na sliki 6.23) sprejemnika (uporabnika) od posameznega oddajnika (točke P_1 , P_2 in P_3 na sliki 6.23), ki jo dobimo z upoštevanjem modela jakosti signala, predstavimo s krožnico.



Slika 6.23: Trilateracija

Pri uporabi treh oddajnikov (to pomeni, da imamo tri krožnice z znanim polmerom d) lahko s trilateracijo določimo presečišče treh krožnic, ki nam pove trenutni položaj sprejemnika (uporabnika). Presečišče treh krožnic pa obstaja le v idealnem primeru, ko so razdalje med oddajniki in sprejemnikom natančno določene. Ker pa v realnosti razdalje med sprejemnikom in oddajnikom niso določene povsem natančno (meritve jakosti signalov vsebujejo šum ter tudi modeli širjenja signalov niso popolni), se lahko zgodi, da se tri krožnice sploh ne sekajo oz. lahko obstaja več presečišč treh krožnic (slika 6.24). Zato pri določanju položaja s trilateracijo uporabljamo metode [205], ki minimizirajo napako (metoda najmanjših kvadratov ali metoda najmanjših absolutnih residuov). Na sliki 6.24 je optimalna rešitev trilateracije (pri minimalni napaki) prikazana s križcem znotraj preseka treh krožnic.



Slika 6.24: Realen problem trilateracije, kjer se krožnice ne sekajo v eni točki

6.3.1 Določanje položaja z uporabo lineariziranega modela

Linearni problem najmanjših kvadratov

Uporaba metode linearnih najmanjših kvadratov [206] predstavlja zelo razširjen pristop k reševanju problema, pri katerem je potrebno izračunati parametre izbranega linearnega modela tako, da se ta (v smislu najmanjših kvadratov) dobro prilega danim podatkom. Matematično gledano gre pri metodi linearnih najmanjših kvadratov za aproksimativno reševanje (predoločenega) sistema linearnih enačb, kjer je najboljša rešitev tista, ki minimizira vsoto kvadratov razlik med vrednostmi iz podatkov in pripadajočimi vrednostmi iz modela. Problemi, ki jih rešujemo z metodo linearnih najmanjših kvadratov, so konveksni in imajo končno rešitev (ang. closed-form solution), ki je enolična v primeru, da imamo vsaj toliko podatkovnih točk (in linearnih enačb) kot je število neznanih parametrov modela. Pri metodi nelinearnih najmanjših kvadratov pa problem, ki je lahko nekonveksen z več lokalnimi optimumi za dano funkcijo, rešujemo iterativno.

Denimo, da imamo predoločen sistem m linearnih enačb, ki imajo n neznanih spremenljivk $\beta_1, \beta_2, \dots, \beta_n$ (kjer velja $m > n$). Ta sistem lahko zapišemo kot:

$$y_i = \sum_{j=1}^n X_{ij} \beta_j \quad (i = 1, 2, \dots, m) \quad (6.53)$$

ali v matrični obliki kot:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}, \quad (6.54)$$

kjer je

$$\mathbf{X} = \begin{bmatrix} X_{11} & \cdots & X_{1n} \\ \vdots & \ddots & \vdots \\ X_{m1} & \cdots & X_{mn} \end{bmatrix}, \mathbf{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}. \quad (6.55)$$

Ker sistem (6.54) v splošnem nima rešitve, moramo poiskati take parametre $\mathbf{\beta}$, da se bo linearni model (v smislu najmanjših kvadratov) najbolj prilagal danim podatkom. Torej iščemo minimum funkcije $S(\mathbf{\beta})$:

$$\hat{\mathbf{\beta}} = \underset{\mathbf{\beta}}{\operatorname{argmin}} S(\mathbf{\beta}), \quad (6.56)$$

kjer je

$$S(\mathbf{\beta}) = \sum_{i=1}^m (y_i - \sum_{j=1}^n X_{ij}\beta_j)^2 = \|\mathbf{y} - \mathbf{X}\mathbf{\beta}\|^2. \quad (6.57)$$

Če definiramo i -ti residu kot:

$$r_i = y_i - \sum_{j=1}^n X_{ij}\beta_j, \quad (6.58)$$

potem lahko poenostavljeno zapišemo kriterijsko funkcijo kot:

$$S = \sum_{i=1}^m r_i^2. \quad (6.59)$$

Funkcija S doseže minimum, ko je gradientni vektor enak nič. Elementi gradientnega vektorja so določeni kot parcialni odvodi funkcije S glede na parametre β :

$$\frac{\partial S}{\partial \beta_j} = 2 \sum_{i=1}^m r_i \frac{\partial r_i}{\partial \beta_j} \quad (j = 1, 2, \dots, n). \quad (6.60)$$

Parcialni odvodi so:

$$\frac{\partial r_i}{\partial \beta_j} = -X_{ij}. \quad (6.61)$$

Če v gradientno enačbo vstavimo izraza za residue in parcialne odvode, dobimo

$$\frac{\partial S}{\partial \beta_j} = 2 \sum_{i=1}^m (y_i - \sum_{k=1}^n X_{ik}\beta_k) (-X_{ij}) \quad (j = 1, 2, \dots, n) \quad (6.62)$$

in če vemo, da vektor parametrov $\hat{\mathbf{\beta}}$ minimizira funkcijo S , potem lahko zapišemo:

$$2 \sum_{i=1}^m (y_i - \sum_{k=1}^n X_{ik}\hat{\beta}_k) (-X_{ij}) = 0 \quad (j = 1, 2, \dots, n). \quad (6.63)$$

Če zapis enačb (6.63) nekoliko preuredimo, lahko dobimo sistem enačb v normalni obliki:

$$\sum_{i=1}^m \sum_{k=1}^n X_{ij}X_{ik}\hat{\beta}_k = \sum_{i=1}^m X_{ij}y_i \quad (j = 1, 2, \dots, n) \quad (6.64)$$

oz. v matrični obliki:

$$(\mathbf{X}^T \mathbf{X}) \hat{\mathbf{\beta}} = \mathbf{X}^T \mathbf{y}. \quad (6.65)$$

Če je matrika $\mathbf{X}$ polnega ranga (vsi stolpci matrike $\mathbf{X}$ so linearno neodvisni in posledično je $\mathbf{X}^T \mathbf{X}$ pozitivno definitna matrika), potem je izpolnjen pogoj za izračun

minimuma funkcije S . Rešitev sistema normalnih enačb je vektor optimalnih parametrov $\hat{\beta}$, za katerega je funkcija S minimalna:

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (6.66)$$

Matematično gledano je rešitev (6.66) povsem korektna, vendar pa ta način reševanja sistema enačb ni računsko učinkovit [206] zaradi računanja inverza matrike $\mathbf{X}^T \mathbf{X}$. Če je matrika $\mathbf{X}^T \mathbf{X}$ dobro pogojena in pozitivno definitna (je polnega ranga), je normalno enačbo (6.64) bolje rešiti direktno z uporabo razcepa Choleskega $\mathbf{R}^T \mathbf{R}$, kjer je $\mathbf{R}$ zgornje trikotna matrika:

$$\mathbf{R}^T \mathbf{R} \hat{\beta} = \mathbf{X}^T \mathbf{y}. \quad (6.67)$$

V tem primeru dobimo rešitev v dveh korakih: prvi korak je zamenjava »naprej« (ang. forward substitution) $\mathbf{R}^T \mathbf{z} = \mathbf{X}^T \mathbf{y}$ in drugi korak je zamenjava »nazaj« (ang. backward substitution) $\mathbf{R} \hat{\beta} = \mathbf{z}$.

Problem najmanjših kvadratov lahko rešujemo tudi z metodami ortogonalne dekompozicije [206]. Ta način je numerično bolj stabilen (ker se ne računa produkt $\mathbf{X}^T \mathbf{X}$), vendar počasnejši v primerjavi z metodo normalnih enačb.

Prvi način za reševanje problema najmanjših kvadratov, ki temelji na ortogonalni dekompoziciji, vključuje QR razcep matrike $\mathbf{X}$. Residue zapišemo v matrični obliki kot $\mathbf{r} = \mathbf{y} - \mathbf{X} \hat{\beta}$. Razcep matrike $\mathbf{X}$ je enak $\mathbf{X} = \mathbf{Q} \mathbf{R}$, kjer je $\mathbf{Q}$ ortogonalna matrika ($\mathbf{Q}^T \mathbf{Q} = \mathbf{I}$) velikosti $m \times n$ in $\mathbf{R}$ zgoraj trikotna matrika ($r_{ii} > 0$) velikosti $n \times n$. Minimum funkcije $S = \|\mathbf{r}\|^2$ dobimo pri vektorju parametrov $\hat{\beta}$, ki je rešitev enačbe $\mathbf{R} \hat{\beta} = \mathbf{Q}^T \mathbf{y}$. Slednji sistem je enostavno rešljiv, saj je $\mathbf{R}$ zgoraj trikotna matrika.

Razcep matrike $\mathbf{X}$ pa je mogoče zapisati tudi z uporabo singularnega razcepa (ang. singular value decomposition - SVD): $\mathbf{X} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$, kjer je sta $\mathbf{U}_{m \times m}$ in $\mathbf{V}_{n \times n}$ ortogonalni matriki in $\mathbf{\Sigma}_{m \times n}$ diagonalna matrika. Če rešitev za optimalne parametre $\hat{\beta}$ zapišemo z Moore-Penrose psevdoinverzom $\mathbf{X}^+$ [207, 208] in upoštevamo singularni razcep $\mathbf{X} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$, dobimo:

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = \mathbf{X}^+ \mathbf{y} = \mathbf{V} \mathbf{\Sigma}^+ \mathbf{U}^T \mathbf{y}, \quad (6.68)$$

kjer je $\mathbf{X}^+ = \mathbf{V} \mathbf{\Sigma}^+ \mathbf{U}^T$. Psevdoinverz $\mathbf{\Sigma}^+$ je možno enostavno izračunati tako, da vzamemo elemente na diagonali matrike $\mathbf{\Sigma}$ in jih zamenjamo z njihovimi inverznimi vrednostmi. Reševanje problema najmanjših kvadratov z uporabo singularnega

razcepa ni časovno optimalno, vendar pa je zelo uporabno v primeru, ko je matrika $\mathbf{X}^T \mathbf{X}$ slabo pogojena.

Opis linearnega sistema enačb trilateracije

Razdalja med i -tim Bluetooth oddajnikom s položajem (x_{bi}, y_{bi}) in sprejemnikom (mobilna naprava) s položajem (x, y) je določena kot:

$$d_i = \sqrt{(x - x_{bi})^2 + (y - y_{bi})^2}. \quad (6.69)$$

Razdalja d_i predstavlja polmer krožnice s središčem v (x_{bi}, y_{bi}) , na kateri se lahko nahaja sprejemnik. Za določanje trenutnega položaja sprejemnika (x, y) uporabimo m oddajnikov, za katere poznamo njihov položaj (x_{bi}, y_{bi}) . Z modelom jakosti signala $d = f(RSSI_i)$ določimo razdaljo med trenutnim položajem (x, y) in i -tim oddajnikom (x_{bi}, y_{bi}) . Na ta način dobimo sistem m nelinearnih enačb (trilateracije), ki jih lineariziramo [209], da dobimo sistem linearnih enačb v obliki:

$$\mathbf{X}\boldsymbol{\beta} = \mathbf{y}, \quad (6.70)$$

kjer je

$$\mathbf{X} = \begin{bmatrix} x_{b2} - x_{b1} & y_{b2} - y_{b1} \\ \vdots & \vdots \\ x_{bm} - x_{b1} & y_{bm} - y_{b1} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} x - x_{b1} \\ y - y_{b1} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} b_{21} \\ \vdots \\ b_{m1} \end{bmatrix}, \quad (6.71)$$

$$b_{i1} = \frac{1}{2}(d_1^2 - d_i^2 + l_{i1}^2) \quad (i = 2, \dots, m), \quad (6.72)$$

$$l_{i1} = \sqrt{(x_{bi} - x_{b1})^2 + (y_{bi} - y_{b1})^2}. \quad (6.73)$$

Ker gre indeks $i = 2, \dots, m$, imamo sistem $(m - 1)$ linearnih enačb s dvema neznankama, kar pomeni, da potrebujemo vsaj tri oddajnike za enolično rešitev. Sistem (6.70) lahko rešimo analitično z metodo najmanjših kvadratov kot:

$$\boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}, \quad (6.74)$$

pri čemer minimiziramo funkcijo:

$$S = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}). \quad (6.75)$$

Rešitev (ki jo dobimo z uporabo razcepa Choleskega ali QR razcepa) obstaja, če je matrika $\mathbf{X}^T \mathbf{X}$ nesingularna in dobro pogojena. V nasprotnem primeru pa moramo uporabiti singularni razcep (ang. Singular Value Decomposition – SVD) in psevdoinverz matrike $\mathbf{X}$:

$$\boldsymbol{\beta} = \mathbf{X}^+ \mathbf{y}. \quad (6.76)$$

6.3.2 Določanje položaja z uporabo nelinearne optimizacije

Nelinearni problem najmanjših kvadratov

Pri reševanju nelinearnega problema najmanjših kvadratov gre za iskanje n neznanih parametrov nelinearnega modela, ki v smislu najmanjših kvadratov najbolje opisuje množico m ($m \geq n$) opazovanj. Pri vseh metodah, ki se uporabljajo za namen reševanja nelinearnega problema najmanjših kvadratov, je ključna naloga poiskati približen linearen model in nato v zaporednih iteracijah izboljševati parametre linearnega modela.

Denimo, da imamo množico m podatkovnih točk $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ in funkcijo nelinearnega modela $y = f(x, \beta)$, ki je poleg spremenljivke x odvisna tudi od n parametrov $\beta = (\beta_1, \beta_2, \dots, \beta_n)$, pri čemer je $m \geq n$. Pri reševanju nelinearnega problema najmanjših kvadratov je potrebno poiskati vektor β , tako da se funkcija nelinearnega modela najbolje ujema z danimi podatki v smislu najmanjših kvadratov. To pomeni, da je vsota kvadratov residuov

$$S = \sum_{i=1}^m r_i^2 \quad (6.77)$$

minimalna, pri čemer so residui določeni kot:

$$r_i = y_i - f(x_i, \beta) \quad (\text{za } i = 1, 2, \dots, m). \quad (6.78)$$

Minimalna vrednost S je dosežena, ko je gradient enak nič. Ker model vsebuje n parametrov, imamo potem n gradientnih enačb:

$$\frac{\partial S}{\partial \beta_j} = 2 \sum_i r_i \frac{\partial r_i}{\partial \beta_j} = 0 \quad (j = 1, \dots, n). \quad (6.79)$$

Pri nelinearnih sistemih so odvodi $\frac{\partial r_i}{\partial \beta_j}$ funkcije neodvisne spremenljivke in parametrov β , tako da te gradientne enačbe nimajo analitične rešitve. Parametri se zato računajo z zaporedno aproksimacijo, kar pomeni, da se z vsako novo iteracijo vrednosti parametrov nekoliko izboljšajo:

$$\beta_j \approx \beta_j^{k+1} = \beta_j^k + \Delta \beta_j, \quad (6.80)$$

kjer k označuje trenutno iteracijo in $\Delta \beta_j$ vektor sprememb vrednosti parametrov. Z vsako novo iteracijo je potrebno model linearizirati, tako da se poišče približek s Taylorjevo vrsto prvega reda v točki β^k :

$$f(x_i, \beta) \approx f(x_i, \beta^k) + \sum_j \frac{\partial f(x_i, \beta^k)}{\partial \beta_j} (\beta_j - \beta_j^k) \approx f(x_i, \beta^k) + \sum_j J_{ij} \Delta \beta_j. \quad (6.81)$$

Jacobijeva matrika J je funkcija konstant, neodvisne spremenljivke in parametrov ter se spreminja z vsako novo iteracijo. Zato v smislu lineariziranega modela velja $\frac{\partial r_i}{\partial \beta_j} = -J_{ij}$, kar pomeni, da so residui določeni kot:

$$r_i = \Delta y_i - \sum_{s=1}^n J_{is} \Delta \beta_s, \quad (6.82)$$

kjer je $\Delta y_i = y_i - f(x_i, \beta^k)$. Če slednje izraze upoštevamo v gradientnih enačbah, dobimo n linearnih (normalnih) enačb:

$$\sum_{i=1}^m \sum_{s=1}^n J_{ij} J_{is} \Delta \beta_s = \sum_{i=1}^m J_{ij} \Delta y_i \quad (j = 1, \dots, n) \quad (6.83)$$

in te nato zapišemo v matrični obliki kot:

$$(J^T J) \Delta \beta = J^T \Delta y. \quad (6.84)$$

Če želimo posamezna opazovanja različno obtežiti, potem moramo minimizirati uteženo vsoto kvadratov residuov $S = \sum_{i=1}^m W_{ii} r_i^2$. Idealna vrednost za vsak element W_{ii} , ki leži na diagonali utežnostne matrike $\mathbf{W}$, bi bila enaka recipročni vrednosti variance napake meritve. V tem primeru normalne enačbe zapišemo v matrični obliki kot:

$$(J^T \mathbf{W} J) \Delta \beta = (J^T \mathbf{W}) \Delta y \quad (6.85)$$

in jih lahko rešimo za $\Delta \beta$ z uporabo Gauss-Newtonove (GN) metode, tako da upoštevamo razcep Choleskega. Parametri se izboljšujejo (v primeru konvergence) z vsako novo iteracijo:

$$\beta^{k+1} = \beta^k + \Delta \beta. \quad (6.86)$$

Ta način reševanja nelinearnega problema najmanjših kvadratov ni primeren, če prihaja do divergence. Zato moramo v takem primeru uporabiti metodo, ki z vsako iteracijo prilagaja velikost koraka (spremembo vrednosti parametrov) $\Delta \beta$ in pa tudi njegovo smer (najbolj pogosto se uporablja Levenberg–Marquardt algoritem, ki spada med metode omejenega koraka).

Metoda omejenega koraka

Metoda omejenega koraka [210] (tudi metoda območja zaupanja, ang. Trust region) je postopek reševanja optimizacijskih problemov, pri katerem v zaporednih iteracijah rešujemo približek prvotnega optimizacijskega problema, ki mu postavljamo »omejitev« (tj. predpis, s katerim omejimo množico možnih rešitev optimizacijskega problema) koraka.

Metoda omejenega koraka je eden od dveh prototipnih algoritmov, ki se pretežno uporabljata pri reševanju problemov nelinearnega programiranja. Drug pristop je minimizacija v dani smeri (ang. line search), ki jo pri optimizacijskih problemih z omejitvami v glavnem kombiniramo z metodo množice aktivnih omejitev (ang. active set method). Pri metodi območja zaupanja se najprej izbere velikost koraka (velikost območja zaupanja) in nato smer koraka, pri minimizaciji v dani smeri pa se najprej izbere smer koraka in šele nato velikost koraka.

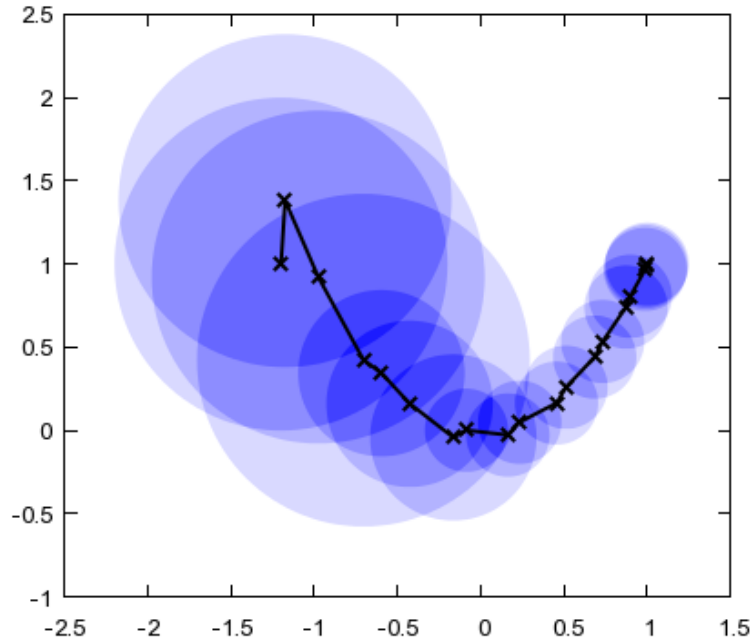
Pri omejitvi koraka zahtevamo, da je razdalja med trenutnim in naslednjim približkom pod predpisano mejo, ki se med postopkom sproti prilagaja. S tem zagotovimo, da poteka reševanje približnega problema na območju, kjer je ta dovolj dober približek prvotnega problema (območje zaupanja). Območje zaupanja predstavlja podmnožico elementov iz območja kriterijske funkcije, ki ga dobro opisuje izbrani model. Če je ta model, ki opisuje kriterijsko funkcijo znotraj območja zaupanja dovolj dober, potem se območje zaupanja v naslednjem koraku razširi. V primeru, ko je približek, določen z modelom, slab, pa se območje zaupanja zmanjša. Omejevanje velikosti koraka je pomembno predvsem v primeru slabe konvergence (ko je začetna ocena zelo oddaljena od optimuma), saj bi v nasprotnem primeru lahko metoda izbrala prevelik korak v napačno smer oz. stran od optimuma.

Metode omejenega koraka se poslužujejo enostavnega in zmogljivega koncepta s področja optimizacije. Da bi razumeli pristop k optimizaciji, ki se uporablja pri teh metodah, bomo najprej obravnavali neomejen minimizacijski problem, pri katerem želimo poiskati minimum funkcije $f(x)$. V splošnem imamo m nelinearnih funkcij $f_i(x)$, pri čemer vektor x vsebuje n elementov (velja $m \geq n$, kar pomeni, da lahko rešujemo tudi predoločen sistem enačb). Cilj reševanja enačb je najti tak vektor x , da bo za vseh m enačb veljalo $f_i(x) = 0$ (oz. bodo vrednosti $f_i(x)$ enake minimumom funkcij).

Denimo, da se v n dimenzionalnem prostoru nahajamo v točki x in da se želimo premakniti v točko, v kateri ima funkcija $f(x)$ manjšo vrednost. Osnovna ideja je poiskati bolj enostavno funkcijo ψ , ki bo dobro opisovala funkcijo f na določenem območju N okoli točke x . To območje imenujemo »območje zaupanja« (ang. trust region). Minimum funkcije f iščemo v zaporednih iteracijah. V vsaki iteraciji izračunamo poskusni korak s_k z minimiziranjem funkcije ψ :

$$\min_s \{\psi(s), s \in N\}. \quad (6.87)$$

Trenutna točka x_k se premakne v točko $x_{k+1} = x_k + s_k$, če velja $f(x_k + s_k) < f(x_k)$, v nasprotnem primeru ostane točka $x_{k+1} = x_k$ nespremenjena, območje zaupanja N se zmanjša ter ponovi se izračun poskusnega koraka (slika 6.25).



Slika 6.25: Zmanjševanje območja zaupanja pri optimizaciji z metodo omejenega koraka

Pri minimiziranju funkcije $f(x)$ z izbrano metodo omejenega koraka je potrebno rešiti tri ključna vprašanja: kako izbrati funkcijo ψ in izračunati približek v trenutni točki x , kako izbrati in nato prilagajati območje zaupanja N ter kako nato natančno rešiti podproblem območja zaupanja.

Pri standardni metodi omejenega koraka [211] (za neomejen minimizacijski problem) je kvadratična aproksimacija ψ določena s prvima dvema členoma Taylorjeve vrste, ki približno opisujeta funkcijo f v točki x . Območje zaupanja N ima običajno sferično ali elipsoidno obliko. Podproblem območja zaupanja se matematično zapiše kot:

$$\min \{\psi_k(s) = \frac{1}{2} s^T H_k s + s^T g_k \text{ pri } \|D_k s\| \leq \Delta_k\}, \quad (6.88)$$

kjer je $\psi_k(s)$ kvadratična funkcija, $g_k = \nabla f(x_k)$ je gradient funkcije f v trenutni točki x_k , $H_k = \nabla^2 f(x_k)$ je Hessejeva matrika (simetrična matrika drugih parcialnih

odvodov), D_k je diagonalna skalirna matrika, Δ_k je pozitiven skalar, ki predstavlja velikost območja zaupanja ter $\|\cdot\|$ je evklidska norma. Za reševanje enačbe (6.88) se lahko uporabijo algoritmi [183], ki vključujejo izračun lastnih vrednosti in lastnih vektorjev ter uporabo Newtonovega procesa za karakteristično enačbo:

$$\frac{1}{\Delta_k} - \frac{1}{\|s_k\|} = 0. \quad (6.89)$$

Ti algoritmi vrnejo natančno rešitev, vendar pa zahtevajo veliko časa, ki je proporcionalen vsem faktorizacijam Hessejeve matrike H . Zato je potreben drugačen pristop pri reševanju problemov z metodami omejenega koraka. V literaturi [212, 213] je bilo predlaganih več aproksimativnih in hevrističnih strategij, ki temeljijo na enačbi (6.88). Aproksimativni pristop, ki se uporablja znotraj Matlabovih funkcij za metode omejenega koraka, vključuje tehniko omejevanja podproblema območja zaupanja na dvodimenzionalni podprostor S ([212, 214]). Ko je podprostor S enkrat določen, je možno povsem enostavno rešiti enačbo (6.88) tudi, če je pri tem potrebna polna informacija o lastnih vrednostih in lastnih vektorjih (problem je le dvodimenzionalen). Torej je v tem primeru glavna naloga določiti podprostor S .

Dvodimenzionalni podprostor S je mogoče določiti z uporabo predpogojene konjugirano gradientne metode (ang. preconditioned conjugate gradients method), opisane v nadaljevanju. Dvodimenzionalni podprostor S je definiran kot linearni prostor med mejama s_1 in s_2 ($s \in [s_1, s_2]$), pri čemer gre $s_1 = -g$ v smeri gradienta g ter s_2 v smeri, ki je približno enaka »Newtonovi smeri« (ang. Newton direction). Slednja meja (ki predstavlja najmanjšo vrednost kvadratičnega modela) je določena kot rešitev enačbe:

$$H \cdot s_2 = -g \quad (6.90)$$

ali kot smer negativne ukrivljenosti (ang. negative curvature), v primeru, ko je matrika H nedefinitna:

$$s_2^T \cdot H \cdot s_2 < 0. \quad (6.91)$$

Namen, ki se skriva za tem načinom izbire podprostora S , je vsiliti globalno konvergenco (z upoštevanjem smeri padanja gradienta ali smeri negativne ukrivljenosti) in doseči hitro lokalno konvergenco (z upoštevanjem Newtonovega koraka, če ta obstaja).

Velikost območja zaupanja Δ_k se prilagaja glede na standardna pravila metod omejenega koraka. To pomeni, da se vrednost Δ_k zmanjša, če poskusni korak s_k ne

izpolni pogoja $f(x_k + s_k) < f(x_k)$, oz. se v nasprotnem primeru poveča ali ostane enako velika [215, 216]. Ali naj se območje zaupanja poveča ali zmanjša, lahko določimo z upoštevanjem razmerja, ki v osnovi predstavlja kriterij pri odločanju, ali naj se trenutna vrednost x spremeni ali ne:

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{\psi_k(0) - \psi_k(s_k)}. \quad (6.92)$$

Če velja, da je $\rho_k > \mu$ (μ je izbrana meja) potem velja: $x_{k+1} = x_k + s_k$ (območje zaupanja Δ_k se lahko v tem primeru poveča [217]), drugače pa ostane vektor x_{k+1} nespremenjen: $x_{k+1} = x_k$ (območje zaupanja Δ_k ostane enako veliko ali pa se zmanjša). $(\psi_k(0) - \psi_k(s_k))$ predstavlja napoved (predikcijo) zmanjšanja funkcije f pri preizkusnem koraku s_k na podlagi kvadratične aproksimacije ψ_k .

Neomejeno minimizacijo z uporabo metode omejenega koraka lahko strnemo v štirih točkah:

- Določimo dvodimenzionalni podproblem območja zaupanja.
- Rešimo enačbo (6.88), da dobimo poskusni korak s_k .
- Če velja $\rho_k > \mu$, potem je nova (boljša) rešitev enaka: $x_{k+1} = x_k + s_k$.
- Prilagodimo Δ_k .

Predpogojena konjugirano gradientna metoda

Predpogojena konjugirano gradientna metoda (ang. preconditioned conjugate gradients method) [218] predstavlja popularen način za reševanje velikih simetričnih (pozitivno definitnih) sistemov linearnih enačb $Hp = -g$ (p predstavlja korak v Newtonovi smeri). Ta iterativen pristop zahteva možnost izračuna produkta $H \cdot v$ med poljubnim vektorjem v in matriko H . V kombinaciji s to metodo se uporablja simetrična pozitivno definitna matrika M , ki predstavlja predpogoj za matriko H . Predpogoj (ang. »preconditioner«) je definiran s transformacijo, ki preoblikuje dani problem v bolj ustrezno obliko za metode, ki temeljijo na numeričnem reševanju problema. To pomeni, da se z določenim predpogojem izboljša konvergenca konjugirano gradientne metode. Za ta predpogoj mora veljati: $M = D^2$, kjer je $D^{-1}HD^{-1}$ dobro pogojena matrika (število pogojenosti (ang. condition number) za $D^{-1}HD^{-1}$ ima veliko manjšo vrednost kot za matriko H).

V reševanju minimizacijskega problema lahko predpostavimo, da je Hessejeva matrika H simetrična. Vendar le v okolici »močnega« minimuma je zagotovljeno, da

je Hessejeva matrika tudi pozitivno definitna. Algoritem za konjugirano gradientno metodo obstaja le v primeru, ko je izpolnjen pogoj $d^T H d \leq 0$ (kjer je d korak v smeri iskanja), ki določa smer negativne ukrivljenosti. Smer p , ki jo določa izhod algoritma za konjugirano gradientno metodo, je enaka smeri negativne ukrivljenosti ali pa smeri, ki jo določa približna rešitev Newtonovega sistema $H p = -g$. V obeh primerih pa smer p pomaga definirati dvodimenzionalni podprostor S (določa mejo s_2).

Omejena nelinearna optimizacija

Pri omejeni minimizaciji gre za iskanje vektorja x , ki predstavlja lokalni minimum skalarne funkcije $f(x)$, pri čemer se upoštevajo podane omejitve za vektor x . Te omejitve so lahko podane kot: linearna omejitev ($A \cdot x \leq b$, $A_{eq} \cdot x = b_{eq}$), nelinearna omejitev ($c(x) \leq 0$ ali $c_{eq}(x) = 0$) ali omejitev s podanimi mejami $l \leq x \leq u$. V nadaljevanju se bomo osredotočili le na nelinearno optimizacijo s podanimi mejami (ang. box constraints) l (vektor spodnjih mej) in u (vektor zgornjih mej) [217]. V tem primeru torej rešujemo naslednji problem:

$$\min\{f(x) \text{ pri } l \leq x \leq u\}. \quad (6.93)$$

Nekateri (ali vsi) elementi vektorja l so lahko enaki $-\infty$, in nekateri (ali vsi) elementi vektorja u so lahko enaki ∞ . Metoda generira zaporedje možnih točk x , ki ustrezajo podanim mejam. Za zagotavljanje izvedljivosti (to pomeni, da je izpolnjen pogoj $l \leq x \leq u$) in robustne konvergence se skupaj uporabljata dve metodi. Prva metoda temelji na (skalirno) modificiranem Newtonovem koraku, ki nadomešča neomejen Newtonov korak (pri določanju dvodimenzionalnega podprostora S). Druga metoda pa temelji na »odbojih« (ang. reflections), ki omogočajo povečanje dolžine korakov.

Skalirno modificiran Newtonov korak sledi iz Kuhn-Tuckerjevega potrebnega pogoja za enačbo (6.93):

$$(D(x))^{-2} g = 0, \quad (6.94)$$

kjer je

$$D(x) = \text{diag}(|v_k|^{-1/2}) \quad (6.95)$$

in vektor $v(x)$ je definiran (za vse $1 \leq i \leq n$) kot:

- Če $g_i < 0$ in $u_i < \infty$, potem $v_i = x_i - u_i$.
- Če $g_i \geq 0$ in $l_i > -\infty$, potem $v_i = x_i - l_i$.
- Če $g_i < 0$ in $u_i = \infty$, potem $v_i = -1$.

- Če $g_i \geq 0$ in $l_i = -\infty$, potem $v_i = 1$.

Nelinearni sistem enačb (6.94) ni na celotnem območju odvedljiv. Težava z odvedljivostjo se pojavi pri $v_i = 0$. Taki točki se lahko izognemo z upoštevanjem stroge omejitve $l < x < u$ (ang. strict feasibility). Skalirno modificiran Newtonov korak s_k za nelinearen sistem enačb (podan z enačbo (6.94)) je definiran kot rešitev linearnega sistema (pri k -ti iteraciji):

$$\hat{M}_k D_k s_k^N = -\hat{g}_k, \quad (6.96)$$

kjer je:

$$\hat{g}_k = D_k^{-1} g_k = \text{diag}(|v_k|^{1/2}) g_k \quad (6.97)$$

in

$$\hat{M}_k = D_k^{-1} H_k D_k^{-1} + \text{diag}(g_k) J_k^v. \quad (6.98)$$

V enačbi (6.98) $J^v(x)$ predstavlja Jacobijevo matriko vektorja $|v(x)|$. Diagonalni elementi (diagonalne) matrike J^v so enaki 0, ± 1 . Če so vsi elementi vektorjev l in u končni, potem je Jacobijeva matrika enaka: $J^v = \text{diag}(\text{sign}(g))$. V točki, kjer je $g_i = 0$, v_i mogoče ni odvedljiva. Za take točke je definirana vrednost $J_{ii}^v = 0$.

Enačba (6.96) upošteva uporabo afine transformacije $\hat{x} = D_k x$ (matrika D_k je simetrična matrika: $D_k = D_k^T$). Ta transformacija pretvori problem omejene nelinearne optimizacije v problem neomejene nelinearne optimizacije, kjer minimiziramo funkcijo v novih koordinatah $\hat{x}$. To pomeni, da se v tem primeru rešuje minimizacijski problem:

$$\min\{\hat{\psi}_k(\hat{s}) = \frac{1}{2} \hat{s}^T \hat{M}_k \hat{s} + \hat{g}_k^T \hat{s} \text{ pri } \|\hat{s}\| \leq \Delta_k\} \quad (6.99)$$

oz., če upoštevamo $s = D_k^{-1} \hat{s}$ (dobimo v prvotnem prostoru):

$$\min\{\psi_k(s) = \frac{1}{2} s^T M_k s + s^T g_k \text{ pri } \|D_k s\| \leq \Delta_k\}, \quad (6.100)$$

kjer je:

$$C_k = D_k \text{diag}(g_k) J_k^v D_k, \quad (6.101)$$

$$M_k = H_k + C_k. \quad (6.102)$$

Pri drugi metodi, ki zagotavlja izvedljivost (da je izpolnjen pogoj $l \leq x \leq u$), pa se upoštevajo »odboji« (ang. reflections) za namen povečanja koraka. Posamezen refleksijski korak je definiran na naslednji način. Če imamo dan korak p , ki seka

postavljeno mejo (i -to spodnjo ali zgornjo mejo), potem je refleksijski korak p^R določen kot $p^R = p$, razen i -ta komponenta je enaka: $p_i^R = -p_i$.

Metoda najmanjših absolutnih residuov

Metoda najmanjših absolutnih residuov (ang. Least absolute residual - LAR) predstavlja statistično optimalni kriterij, ki se lahko uporablja (namesto vsote kvadratov residuov) v kombinaciji z metodo omejenega koraka za doseg robustne nelinearne optimizacije.

Če so residui določeni kot:

$$r_i = y_i - f(x_i) \text{ (za } i = 1, 2, \dots, m) \quad (6.103)$$

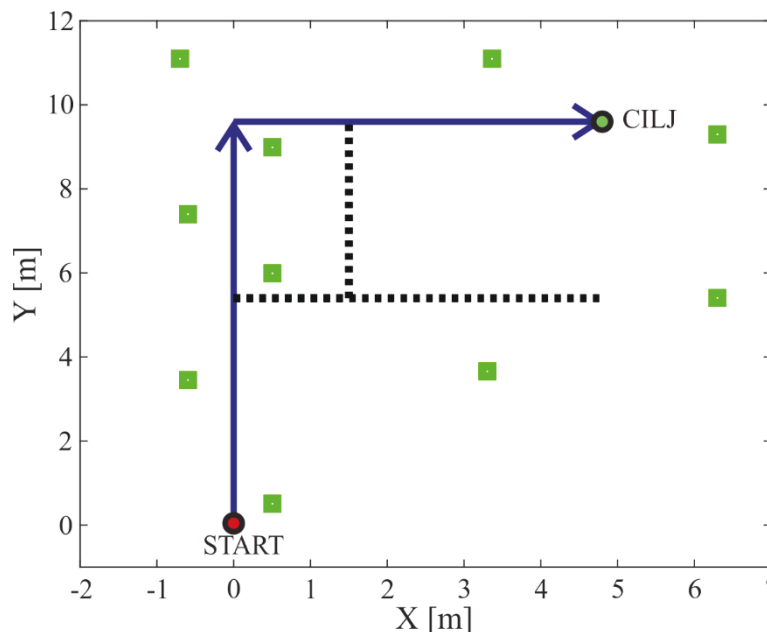
in vsota absolutnih residuov kot:

$$S = \sum_{i=1}^m |r_i|, \quad (6.104)$$

potem je cilj optimizacije poiskati take parametre funkcije f , da bo vsota S minimalna. Metoda LAR je v primerjavi [219, 220] z metodo najmanjših kvadratov bolj robustna, saj imajo odstopajoče točke (ang. outliers) pri njeni uporabi manjši vpliv na kriterijsko funkcijo. Pri metodi najmanjših kvadratov imajo zelo odstopajoče točke večji vpliv zaradi kvadriranja. Slabost metode najmanjših absolutnih residuov je ta, da v nekaterih primerih uporabe tega kriterija obstaja več rešitev (npr. ko iščemo parametre enačbe premice in so vse točke iz množice podatkov razporejene simetrično glede na horizontalno premico).

6.4 Rezultati lokalizacije na osnovi trilateracije

V laboratoriju velikosti približno 85 m² smo za namen preizkusa algoritmov lokalizacije, ki temeljijo na merjenju jakosti Bluetooth signalov, namestili 10 oddajnikov podjetja Kontakt. Oddajnike smo razporedili po prostoru, kot je prikazano na sliki 6.26. Da bi prisotnost človeškega telesa v prostoru čim manj vplivala na meritve jakosti signalov, smo oddajnike namestili na višino 2 m od tal. Oddajniki so bili nastavljeni tako, da so oddajali pakete vsakih 100 ms z maksimalno močjo (TxPower) 4 dBm.



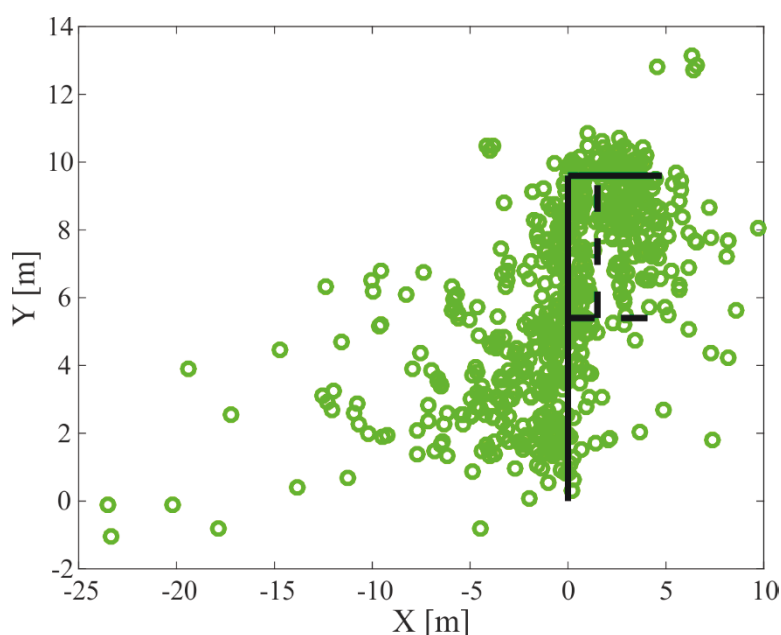
Slika 6.26: Položaj Bluetooth oddajnikov v prostoru

6.4.1 Lokalizacija z uporabo trilateracije in metode linearnih najmanjših kvadratov

Za preizkus lokalizacijskega algoritma, ki temelji na trilateraciji (opisani z linearnim sistemom enačb (6.70)) in metodi linearnih najmanjših kvadratov, smo uporabili predhodno zgrajene modele širjenja signalov v prostoru (ki so opisani v poglavju 6.2). Ti namreč omogočajo izračun razdalje med posameznim oddajnikom in pa trenutnim položajem sprejemnika (pametnega telefona) glede na meritve jakosti Bluetooth signalov (RSSI).

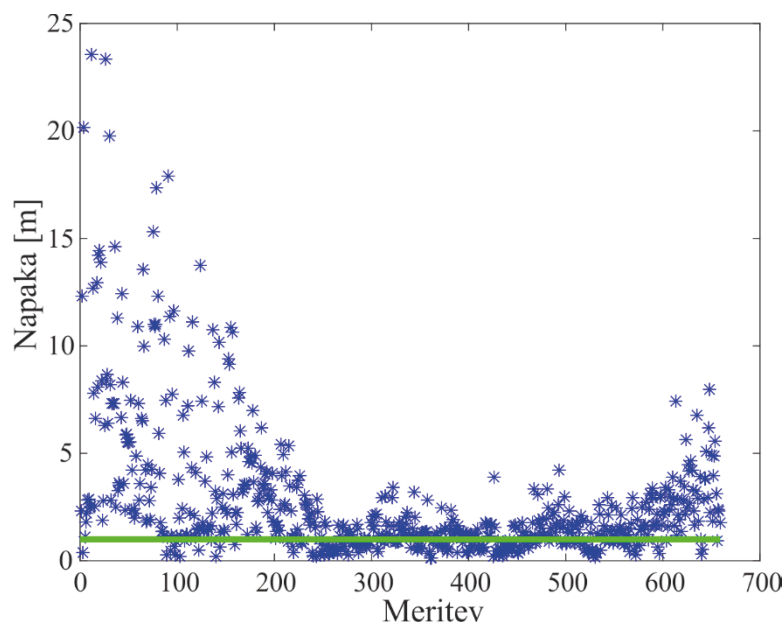
S pametnim telefonom Android so bile zajete meritve jakosti signalov za celotno testno pot (od začetne točke »START« do končne točke »CILJ«), ki je prikazana na sliki 6.26. Za določanje trenutnega položaja v prostoru se upošteva zadnjih pet meritev (to so najnovejše meritve) iz različnih oddajnikov, kar pomeni, da pri določanju položaja po principu trilateracije (kjer so potrebni vsaj trije oddajniki) iščemo presečišče petih krogov. Predoločen sistem enačb, ki ga dobimo z uporabo lineariziranih enačb trilateracije (6.70), smo reševali v okolju Matlab s funkcijo »mldivide« (levo deljenje $\backslash$, ki glede na sistem izbere eno izmed metod: razcep Choleskega, QR razcep, LDL razcep, LU razcep itd.) tako, da je bila napaka minimalna v smislu najmanjših kvadratov. Rezultati tega pristopa k reševanju lokalizacijskega

problema so prikazani na sliki 6.27, na kateri lahko opazimo, da so zeleni krogi, ki predstavljajo izračunani položaj, zelo oddaljeni (do 25 m) od poti (črna črta na sliki 6.27), po kateri je hodil pešec opremljen s pametnim telefonom. Slabi rezultati so posledica dejstva, da so razdalje med oddajniki in sprejemnikom zelo nenatančno določene, saj jakosti signalov zelo variirajo v isti točki (v signalu je prisotno veliko šuma) in tudi sami modeli širjenja signalov niso popolni. Z uporabo trilateracije, ki temelji na sistemu lineariziranih enačb, lahko dobimo ustrezno dober rezultat le v primeru, ko so razdalje med oddajniki in sprejemnikom natančno določene.



Slika 6.27: Določanje položaja s trilateracijo in linearno metodo najmanjših kvadratov

Na sliki 6.28 so prikazane napake pri določanju položaja v prostoru za vse meritve od starta do cilja poti. Napake so izračunane kot evklidske razdalje med dejanskim položajem in izračunanim položajem sprejemnika v prostoru. Dejanski položaj smo določili z lokalizacijskim algoritmom, ki temelji na fuziji vizualne odometrije in inercialnega navigacijskega sistema [196]. Zelena črta na sliki 6.28 prikazuje naš cilj (da bi bila maksimalna napaka manjša od 1 m), ki smo si ga zastavili pri razvoju algoritmov za lokalizacijo na osnovi jakosti radijskih signalov.



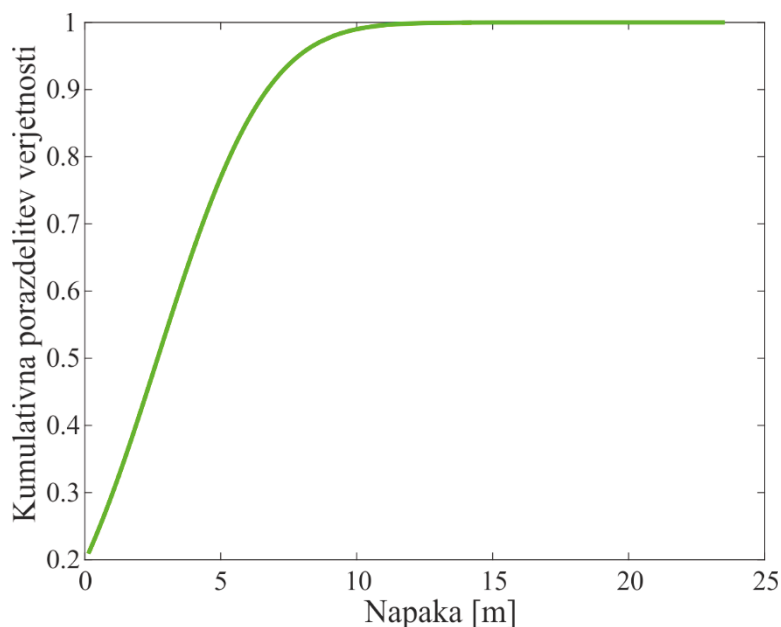
Slika 6.28: Napaka pri določanju položaja s trilateracijo in linearno metodo najmanjših kvadratov

Za prikaz natančnosti lokalizacijskega pristopa se v literaturi pogosto uporablja graf kumulativne porazdelitve verjetnosti napake, ki ga lahko vidimo na sliki 6.29. Ta nam pove, kakšna je verjetnost, da bo napaka manjša od določene vrednosti. Pri lokalizacijskih algoritmi si želimo, da bi graf kumulativne porazdelitve verjetnosti napake čim hitreje dosegel vrednost ena, saj to pomeni, da je v tem primeru večina napak majhnih (so skoncentrirane pri nižjih vrednostih). Za opisani eksperiment, pri katerem smo za lokalizacijo uporabili trilateracijo na osnovi sistema lineariziranih enačb in metode najmanjših kvadratov, lahko razberemo iz slike 6.29, da je npr. za 77 % ocenjenih položajev napaka manjša od 5 m in za 99 % ocenjenih položajev manjša od 10 m (99 % vseh napak je manjših od 10 m).

6.4.2 Lokalizacija z uporabo trilateracije in omejene nelinearne optimizacije

Ker pri reševanju problema trilateracije z linearnim sistemom enačb in metodo linearnih najmanjših kvadratov nismo dosegli ustrezne lokalizacijske natančnosti, smo se lotili reševanja nelinearnega sistema enačb trilateracije $(x - x_{bi})^2 + (y - y_{bi})^2 - d_i^2 = 0$ (kjer je (x_{bi}, y_{bi}) položaj i -tega oddajnika, (x, y) trenutni položaj sprejemnika in d_i razdalja med i -tim oddajnikom in sprejemnikom) z omejeno nelinearno optimizacijo. Uporabili smo metodo omejenega koraka in kriterij, ki temelji na metodi absolutnih najmanjših residuov (da imajo odstopajoče točke manjši vpliv na kriterijsko

funkcijo). Za ta primer smo uporabili iste meritve jakosti Bluetooth signalov kot pri lokalizaciji z uporabo trilateracije in metode linearnih najmanjših kvadratov.

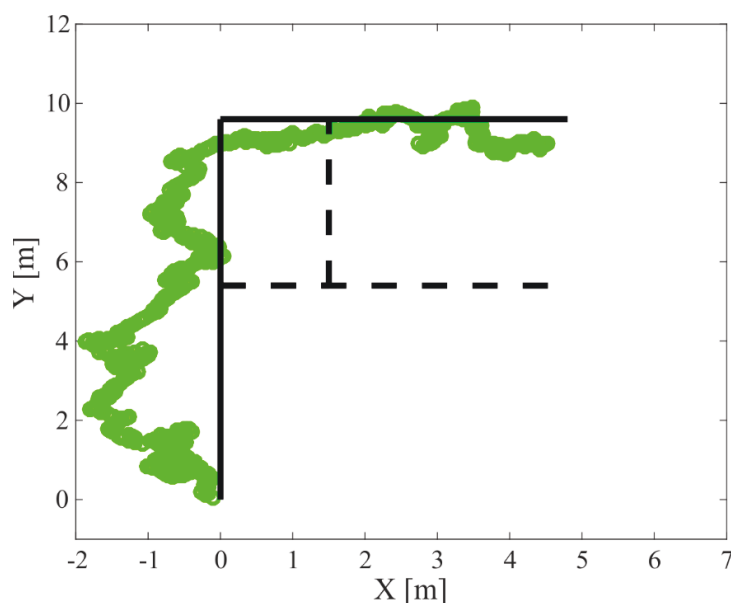


Slika 6.29: Natančnost lokalizacije podana s kumulativno porazdelitvijo verjetnosti napake

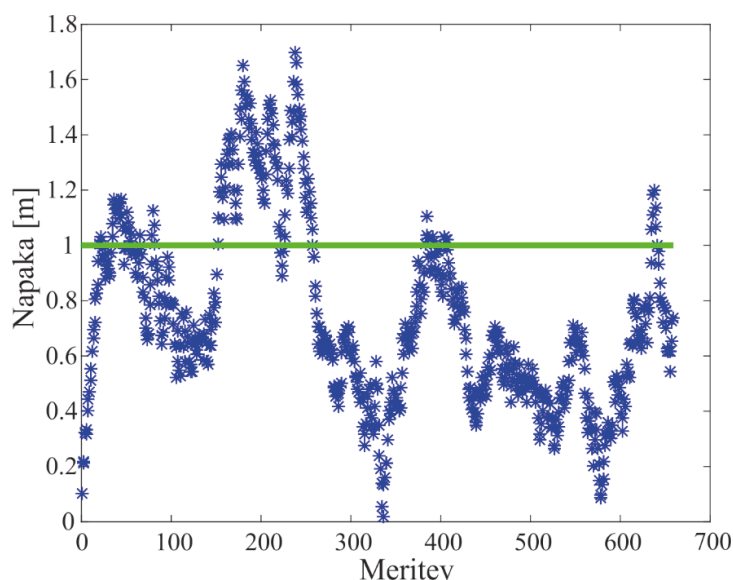
Pri izračunu trenutnih lokacij sprejemnika smo v postopku optimizacije izbrali začetni pogoj tako, da je bil začetni položaj (x_z, y_z) (točka, v kateri optimizacija prične z iskanjem rešitve) enak rešitvi optimizacije pri prejšnji meritvi jakosti signalov. Da bi dosegli efekt filtriranja na izračunanih položajih sprejemnika, smo uporabili omejeno optimizacijo. Za vsak nov izračun položaja smo uporabili omejitve območja, znotraj katerega se lahko nahaja izračunani položaj. Omejitve smo določili z dvema intervaloma, na katerima se mora nahajati x oz. y koordinata: $(x_z - 0,05) \leq x \leq (x_z + 0,05)$ in $(y_z - 0,05) \leq y \leq (y_z + 0,05)$. Na sliki 6.30 je z zeleno krivuljo prikazana pot, ki jo dobimo z združitvijo vseh točk lokalizacije sprejemnika. Opazimo lahko, da smo z omejeno optimizacijo uspešno dosegli efekt filtriranja, ki pripomore k precejšnji izboljšavi natančnosti lokalizacije.

Na sliki 6.31 so prikazane napake (razdalje med izračunanimi in dejanskimi položaji, ki so bili določeni s fuzijo vizualne odometrije in inercialnega navigacijskega sistema) pri določanju položaja s trilateracijo in omejeno nelinearno optimizacijo. Opazimo lahko, da je v tem primeru maksimalna napaka znatno manjša (okoli 1,7 m)

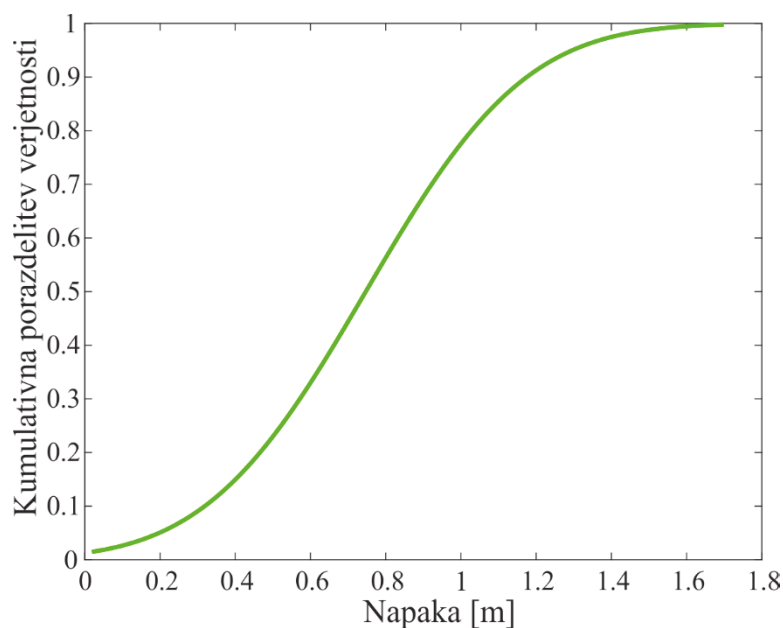
kot pri reševanju problema trilateracije z linearnim sistemom enačb in metodo linearnih najmanjših kvadratov. Na sliki 6.31 lahko vidimo tudi, da je večina napak manjših od 1 m (kar je naš cilj). Bolj natančne informacije lahko dobimo z grafom kumulativne porazdelitve verjetnosti napake, ki je prikazan na sliki 6.32. Iz njega lahko razberemo, da je pri 78 % ocenjenih položajev napaka manjša od 1 m.



Slika 6.30: Določanje položaja z uporabo trilateracije in omejene nelinearne optimizacije

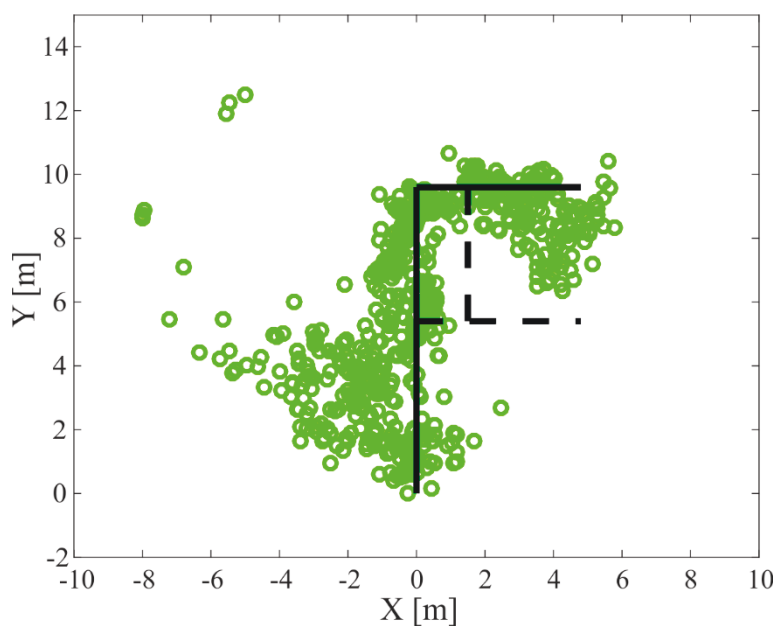


Slika 6.31: Napaka pri določanju položaja s trilateracijo in omejeno nelinearno optimizacijo

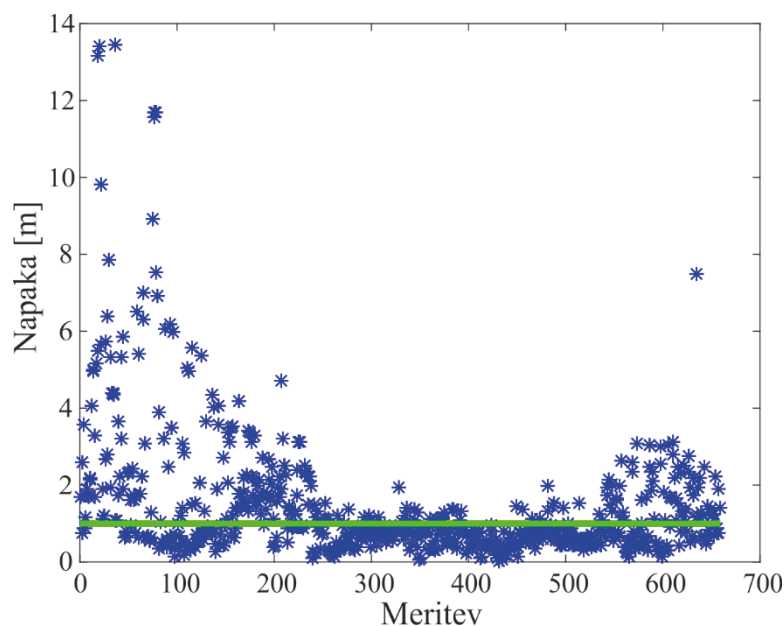


Slika 6.32: Natančnost lokalizacije, podana s kumulativno porazdelitvijo verjetnosti napake

Za primerjavo rezultatov smo na sliki 6.33 prikazali rezultate neomejene nelinearne optimizacije (ki temelji na metodi omejenega koraka). Opazimo lahko, da so v tem primeru rezultati lokalizacije veliko slabši kot pri omejeni nelinearni optimizaciji, saj so napake (slika 6.34) veliko večje (pri 60 % ocenjenih položajev je napaka večja od 1 m) in tudi maksimalna napaka je v tem primeru enaka približno 13,5 m.



Slika 6.33: Določanje položaja z uporabo trilateracije in (neomejene) nelinearne optimizacije



Slika 6.34: Napaka pri določanju položaja s trilateracijo in (neomejeno) nelinearno optimizacijo

Z uporabo omejene nelinearne optimizacije smo prišli že zelo blizu zastavljenemu cilju, tj. da bi bile vse napake manjše od 1 m. Vendar pa moramo pri preizkušanju različnih algoritmov vzeti v obzir še en zelo pomemben faktor, tj. računska zahtevnost (časovna potratnost) algoritma, ki je ključnega pomena v primeru, ko želimo, da se algoritem izvaja v realnem času na procesorsko omejeni napravi. In ravno tu se skriva največja slabost algoritma trilateracije, ki temelji na omejeni nelinearni optimizaciji. Pri preizkusu v okolju Matlab je ta algoritem (pri nastavljenem maksimalnem številu iteracij optimizacije z omejenim korakom na 50) porabil kar 350 krat več časa za izračun enega položaja kot pa algoritem triangulacije, ki temelji na linearnem sistemu enačb in linearni metodi najmanjših kvadratov (ta je potreboval 0,17 ms za eno meritev).

6.5 Lokalizacija z uporabo optimizacije z rojem delcev

6.5.1 Optimizacija z rojem delcev

Optimizacija z rojem delcev (ang. particle swarm optimization – PSO) je robustna stohastična optimizacijska metoda, ki rešuje dani problem tako, da v zaporednih iteracijah poskuša izboljšati kandidate za rešitev problema glede na izbrano kriterijsko funkcijo.

Metoda PSO optimizira dani problem z izbranimi kandidati oz. delci, ki sestavljajo roj (tj. populacijo potencialnih rešitev). S premikanjem teh delcev znotraj omejenega območja iskanja po preprostih matematičnih formulah (ki delcu določijo nov položaj in hitrost) pridemo do skupnega položaja, v katerem se zberejo vsi delci (vsi delci konvergirajo proti isti točki). Ta položaj predstavlja globalni optimum kriterijske funkcije. Na vsak naslednji premik delca vpliva njegov trenutno najboljši znani položaj ter trenutno najboljši znani položaj kateregakoli izmed delcev v območju iskanja rešitve. Za pričakovati je, da se bodo na ta način vsi delci znotraj roja pomikali proti skupni točki, ki predstavlja najboljšo rešitev optimizacijskega problema.

Metodo PSO so v svojem delu opisali Kennedy, Eberhart in Shi [221, 222, 223], njen prvotni namen pa je bil simuliranje socialnega obnašanja [224], kot je opis gibanja organizmov znotraj jate ptic ali rib.

Pristop z metodo PSO je zmogljiv metahevristični način reševanja kompleksnih optimizacijskih problemov, saj ne zahteva dodatnih predpostavk o problemu. Z ustrezno izbranim številom kandidatov (delcev), ki predstavljajo potencialno rešitev, lahko metoda PSO poišče optimum kriterijske funkcije tudi znotraj velikih prostorov. Vendar pa pri tej metodi zaradi metahevrističnega pristopa ni zagotovljeno, da bo optimalna rešitev sploh kdaj najdena. PSO namreč ne uporablja gradienta optimizacijskega problema, kar pomeni, da v tem primeru ni zahtevan pogoj, da je ta odvedljiv, kot je zahtevano v klasičnih optimizacijskih metodah (npr. metoda padajočega gradienta ali kvazi-Newtonova metoda).

Denimo, da je $f: \mathbb{R}^n \rightarrow \mathbb{R}$ kriterijska funkcija, ki jo želimo minimizirati. Cilj je najti rešitev $\mathbf{a}$, za katero velja $f(\mathbf{a}) < f(\mathbf{b})$ za vse kandidate oz. delce $\mathbf{b}$ v iskalnem območju. To pomeni, da $\mathbf{a}$ predstavlja globalni minimum. Če želimo poiskati rešitev, ki maksimizira dani problem, potem mora biti upoštevana kriterijska funkcija $h = -f$.

Naj bo S število vseh delcev v roju, pri čemer njihov položaj v iskalnem območju označimo z $\mathbf{x}_i \in \mathbb{R}^n$ in njihovo hitrost z $\mathbf{v}_i \in \mathbb{R}$. Naj bo $\mathbf{p}_i$ najboljši znani položaj delca i in $\mathbf{g}$ najboljši položaj celotnega roja. Število vseh delcev se določi glede na predlagane formule (npr. $S = 10 + 2\sqrt{n}$, kjer je n dimenzija iskalnega območja) ali eksperimentalno.

Pri metodi PSO [223] dobimo nov položaj delca $\mathbf{x}_i^{k+1}$ z modelom gibanja delcev (slika 6.35) tako, da se upošteva njegov trenutni položaj $\mathbf{x}_i^k$, njegova trenutna hitrost

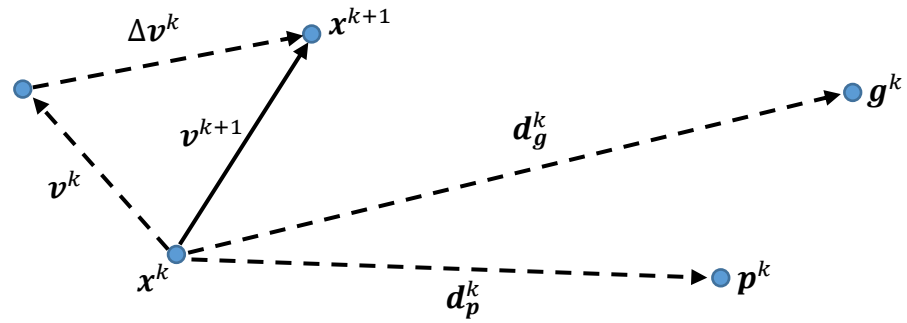
$\mathbf{v}_i^k$, razdalja $\mathbf{d}_{p_i}^k$ med trenutnim položajem delca $\mathbf{x}_i^k$ in najboljšim položajem delca $\mathbf{p}_i^k$ ter razdalja $\mathbf{d}_{g_i}^k$ med trenutnim položajem delca $\mathbf{x}_i^k$ in najboljšim položajem vseh delcev roja v danem trenutku $\mathbf{g}^k$:

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1}, \quad (6.105)$$

kjer je:

$$\mathbf{v}_i^{k+1} = \omega \mathbf{v}_i^k + \Delta \mathbf{v}_i^k, \quad (6.106)$$

$$\Delta \mathbf{v}_i^k = \varphi_p r_p (\mathbf{p}_i^k - \mathbf{x}_i^k) + \varphi_g r_g (\mathbf{g}^k - \mathbf{x}_i^k) = \varphi_p r_p \mathbf{d}_{p_i}^k + \varphi_g r_g \mathbf{d}_{g_i}^k. \quad (6.107)$$



Slika 6.35: Model gibanja delcev

Parameter ω predstavlja inercialno utež [223], ki določa, v kakšni meri naj delec pomni svojo prejšnjo smer (velike vrednosti parametra omogočajo hitrejšo konvergenco, majhne vrednosti pa bolj natančno lokalno raziskovanje), parameter φ_p predstavlja kognitivni učni faktor in parameter φ_g predstavlja družbeni učni faktor. r_p in r_g sta naključni števili, ki ležita na intervalu med 0 in 1.

Osnovni algoritem PSO [225] lahko strnemo v naslednje točke:

- Za vsak delec $i = 1, \dots, S$:
 - Inicializiraj začetne položaje delcev tako, da so enakomerno porazdeljeni v iskalnem območju: $\mathbf{x}_i \sim U(b_{lo}, b_{up})$, kjer sta b_{lo} in b_{up} spodnja oz. zgornja meja iskalnega območja.
 - Inicializiraj najboljši znani položaj delca: $\mathbf{p}_i \leftarrow \mathbf{x}_i$.
 - Če je $f(\mathbf{p}_i) < f(\mathbf{g})$, posodobi najboljši znani položaj roja: $\mathbf{g} \leftarrow \mathbf{p}_i$.
 - Inicializiraj hitrosti delcev: $\mathbf{v}_i \sim U(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|)$.

- Ponavljaj, dokler ni izpolnjen kriterij za konec iskanja rešitve (npr. da je število iteracij preseglo določeno mejo, ali pa je bila najdena rešitev, pri kateri je kriterijska funkcija manjša od določene meje).
 - o Za vsak delec $i = 1, \dots, S$:
 - Za vsako dimenzijo $d = 1, \dots, n$:
 - Izberi naključni števili $r_p, r_g \sim U(0,1)$.
 - Ponovno izračunaj hitrost delca:

$$\mathbf{v}_{i,d} \leftarrow \omega \mathbf{v}_{i,d} + \varphi_p r_p (\mathbf{p}_{i,d} - \mathbf{x}_{i,d}) + \varphi_g r_g (\mathbf{g}_d - \mathbf{x}_{i,d})$$
 - Ponovno izračunaj položaj delca: $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$.
 - Če je $f(\mathbf{p}_i) < f(\mathbf{g})$, posodobi najboljši položaj roja: $\mathbf{g} \leftarrow \mathbf{p}_i$.
- Končna vrednost $\mathbf{g}$ predstavlja najboljšo najdeno rešitev minimizacijskega problema.

Parametra r_p in r_g predstavljata naključni pospešek v smeri najboljšega položaja delca $\mathbf{p}_i$ in v smeri najboljšega položaja celotnega roja $\mathbf{g}$. Učinkovitost metode PSO je močno odvisna od izbire parametrov ω , φ_p in φ_g , ki morajo biti ustrezno prilagojeni danemu optimizacijskemu problemu [226-234]. Pri izbiri parametrov je potrebno največ pozornosti posvetiti parametru ω , ki ima največji vpliv na učinkovitost (hitrost konvergence) PSO metode. Zato je bil ta parameter tudi predmet številnih študij [226-234], v katerih so avtorji predlagali različne (optimalne) načine, kako naj bi se parameter ω spreminjal (linearno ali nelinearno s številom iteracij ali glede na izboljšanje delcev v posamezni iteraciji) tudi med samo optimizacijo. Parametri metode PSO so lahko pridobljeni tudi z dodatno optimizacijo, ki jo imenujemo »metaoptimizacija« [235-237].

Pri iskanju globalnega minimuma z osnovno metodo PSO lahko kaj hitro dobimo rešitev, ki je lokalni (in ne globalni) minimum funkcije. Temu se lahko izognemo tako, da pri izračunu premika posameznega delca ne upoštevamo najboljšega položaja delcev iz celotnega roja, ampak najboljši položaj sosednjih (npr. m najbližjih) delcev. V tem primeru je metoda PSO lokalno optimalna (ang. local best), konvergenca h globalnemu optimumu pa se upočasni. Konvergenca je v veliki meri odvisna od izbire parametrov metode PSO, ki morajo biti določeni tako, da vsi delci roja konvergirajo proti isti točki [230-232].

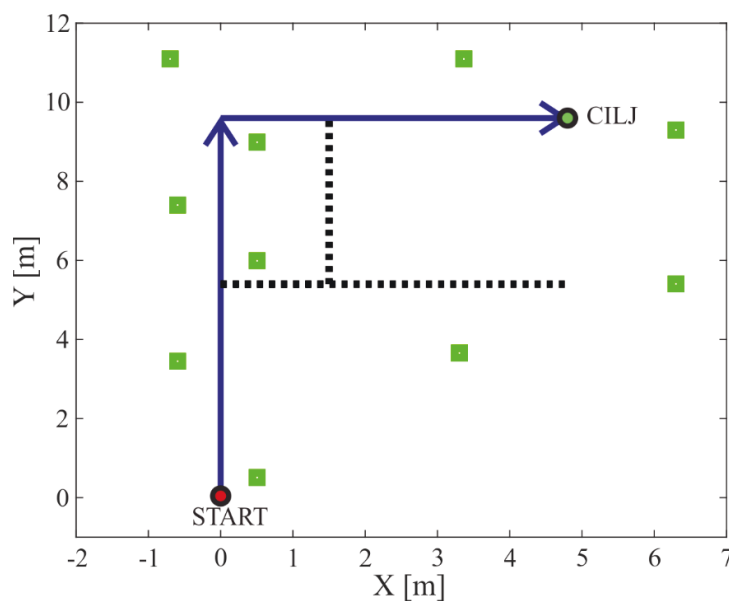
Algoritmi PSO se pojavljajo v različnih izvedbah [229], ki npr. na različne načine inicializirajo delce in njihove hitrosti, spreminjajo hitrosti, različno upoštevajo interakcije med delci itd. Da bi se metoda PSO uveljavila na različnih znanstvenih področjih za namen reševanja optimizacijskih problemov, so vodilni raziskovalci na tem področju definirali standardno obliko algoritma [238, 239]. Novejši in bolj sofisticirani algoritmi PSO se uporabljajo v hibridnem načinu z drugimi optimizacijskimi algoritmi (npr. genetskimi algoritmi) [240-242].

Algoritem PSO ima naslednje prednosti: enostavna implementacija, enostavna paralelizacija, odvodi niso potrebni, število parametrov algoritma je majhno. Slabost algoritma PSO pa je počasna konvergenca v okolici optimuma.

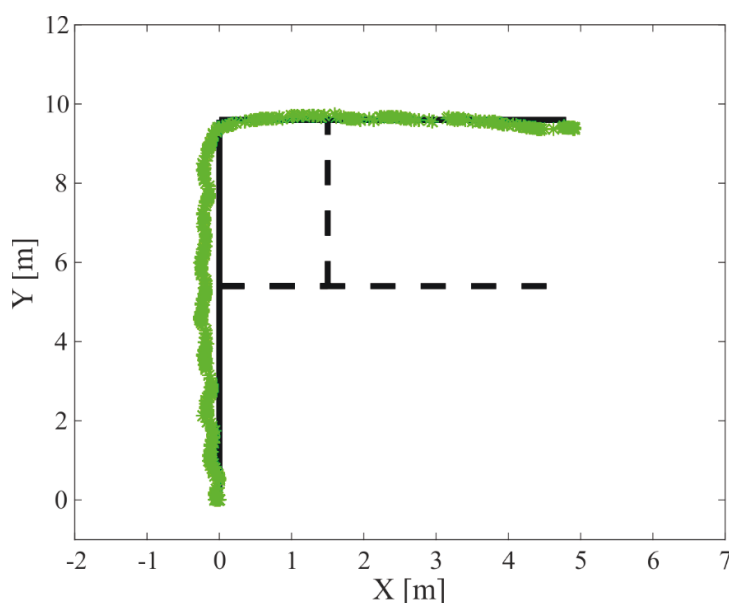
6.5.2 Rezultati lokalizacije na osnovi optimizacije z rojem delcev

Cilj lokalizacije na osnovi izmerjenih jakosti Bluetooth signalov in optimizacije z rojem delcev je v iskanju trenutnega položaja sprejemnika (x, y) v dvodimenzionalnem prostoru, pri katerem na podlagi modelov jakosti signalov (6.2) dobimo vektor $\mathbf{RSSI}_p = f_{\text{RSSI}}(\mathbf{d})$, ki je glede na kriterijsko funkcijo $f_k = \sum_{i=1}^m |\text{RSSI}_{m_i} - \text{RSSI}_{p_i}|$ (za $i = 1, \dots, 5$) najbolj podoben trenutnemu vektorju meritev jakosti signalov iz petih različnih oddajnikov $\mathbf{RSSI}_m$. Vektor razdalj $\mathbf{d} = [d_1, \dots, d_i]^T$ (za $i = 1, \dots, 5$) je sestavljen iz evklidskih razdalj $d_i = \sqrt{(x - x_{bi})^2 + (y - y_{bi})^2}$ med trenutnim položajem (x, y) sprejemnika in položajem vseh Bluetooth oddajnikov (x_{bi}, y_{bi}) . Denimo, da v nekem trenutku sprejemnik izmeri jakosti signalov iz petih oddajnikov, iz katerih sestavimo vektor $\mathbf{RSSI}_m$. Trenutni položaj Bluetooth sprejemnika lahko nato določimo z uporabo optimizacije z rojem delcev tako, da najprej izberemo področje, znotraj katerega naj bi se trenutno nahajal sprejemnik in nato na tem področju enakomerno razporedimo izbrano število delcev. V zaporednih iteracijah te delce premikamo v skladu z metodo PSO in preverjamo, kako pripadajoči vektorji $\mathbf{RSSI}_p$ ustrezajo vektorju izmerjenih jakosti signalov $\mathbf{RSSI}_m$. Položaj v prostoru (x, y) , v katerem se zberejo vsi delci metode PSO, predstavlja rešitev optimizacije, saj pripadajoče jakosti signalov RSSI_{p_i} (dobljene z uporabo modelov $\text{RSSI}_{p_i} = f_{\text{RSSI}_i}(d_i)$ za $i = 1, \dots, 5$) minimizirajo kriterijsko funkcijo.

Za preizkus lokalizacijskega algoritma, ki temelji na metodi PSO, so bile s pametnim telefonom Android zajete meritve jakosti signalov za celotno testno pot (od začetne točke »START« do končne točke »CILJ«), ki je prikazana na sliki 6.36. Na sliki 6.37 je prikazana dejanska pot, ki je bila določena z uporabo vizualne odometrije in inercialnega navigacijskega sistema. Za določanje trenutnega položaja v prostoru se upošteva zadnjih pet meritev iz različnih Bluetooth oddajnikov.



Slika 6.36: Položaj Bluetooth oddajnikov v prostoru

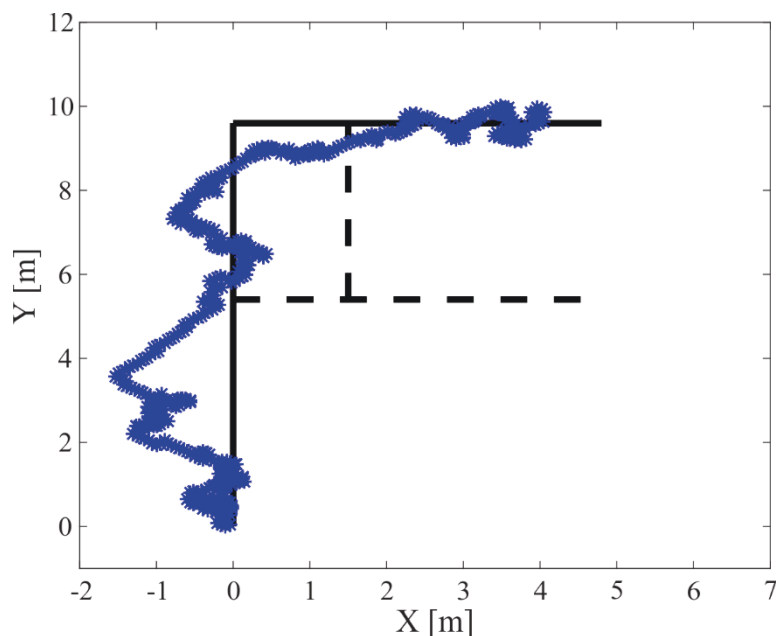


Slika 6.37: Dejanska pot, ki je bila določena z vizualno odometrijo in inercialnim navigacijskim sistemom

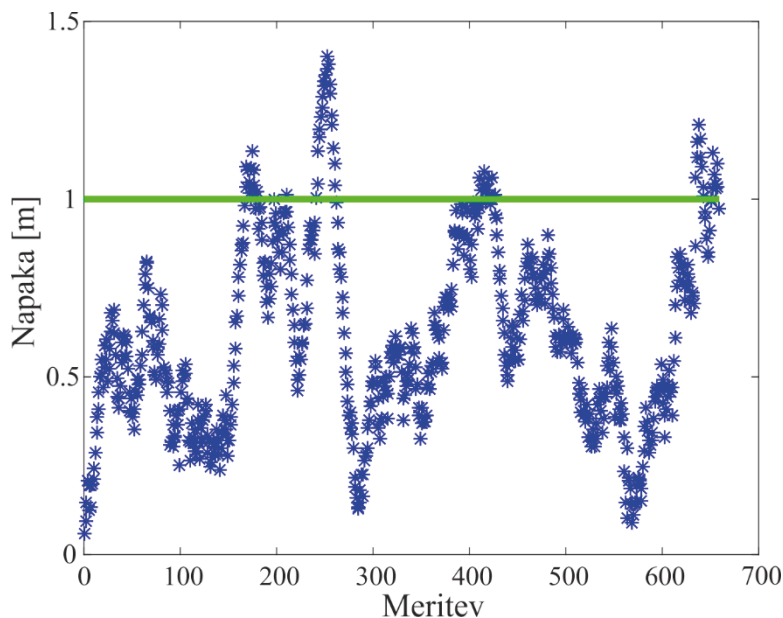
Lokalizacijski algoritem je bil implementiran v okolju Matlab, kjer smo izvedli tudi preizkus na realnih podatkih. Pri preizkusu algoritma smo predpostavili, da poznamo približen začetni položaj sprejemnika v prostoru (da lahko določimo območje iskanja pri metodi PSO), ki ga lahko določimo z neko hitrejšo in manj natančno metodo (npr. s trilateracijo in metodo linearnih najmanjših kvadratov). Namreč z metodo PSO bi potrebovali veliko več časa, da bi določili začetni položaj v prostoru, saj bi morali delce porazdeliti po celotnem prostoru, število delcev bi moralo biti ustrezno veliko in tudi število iteracij bi moralo biti večje, da bi vsi delci skonvergirali v isto točko. Za vsako naslednjo meritev jakosti signalov smo delce metode PSO razporedili okoli trenutno določenega položaja. Iskalno območje, znotraj katerega so bili razporejeni delci, je bilo kvadratne oblike in velikosti $0,1 \times 0,1$ m. Velikost območja je bila določena glede na predpostavko o povprečni hitrosti pešca (1,5 m/s) in frekvenco meritev jakosti signalov (30 ms). Z omejevanjem območja iskanja pri metodi PSO pa dosežemo tudi učinek filtriranja oz. glajenja poti, saj vsak naslednji izračunani položaj sprejemnika ne more biti oddaljen od prejšnjega položaja za več kot $\frac{\sqrt{2}}{2}a$ (kjer je a dolžina stranice kvadrata, ki določa območje iskanja). Optimalne parametre metode PSO $\omega = 0,05$, $\varphi_p = 0,1$ in $\varphi_g = 0,1$ smo izbrali eksperimentalno. Da bi lahko lokalizacijski algoritem deloval kar se da hitro in pri tem dosegal optimalne rezultate tudi na pametnem telefonu, smo glede na velikost iskalnega območja določili, da se pri metodi PSO uporabljajo le 4 delci (saj večje število delcev ne izboljša rezultata, temveč le poveča računsko zahtevnost), število iteracij pa smo omejili na 10. Na sliki 6.38 so prikazani rezultati lokalizacije z uporabo metode PSO. Lokalizacijski algoritem je v okolju Matlab potreboval okoli 3 ms za izračun enega položaja v prostoru.

Na sliki 6.39 so prikazane napake (razdalje med izračunanimi in dejanskimi položaji, ki so bili določeni s fuzijo vizualne odometrije in inercialnega navigacijskega sistema) pri določanju položaja z optimizacijsko metodo PSO in pri uporabi nelinearnih modelov. Maksimalna napaka je v tem primeru manjša od 1,4 m, kar je precej boljši rezultat kot pri določanju položaja s trilateracijo in omejeno nelinearno optimizacijo (tam je bila maksimalna napaka 1,7 m). Tudi pri primerjavi računske zahtevnosti se ta algoritem bolje izkaže, saj porabi kar 20-krat manj časa (v okolju

Matlab) za izračun enega položaja (3 ms) kot pa algoritem, ki temelji na trilateraciji in omejeni nelinearni optimizaciji.



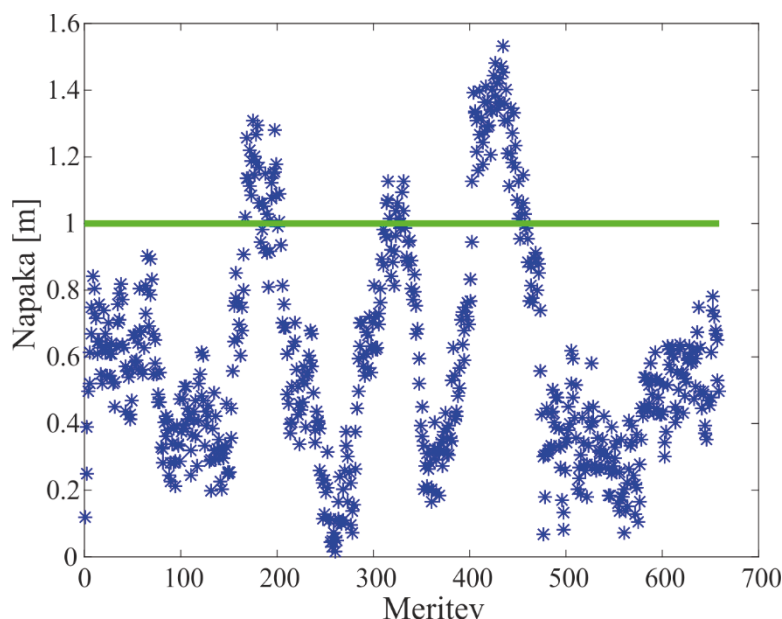
Slika 6.38: Določanje položaja z uporabo nelinearnih modelov in optimizacijske metode PSO



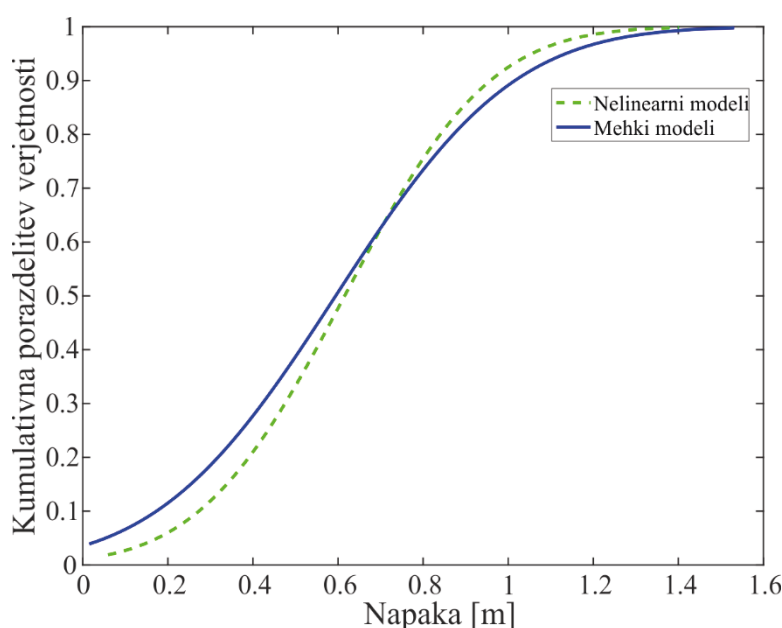
Slika 6.39: Napake pri določanju položaja z metodo PSO (nelinearni modeli)

V okolju Matlab smo preizkusili tudi delovanje optimizacijske metode PSO v kombinaciji z mehкими identifikacijskimi modeli, ki bolj natančno opisujejo širjenje

Da bi lažje opredelili natančnost lokalizacijskega algoritma, ki temelji na metodi PSO, smo na sliki 6.42 prikazali graf kumulativne porazdelitve verjetnosti napake. Iz grafa je mogoče razbrati, da tako pri uporabi nelinearnih modelov, kot pri uporabi mehkih modelov dobimo veliko boljše rezultate lokalizacije kot pri algoritmu, ki temelji na trilateraciji in omejeni nelinearni optimizaciji. Namreč, v obeh primerih je za okoli 90 % ocenjenih položajev napaka manjša od 1 m. Pri trilateraciji je bila napaka manjša od 1 m le v 78 % ocen položajev.



Slika 6.41: Napake pri določanju položaja z metodo PSO (mehki modeli)

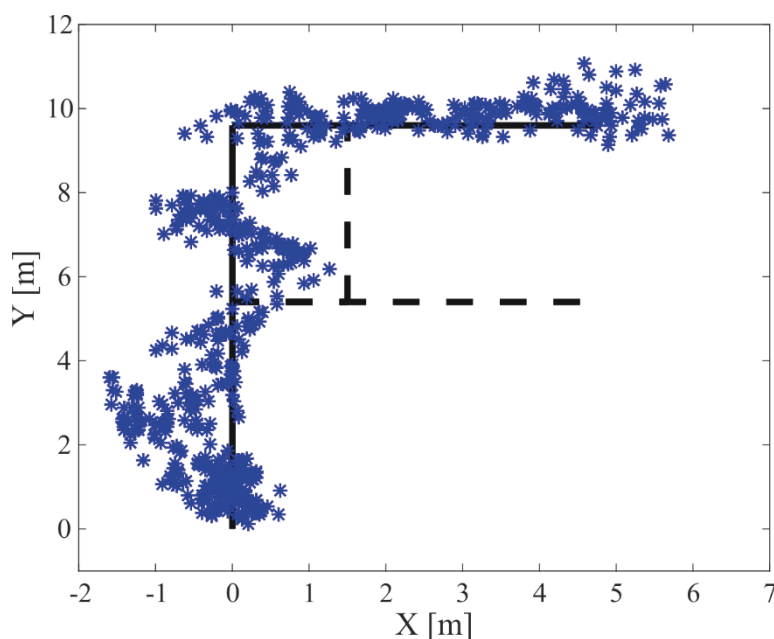


Slika 6.42: Natančnost lokalizacije, podana s kumulativno porazdelitvijo verjetnosti napake pri uporabi nelinearnih oz. mehkih modelov

Na sliki 6.42 lahko opazimo, da ima lokalizacijski algoritem, ki temelji na metodi PSO, pri uporabi mehkih modelov in izbranem iskalnem območju velikosti $0,1 \times 0,1$ m nekoliko več težav v primeru, ko se napaka pri določanju položaja prične povečevati, ker jakosti signalov vsebujejo več šuma (šum, ki je prisoten v meritvah jakosti signalov, povzroči napačno določanje položaja in ker je vsako naslednje območje iskanja vezano na prejšnji položaj, metoda PSO išče rešitev vse bolj stran od

dejanskega položaja) in ker pri nekaterih mehkih modelih jakost signala ne pada monotono z oddaljenostjo od oddajnika. Zaradi slednjega lahko metoda PSO znotraj omejenega območja iskanja (ki lahko več ne zajema dejanskega položaja) najde vektor jakosti signalov (ki je najbolj podoben vektorju meritev), ki mu pripada položaj, ki ni najbližje dejanskemu položaju (iskalno območje se še bolj odmakne od dejanskega položaja). Zato se iskalno območje počasneje vrne nazaj na del prostora, kjer ponovno vključuje (pokriva) dejanski položaj.

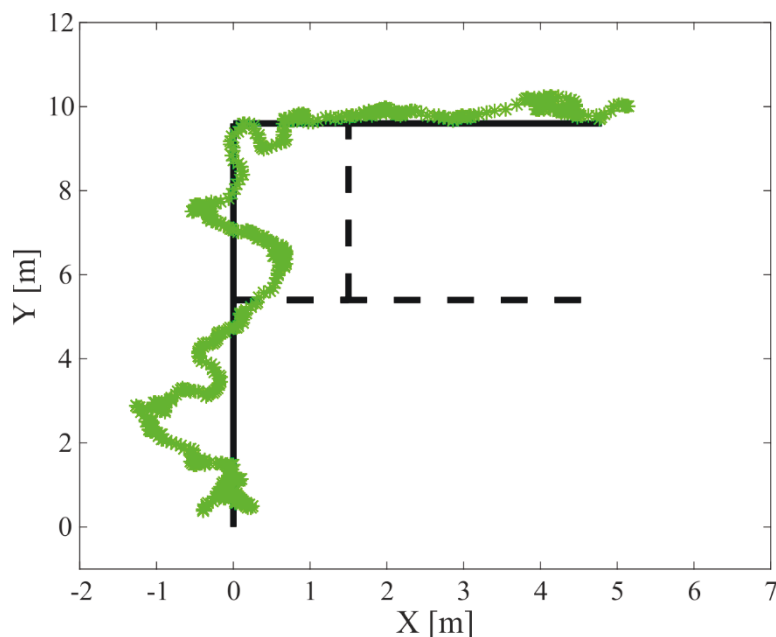
Zato smo postopek lokalizacije z uporabo metode PSO in mehkih modelov ponovili, s tem da smo povečali območje iskanja na $0,8 \times 0,8$ m. S tem smo sicer zmanjšali učinek filtriranja, vendar smo metodi PSO omogočili hitrejši prehod iz napačnega območja iskanja (na katerega je metoda zašla zaradi prisotnosti šuma v meritvah jakosti signalov) na pravilno območje iskanja. Na sliki 6.43 lahko vidimo, da so dobljeni položaji v tem primeru bolj enakomerno razporejeni okoli referenčne poti.



Slika 6.43: Določanje položaja z uporabo mehkih modelov in optimizacijske metode PSO (velikost iskalnega območja: $0,8 \times 0,8$ m)

Ker so dobljeni položaji na sliki 6.43 zelo razpršeni, si želimo skoke med sosednjimi točkami odpraviti. To lahko storimo tako, da uporabimo dodatni nizkopasovni filter za obe koordinati točk: $x_{LP}(k) = 0,9x_{LP}(k-1) + 0,1x(k)$ in

$y_{LP}(k) = 0,9y_{LP}(k-1) + 0,1y(k)$, kjer je $(x(k), y(k))$ trenutni položaj, ki ga dobimo z metodo PSO. $(x_{LP}(k), y_{LP}(k))$ in $(x_{LP}(k-1), y_{LP}(k-1))$ pa sta trenutni in prejšnji filtriran položaj. Z dodatnim (nizkopasovnim) filtriranjem dobimo pot, ki je veliko bolj gladka in se veliko bolje prilega referenčni poti (slika 6.44). Dodan nizkopasovni filter se v smislu računske obremenitve praktično ne pozna.

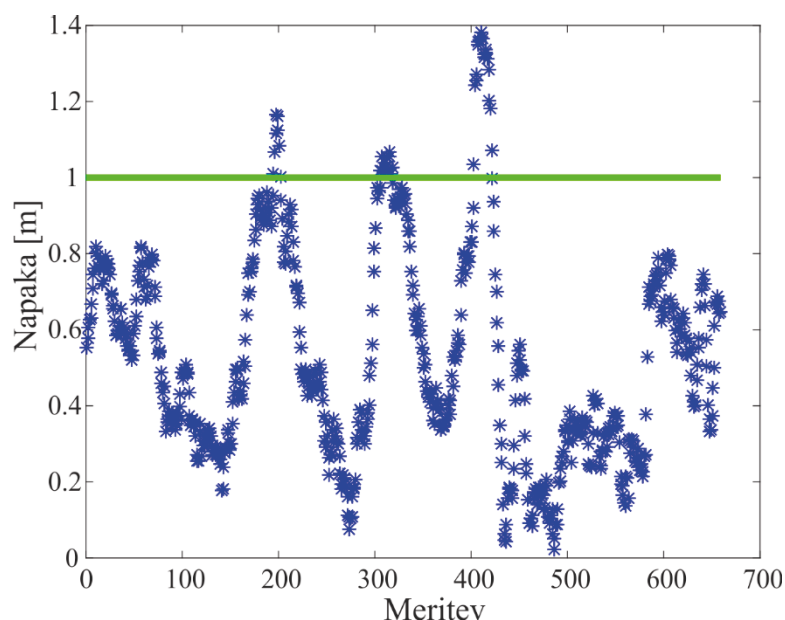


Slika 6.44: Glajenje poti (ki je sestavljena iz položajev, dobljenih z uporabo mehkih modelov in metode PSO) z uporabo nizkopasovnega filtra

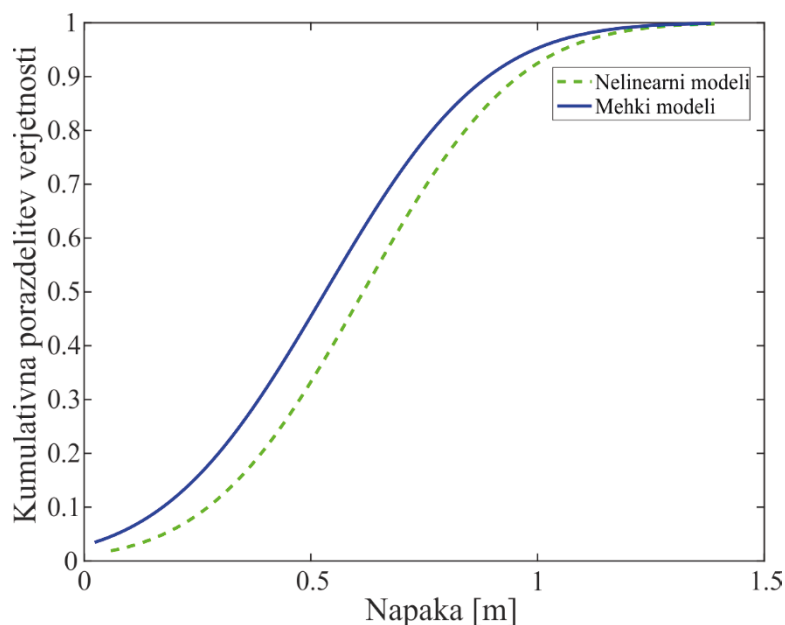
Na sliki 6.45 so prikazane napake pri določanju položaja z metodo PSO (in upoštevanju mehkih modelov) v primeru, ko je uporabljen nizkopasovni filter za glajenje poti. Opazimo lahko, da z uporabo nizkopasovnega filtra dosežemo veliko boljši rezultat pri določanju položaja, kot če prekomerno omejujemo iskalno območje pri metodi PSO. Vendar je to »dodatno« filtriranje smiselno le pri uporabi mehkih modelov. Namreč, pri nelinearnih modelih uporaba nizkopasovnega filtra ni potrebna, saj že z omejevanjem iskalnega področja dosežemo enak efekt oz. če bi povečali iskalno območje (in s tem zmanjšali učinek filtriranja) in nato dodatno uporabili nizkopasovni filter, rezultati lokalizacije ne bi bili nič boljši (opravljen je bil test).

Na sliki 6.46, kjer je prikazana kumulativna porazdelitev verjetnosti napake pri določanju položaja z metodo PSO, lahko vidimo, da z dodatnim (nizkopasovnim) filtriranjem izkoristimo potencial bolj natančnih mehkih modelov, saj je na ta način

kar pri 95 % ocenjenih položajev napaka manjša od 1 m. Pri uporabi nelinearnih modelov pa je napaka manjša od 1 m pri 92 % ocen. Graf na sliki 6.46 potrjuje, da so napake pri uporabi mehkih modelov manjše kot pri uporabi nelinearnih modelov, saj se modra krivulja čez celotno območje napak nahaja nad zeleno.



Slika 6.45: Napake pri določanju položaja z metodo PSO (mehki modeli) v primeru, ko je za glajenje poti uporabljen nizkopasovni filter

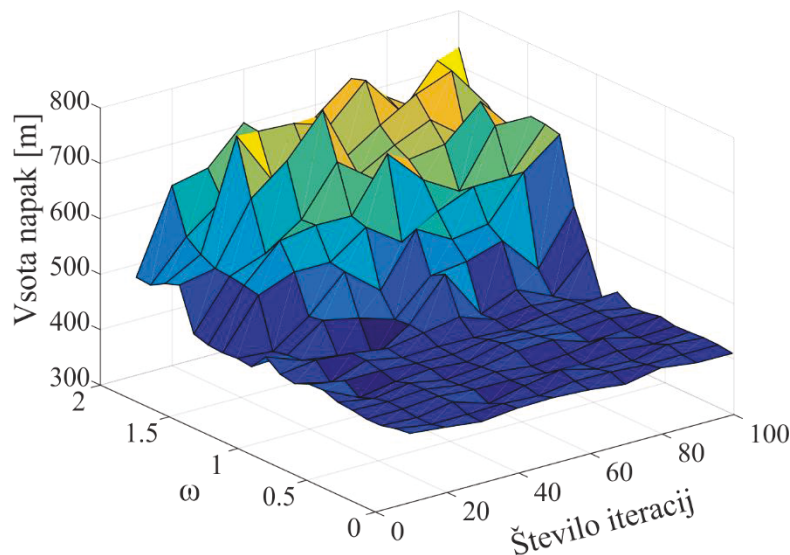


Slika 6.46: Natančnost lokalizacije, podana s kumulativno porazdelitvijo verjetnosti napake pri uporabi nelinearnih oz. mehkih modelov in metode PSO. V kombinaciji z mehki modeli je bil uporabljen nizkopasovni filter za glajenje poti.

6.5.3 Vpliv parametrov metode PSO na natančnost lokalizacije

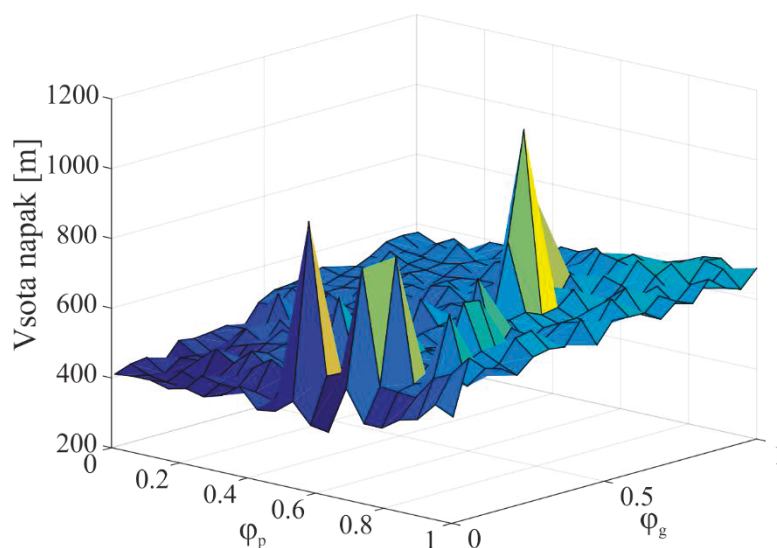
Z namenom, da bi se prepričali, ali so izbrani parametri metode PSO ustrezni, smo izvedli več simulacij v okolju Matlab in opazovali, kako spreminjanje parametrov metode PSO vpliva na natančnost lokalizacije. Za kriterij, po katerem smo primerjali rezultate simulacij, smo izbrali vsoto evklidskih razdalj med izračunanimi (x_i, y_i) in dejanskimi položaji (x_{ti}, y_{ti}) sprejemnika: $S_e = \sum_{i=1}^p \sqrt{(x_i - x_{ti})^2 + (y_i - y_{ti})^2}$, kjer je p število vseh meritev na poti.

Pri prvi simulaciji smo spreminjali število iteracij od 10 do 50 in inercijsko utež ω od 0,01 do 2. Ostali parametri metode PSO so bili konstantni: $\varphi_p = 0,1$ in $\varphi_g = 0,1$, število delcev = 4 in velikost iskalnega območja: $0,1 \times 0,1$ m. Na sliki 6.47, kjer so prikazani rezultati te simulacije, lahko opazimo, da spreminjanje maksimalnega števila iteracij (od 10 naprej) praktično ne vpliva na natančnost lokalizacije. 10 iteracij predstavlja minimalno mejo, pri kateri dosežemo zadovoljive rezultate PSO optimizacije v dovolj kratkem času. Za parameter ω lahko iz slike 6.47 ugotovimo, da so vrednosti od 0,01 do 1 ustrezne, saj dobimo približno enako dobre rezultate lokalizacije. Če je parameter ω večji od 1, na gibanje posameznega delca v večji meri vplivajo njegove predhodne hitrosti, kar poslabša rezultat lokalizacije.



Slika 6.47: Vpliv maksimalnega števila iteracij in parametra ω (metode PSO) na natančnost lokalizacije

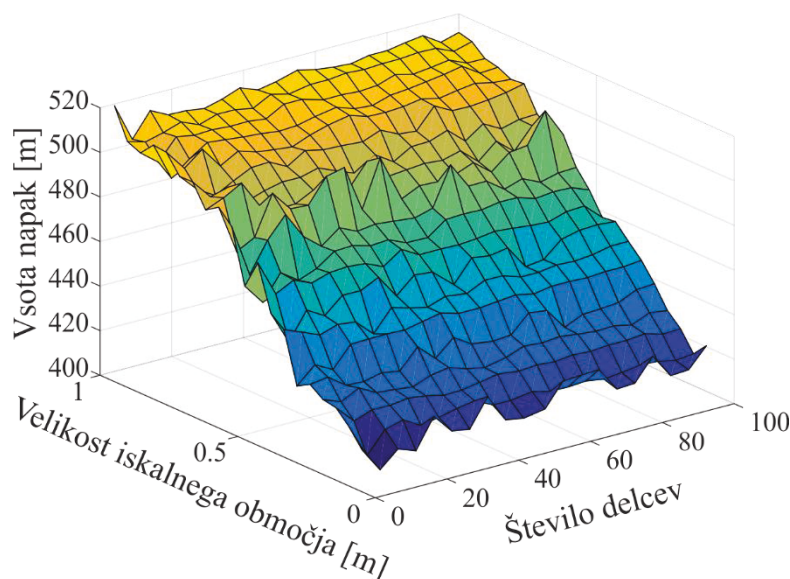
Na sliki 6.48 je prikazan vpliv parametrov φ_p in φ_g (metode PSO) na natančnost lokalizacije. V simulaciji smo parametra spreminjali na območju od 0,005 do 1. Vrednosti ostalih parametrov so bile naslednje: število iteracij = 50, $\omega = 1$, število delcev = 4 in velikost iskalnega območja = $0,1 \times 0,1$ m. Iz slike 6.48 lahko ugotovimo, da so ustrezne vrednosti parametrov φ_p in φ_g naslednje: $\varphi_p \leq 0,1$ in $\varphi_g \leq 0,1$. Namreč, pri večjih vrednostih parametrov je hitrost delcev prevelika (delci delajo prevelike skoke in se ne umirijo v globalnem minimumu) in posledično dobimo slabše rezultate lokalizacije. Posamezni skoki vsote napak, ki jih lahko opazimo na sliki 6.48, so ravno posledica divergence delcev zaradi prevelikih hitrosti.



Slika 6.48: Vpliv parametrov φ_p in φ_g (metode PSO) na natančnost lokalizacije

Pri zadnji simulaciji, katere rezultat je prikazan na sliki 6.49, smo spreminjali velikost iskalnega območja od $0,1 \times 0,1$ m do 1×1 m in pa število delcev PSO metode od 4 do 100. Ostali parametri so bili konstantni: število iteracij = 50, $\omega = 1$, $\varphi_p = 0,1$ in $\varphi_g = 0,1$. Na sliki 6.49 lahko opazimo, da število delcev metode PSO v našem primeru praktično ne vpliva na natančnost lokalizacije, saj lahko pri dovolj velikem številu iteracij (npr. 50) metoda PSO z le nekaj delci najde globalni minimum kriterijske funkcije. Pri povečevanju iskalnega območja od $0,1 \times 0,1$ m do 1×1 m pa se natančnost lokalizacije zelo hitro zmanjšuje (slika 6.49). Iskanje globalnega minimuma na večjem področju je problematično iz več vidikov. Prvi problem je ta, da metoda PSO lahko hitreje zaide v lokalni minimum in tako ne najde globalnega

minimuma. Drugi problem je ta, da pri večjem iskalnem področju izgubimo učinek filtriranja, saj lahko metoda PSO vrne (zaradi napačnih meritev jakosti signalov) trenutno izračunan položaj oddajnika, ki je bolj oddaljen od prejšnjega položaja, kot v primeru majhnega iskalnega območja. Torej je v primeru, ko ne uporabljamo dodatnega nizkopasovnega filtriranja, smiselno izbrati majhno iskalno območje ($0,1 \times 0,1$ m) in majhno število delcev.

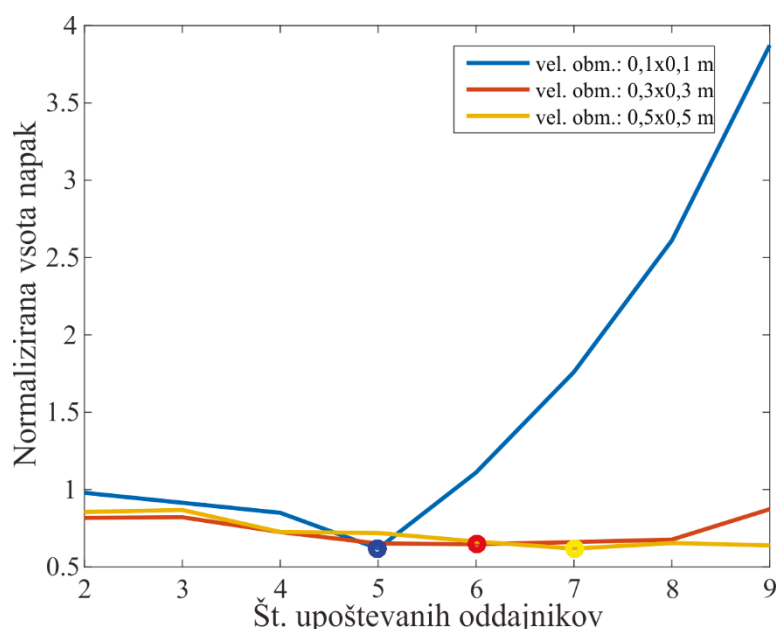


Slika 6.49: Vpliv velikosti iskalnega območja in števila delcev (metode PSO) na natančnost lokalizacije

Vpliv števila upoštevanih oddajnikov

Pri testih algoritma PSO smo že omenjali »optimalno« število oddajnikov, ki so potrebni za doseg natančne lokalizacije. Da bi bolj natančno preverili vpliv števila (hkrati) uporabljenih oddajnikov na natančnost lokalizacije, smo izvedli simulacijo z uporabo realnih podatkov. Posamezne položaje smo določili z metodo PSO, pri kateri smo izbrali naslednje parametre: število iteracij = 50, $\omega = 1$, $\varphi_p = 0,1$, $\varphi_g = 0,1$, število delcev = 4. Ker so v primeru, ko upoštevamo večje število oddajnikov, razdalje med točkami meritev jakosti signalov večje, moramo pri metodi PSO povečati tudi velikost iskalnega območja, saj je v nasprotnem primeru učinek filtriranja prevelik. Zato smo simulacijo trikrat ponovili pri različnih velikostih iskalnega območja metode PSO. Na sliki 6.50 lahko vidimo tri krivulje za različne velikosti iskalnega območja,

ki prikazujejo normalizirano vsoto napak (za celotno pot na sliki 6.37) glede na število upoštevanih oddajnikov. Vsoto napak je potrebno normalizirati glede na število vseh meritev, saj imamo pri upoštevanju 9 oddajnikov znatno manj meritev kot v primeru, ko upoštevamo le 2 oddajnika. Na sliki 6.50 lahko opazimo, da so minimumi vseh treh krivulj (označeni s krogi) povsem skupaj. Najnižjo vrednost normalizirane vsote napak dobimo pri 5 upoštevanih oddajnikih in velikosti iskalnega območja $0,1 \times 0,1$ m. Pri upoštevanju manj kot 5 oddajnikov so v vseh treh primerih napake večje. Glede na potek rumene in rdeče krivulje (slika 6.50) lahko zaključimo, da ni smiselno upoštevati več kot 5 oddajnikov hkrati, saj se natančnost lokalizacije ne izboljša.



Slika 6.50: Normalizirana vsota napak (pri določanju položaja) v odvisnosti od števila upoštevanih oddajnikov pri treh različnih velikostih iskalnega območja metode PSO

6.5.4 Vpliv šuma na natančnost lokalizacije

Pri merjenju jakosti Bluetooth signalov je prisoten šum, ki povzroča velika odstopanja jakosti signalov od prave vrednosti na določeni oddaljenosti od oddajnika. V poglavju 6.2, ki obravnava izgradnjo modelov jakosti signalov za vse Bluetooth oddajnike, smo predpostavili, da imajo residui (odstopanja od ocenjenih vrednosti) pri posameznih modelih normalno porazdelitev z neko pripadajočo standardno deviacijo σ_i . Da bi

preverili, kako velikost residuov vpliva na natančnost lokalizacije, smo izvedli simulacijo, v kateri smo za tri točke ($T_1(x_1, y_1), T_2(x_2, y_2), T_3(x_3, y_3)$) v prostoru določili vse možne položaje, ki bi jih lahko dobili pri različnih kombinacijah maksimalnih odstopanj jakosti signalov. V prvem primeru smo maksimalna odstopanja jakosti signalov določili kot:

$$RSSI_i^{MAX} = f_{RSSI_i}(d_i) \pm 2\sigma_i \text{ za } i = 1, \dots, 5,$$

v drugem primeru pa kot:

$$RSSI_i^{MAX} = f_{RSSI_i}(d_i) \pm 3\sigma_i \text{ za } i = 1, \dots, 5,$$

kjer so d_i razdalje med trenutno izbrano točko v prostoru in i -tim oddajnikom. f_{RSSI_i} predstavlja model jakosti signala i -tega oddajnika, σ_i pa predstavlja standardni odklon normalne porazdelitve residuov pri izbranem modelu i -tega oddajnika. Ker je lahko jakost Bluetooth signala iz i -tega oddajnika v izbrani točki prostora enaka $f_{RSSI_i}(d_i) + 2\sigma_i$ ali $f_{RSSI_i}(d_i) - 2\sigma_i$, imamo potemtakem 2^m (kjer je v našem primeru $m = 5$) vseh kombinacij oz. vektorjev jakosti signalov $\mathbf{RSSI}^{MAX}$. V prvem primeru je znotraj intervala ($f_{RSSI_i}(d_i) \pm 2\sigma_i$) upoštevanih 95,44 % vseh možnih residuov, v drugem primeru ($f_{RSSI_i}(d_i) \pm 3\sigma_i$) pa 99,74 % vseh možnih residuov.

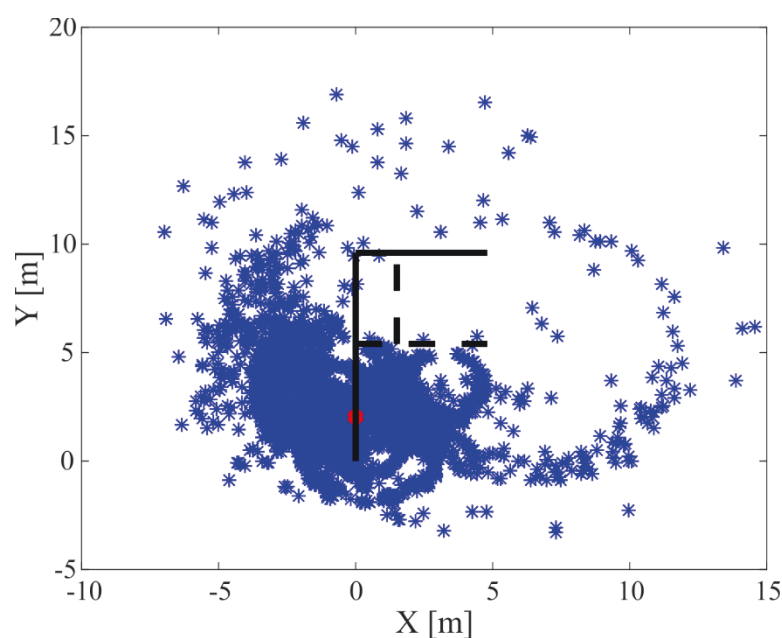
Ker jakost Bluetooth signala pri večjih razdaljah vse počasneje pada, je tam pri enakem odstopanju jakosti signala ($f_{RSSI_i}(d_i) \pm 2\sigma_i$) napaka pri določanju oddaljenosti od oddajnika večja. Zato smo pri simulaciji upoštevali tudi vse možne kombinacije izbora oddajnikov v prostoru, ki so različno oddaljeni od obravnavane točke v prostoru. V našem primeru so hkrati upoštevane meritve jakosti signalov iz petih različnih oddajnikov in ker imamo po celotnem testnem poligonu razporejenih deset oddajnikov, je teoretično vseh kombinacij $\binom{10}{5} = 252$. Nekaj kombinacij v realnosti odpade, ker niso vsi oddajniki dosegljivi v vsaki izbrani točki prostora. Pri simulaciji je bila določena kombinacija oddajnikov izbrana, če so bile vse jakosti signalov znotraj intervala $(-90, -59)$ dBm.

Za prvo izbrano točko $T_1(0, 2)$ so bile vse kombinacije oddajnikov in odmikov jakosti signalov sestavljene na osnovi tabele 6.4. Rezultat simulacije je prikazan na sliki 6.51. Vse možne položaje smo dobili z uporabo metode PSO, kjer smo upoštevali naslednje vrednosti parametrov: število iteracij = 100, velikost iskalnega območja = 8×8 m, $\omega = 1$, $\varphi_p = 0,5$ in $\varphi_g = 0,5$. Kot lahko opazimo na sliki 6.51,

je večina točk zbranih v enem roju. Za točke, ki pa so razpršene okoli roja pa lahko predpostavimo, da so rezultat slabe izbire (kombinacije bolj oddaljenih) oddajnikov (ali tudi slabe konvergence metode PSO).

Tabela 6.4: Odstopajoče jakosti signalov v točki $T_1(0, 2)$

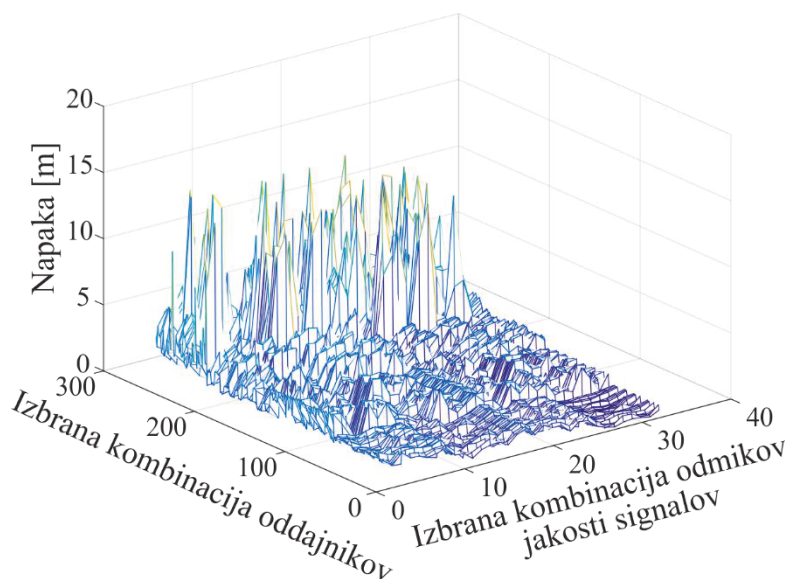
MAC Bluetooth oddajnika	$f_{RSSI_i}(d_i) - 2\sigma_i$ [dBm]	$f_{RSSI_i}(d_i) + 2\sigma_i$ [dBm]
E3:10:72:E5:75:09	-72	-65
C4:A6:FC:77:89:C7	-75	-66
F3:F4:4B:70:65:5E	-88	-79
DD:C2:72:C2:11:3A	-85	-76
E4:28:E3:81:A8:2C	-89	-77
D9:50:3B:F6:AA:46	-86	-67
DD:57:4F:2E:3B:DD	-88	-81
C6:63:3D:43:02:E5	-83	-75
C5:FA:6D:A0:46:25	-86	-75
DA:57:30:EC:6A:D1	-85	-79



Slika 6.51: Možni položaji pri upoštevanju vseh kombinacij iz tabele 6.4

Glede na izbrano kombinacijo oddajnikov in odmikov jakosti signalov (iz tabele 6.4) dobimo različno velike napake (slika 6.52), ki so določene kot razdalje med

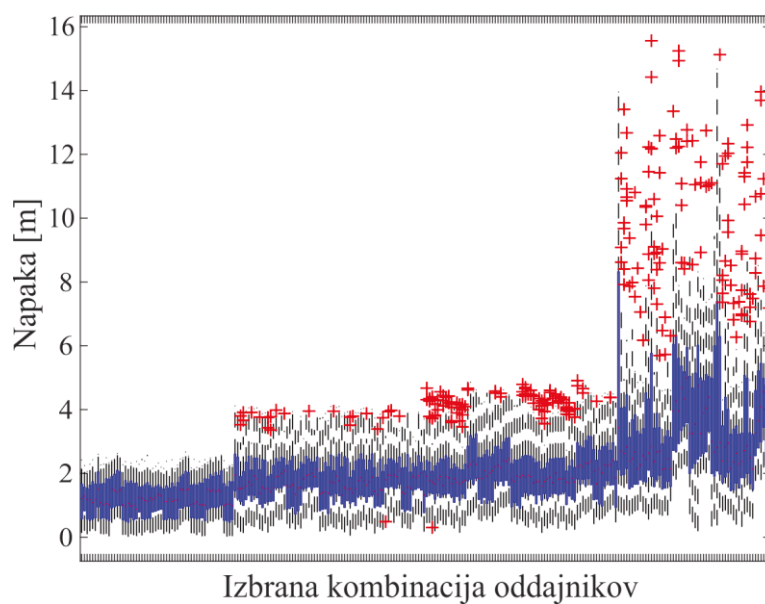
dejanskim položajem (izbrana točka T_1) in izračunanim položajem z metodo PSO. Opazimo lahko, da spreminjanje odmikov jakosti signalov za $\pm 2\sigma_i$ vpliva veliko manj na napako kot pa izbira različno oddaljenih oddajnikov. K napaki lahko prispeva tudi slaba konvergenca metode PSO, ki lahko pri iskanju globalnega minimuma zaide v lokalni minimum.



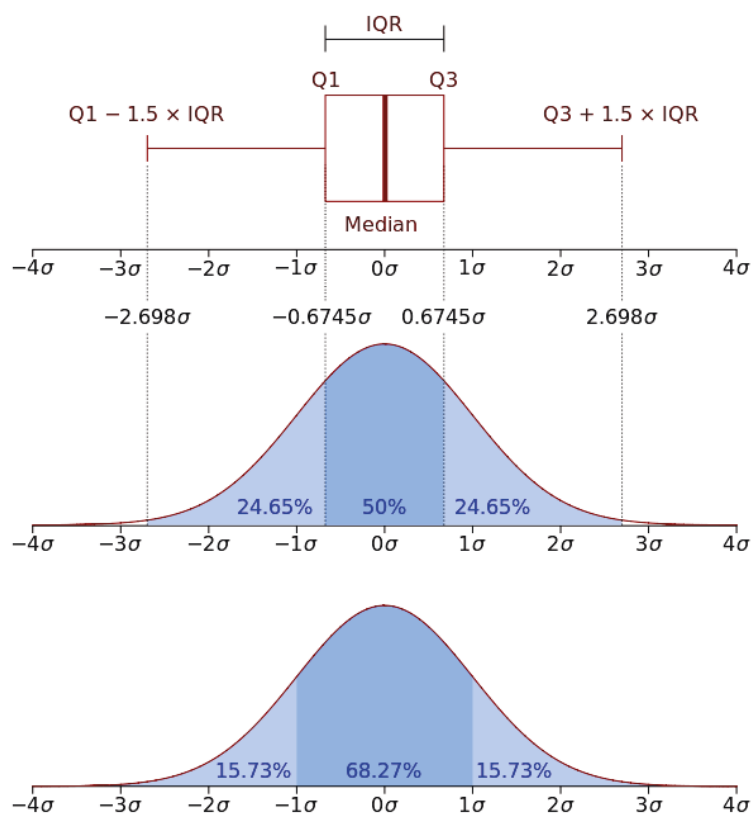
Slika 6.52: Napake pri različnih kombinacijah oddajnikov in odmikov jakosti signalov

Za lažji pregled smo napako pri različnih kombinacijah oddajnikov in odmikov jakosti signalov prikazali še v obliki grafa (slika 6.53), ki se imenuje »škatla z brki« (ang. box plot). Posamezen stolpec na sliki 6.53 prikazuje razpon napake (zaradi različnih kombinacij maksimalnih odmikov jakosti signalov) pri izbrani kombinaciji oddajnikov. V posameznem stolpcu je z modro označeno 50 % vseh napak, ki so razporejene okoli mediane napak oz. znotraj kvartilnega razmika (ang. interquartile range - IQR) med prvim kvartilom $Q1$ (25. percentila) in tretjim kvartilom $Q3$ (75. percentila, slika 6.54). Prikaz v tej obliki se uporablja tudi za določanje odstopajočih točk, pri katerih je napaka manjša od $Q1 - 1,5 * IQR$ ali večja od $Q3 + 1,5 * IQR$. Med slednji dve meji se uvrsti 99,3 % vseh napak zaradi odmikov jakosti signalov. Na desni strani grafa (sliki 6.53) je mogoče opaziti veliko rdečih križcev, ki predstavljajo zelo odstopajoče napake. To kaže na problematičnost izbrane

kombinacije oddajnikov in delno tudi na slabo konvergenco metode PSO znotraj velikega iskalnega področja.



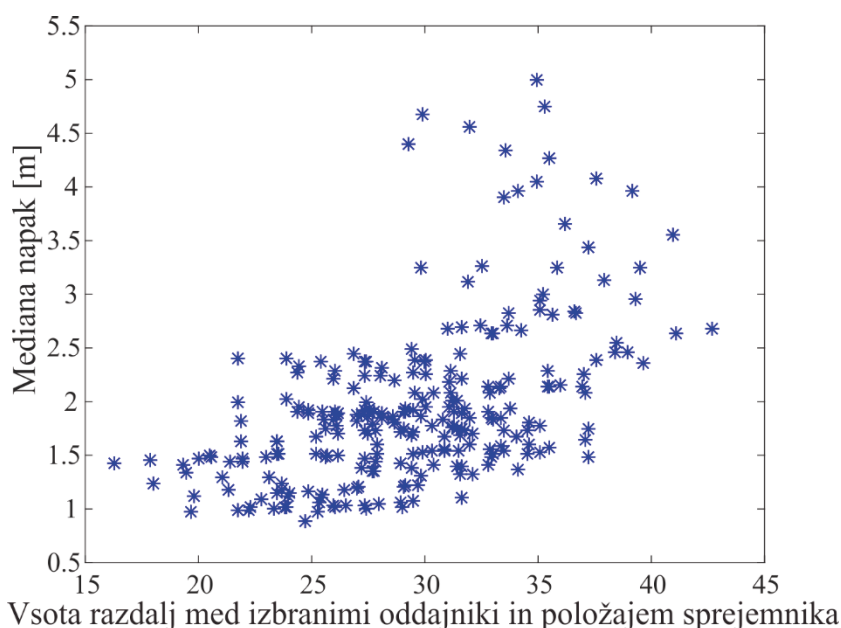
Slika 6.53: Razpon napake pri različnih kombinacijah oddajnikov v točki T_1



Slika 6.54: Označevanje področij znotraj grafa - »škatla z brki«

Z namenom, da bi ugotovili, ali je pri upoštevanju bolj oddaljenih Bluetooth oddajnikov napaka res večja, smo na sliki 6.55 prikazali mediano napak (za različne kombinacije odmikov jakosti signalov) v odvisnosti od vsote razdalj med točko v prostoru (T_1) in izbranimi oddajniki.

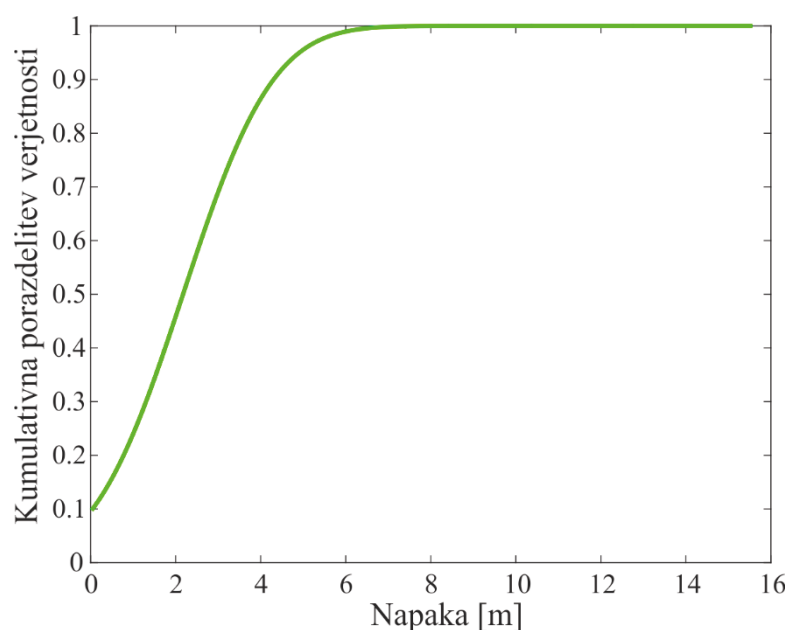
Ugotovitev (iz slike 6.53), da so napake večje pri kombinaciji bolj oddaljenih oddajnikov, potrjuje slika 6.55. Ker imamo pri enaki vsoti razdalj lahko kombinacijo oddajnikov, ki so približno enako oddaljeni od izbrane točke, ali pa kombinacijo zelo oddaljenih in bližnjih oddajnikov, so napake različno velike.



Slika 6.55: Vpliv vsote razdalj med izbranimi oddajniki in položajem sprejemnika na mediano napak

Iz slike 6.56, kjer je prikazana kumulativna porazdelitev verjetnosti napake pri različnih kombinacijah oddajnikov in odmikov jakosti signalov, lahko razberemo, da je (pri upoštevanih maksimalnih odmikih jakosti signalov za $\pm 2\sigma_i$) le 24 % vseh možnih napak manjših od 1 m oz. je 90 % vseh napak manjših od 4,3 m. Ta rezultat je relativno slab, vendar se moramo zavedati, da so v tem primeru upoštevani le »ekstremni« odmiki jakosti signalov.

V drugem primeru smo jakosti signalov spreminjali do še bolj ekstremnih vrednosti ($RSSI_i^{MAX} = f_{RSSI_i}(d_i) \pm 3\sigma_i$ za $i = 1, \dots, 5$), da bi na ta način upoštevali še večji del možnih odmikov.



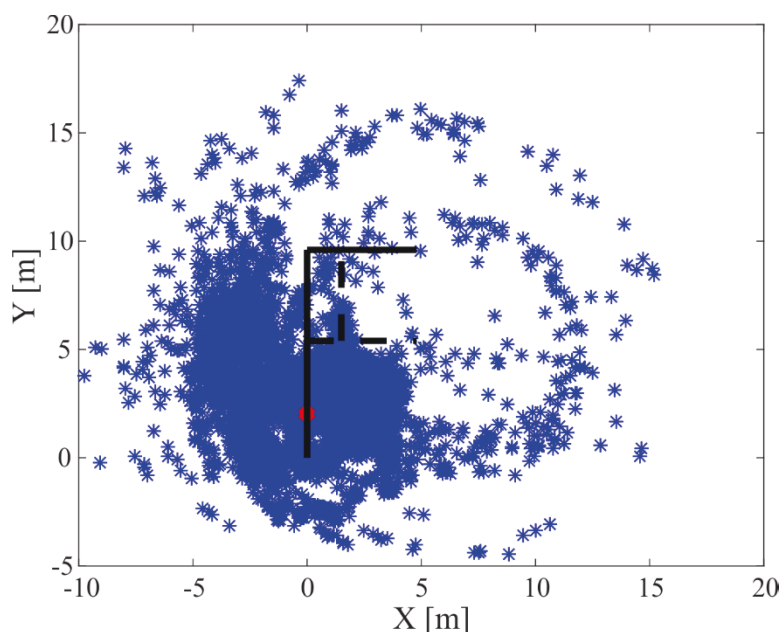
Slika 6.56: Kumulativna porazdelitev verjetnosti napake pri različnih kombinacijah oddajnikov in odmikov jakosti signalov v točki T_1

Tabela 6.5 prikazuje maksimalne odmike jakosti signalov v točki $T_1(0, 2)$, na podlagi katerih tvorimo vse možne kombinacije odmikov (za $\pm 3\sigma_i$) in oddajnikov.

Tabela 6.5: Odstopajoče jakosti signalov v točki $T_1(0, 2)$

MAC Bluetooth oddajnika	$f_{RSSI_i}(d_i) - 3\sigma_i$ [dBm]	$f_{RSSI_i}(d_i) + 3\sigma_i$ [dBm]
E3:10:72:E5:75:09	-74	-63
C4:A6:FC:77:89:C7	-78	-64
F3:F4:4B:70:65:5E	-90	-77
DD:C2:72:C2:11:3A	-87	-74
E4:28:E3:81:A8:2C	-91	-75
D9:50:3B:F6:AA:46	-90	-63
DD:57:4F:2E:3B:DD	-89	-80
C6:63:3D:43:02:E5	-85	-73
C5:FA:6D:A0:46:25	-88	-72
DA:57:30:EC:6A:D1	-86	-78

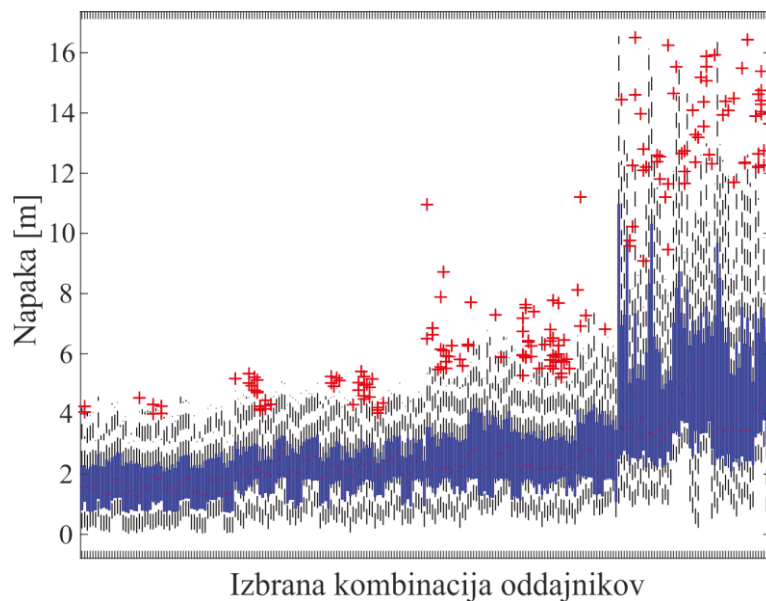
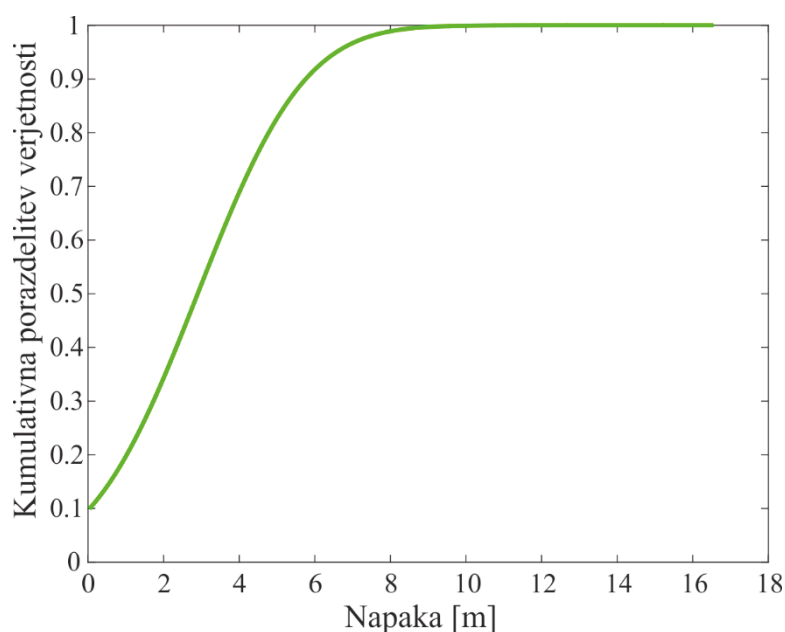
V tabeli lahko opazimo, da so v tem primeru nekatere jakosti signalov manjše od -90 dBm in jih moramo zato izpustiti iz množice možnih kombinacij ekstremnih jakosti signalov. Na sliki 6.57 je prikazan položaj vseh točk, ki jih dobimo z uporabo metode PSO pri različnih kombinacijah oddajnikov in odmikov jakosti signalov (glede na tabelo 6.5).



Slika 6.57: Možni položaji pri upoštevanju vseh kombinacij iz tabele 6.5

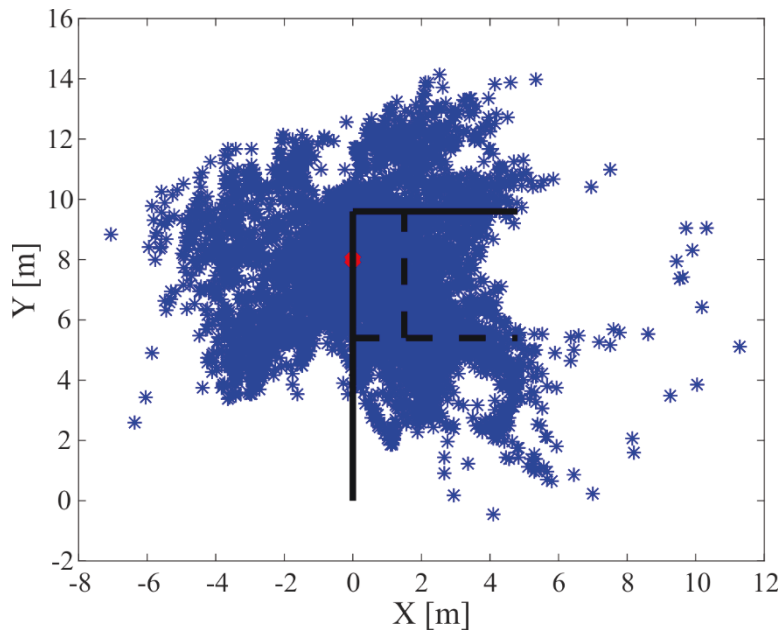
Če primerjamo graf na sliki 6.58 z grafom na sliki 6.53, lahko opazimo, da imajo napake v tem primeru pri določeni kombinaciji oddajnikov še nekoliko večji razpon (za približno 1 m). Iz slike 6.59 lahko razberemo, da je pri upoštevanih maksimalnih odmikih jakosti signalov (v točki T_1) za $\pm 3\sigma_i$ le 20 % vseh napak manjših od 1 m oz. je 90 % vseh napak manjših od 5,8 m. To pomeni, da se v primeru, ko jakosti signalov spreminjamo za $\pm 3\sigma_i$ (namesto $\pm 2\sigma_i$), polmer kroga, ki vsebuje večino (90 %) napak, poveča za okoli 1,5 m.

Pri simulaciji smo ugotovili, da za točko T_1 dobimo največjo napako, če uporabimo kombinacijo oddajnikov, ki so najbolj oddaljeni od točke T_1 . Razlog za to je eksponentno padanje jakosti signala z oddaljenostjo od oddajnika, saj (pri velikih oddaljenostih od oddajnika) že pri majhni spremembi jakosti signala dobimo veliko spremembo oddaljenosti od oddajnika.

Slika 6.58: Razpon napake pri različnih kombinacijah oddajnikov v točki T_1 Slika 6.59: Kumulativna porazdelitev verjetnosti napake pri različnih kombinacijah oddajnikov in odmikov jakosti signalov v točki T_1

V simulaciji smo obravnavali še dve točki, in sicer $T_2(0, 8)$ in $T_3(3; 9, 6)$. Ker smo v točki T_3 dobili zelo podobne rezultate kot v točki T_1 , si bomo v nadaljevanju pogledali le rezultate za točko T_2 . Ta se za razliko od točk T_1 in T_3 nahaja bolj na sredini prostora, kar pomeni, da so razdalje do oddajnikov manjše. Zato v tem primeru pričakujemo, da bodo napake pri različnih kombinacijah oddajnikov bolj izenačene.

Na sliki 6.60 so prikazani vsi možni položaji, ki jih lahko dobimo z metodo PSO, ko se nahajamo v točki T_2 . Posamezni položaji so določeni glede na izbrano kombinacijo odmikov jakosti signalov in kombinacijo oddajnikov (tabela 6.6). Odmike smo v tem primeru določili kot $RSSI_i^{MAX} = f_{RSSI_i}(d_i) \pm 3\sigma_i$ za $i = 1, \dots, 5$, kar pomeni, da bi za 99,74 % vseh možnih residuov (ki se lahko pojavijo pri posameznem modelu jakosti Bluetooth signala) dobili manjšo napako od največje napake, ki jo dobimo pri določeni kombinaciji oddajnikov.



Slika 6.60: Možni položaji pri upoštevanju vseh kombinacij iz tabele 6.6

Kot lahko opazimo na sliki 6.60, so v tem primeru izračunani položaji res bolj enakomerno porazdeljeni okoli dejanskega položaja, ki je na sliki označen z rdečo piko. Da je razpon napak pri različnih kombinacijah oddajnikov bolj izenačen, potrjuje tudi graf na sliki 6.61.

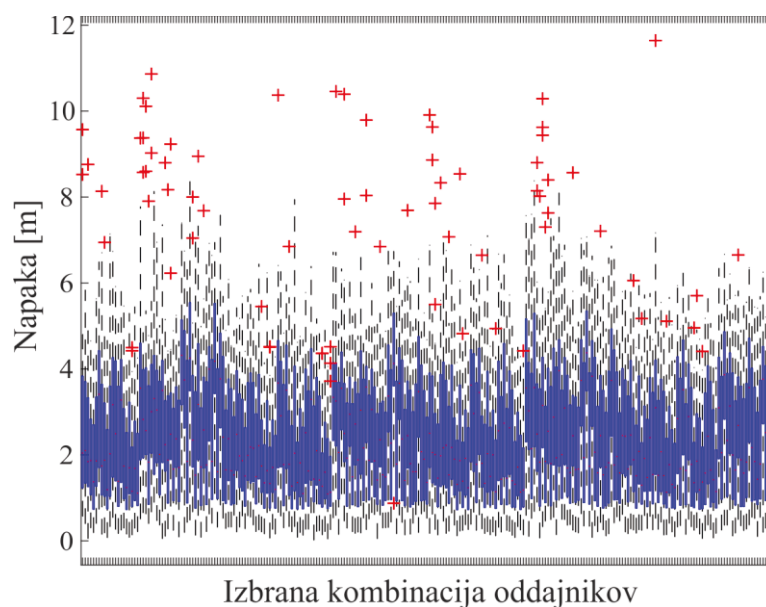
V tem primeru je manj tudi zelo odstopajočih točk (rdeči križci na sliki 6.61), saj so možni položaji manj razpršeni okoli dejanskega položaja. Razlog za to je boljša razporeditev oddajnikov okoli točke T_2 (nobeden oddajnik v primerjavi z drugimi ni zelo oddaljen od točke T_2).

Na sliki 6.62 je prikazan vpliv vsote razdalj med izbranimi oddajniki in položajem sprejemnika na mediano napak. Ugotovimo lahko, da tudi pri točki T_2 napaka (pri različnih kombinacijah maksimalnih odmikov jakosti signalov) narašča glede na vsoto

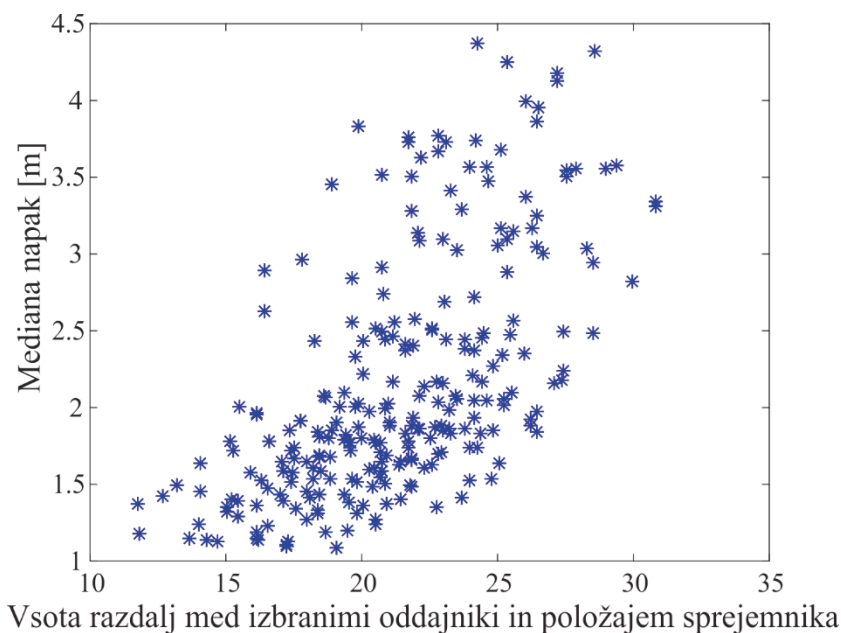
razdalj med izbranimi oddajniki in položajem sprejemnika. Iz dobljenih rezultatov lahko zaključimo, da je vedno bolje izbrati le najbližje oddajnike, če želimo zmanjšati vpliv šuma oz. odmikov jakosti signalov.

Tabela 6.6: Odstopajoče jakosti signalov v točki $T_2(0, 8)$

MAC Bluetooth oddajnika	$f_{RSSI_i}(d_i) - 3\sigma_i$ [dBm]	$f_{RSSI_i}(d_i) + 3\sigma_i$ [dBm]
E3:10:72:E5:75:09	−91	−80
C4:A6:FC:77:89:C7	−88	−75
F3:F4:4B:70:65:5E	−79	−66
DD:C2:72:C2:11:3A	−86	−73
E4:28:E3:81:A8:2C	−83	−67
D9:50:3B:F6:AA:46	−94	−67
DD:57:4F:2E:3B:DD	−84	−74
C6:63:3D:43:02:E5	−80	−68
C5:FA:6D:A0:46:25	−76	−60
DA:57:30:EC:6A:D1	−80	−71

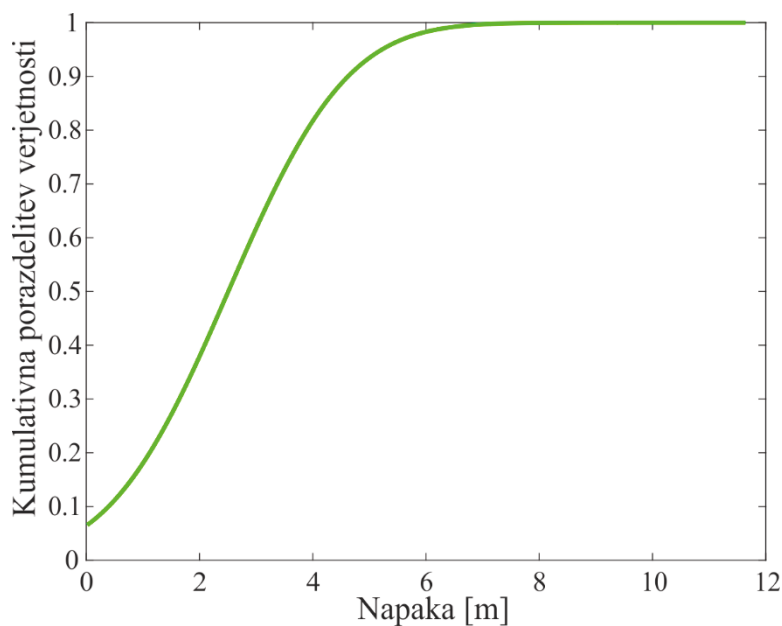


Slika 6.61: Razpon napake pri različnih kombinacijah oddajnikov v točki T_2



Slika 6.62: Vpliv vsote razdalj med izbranimi oddajniki in položajem sprejemnika na mediano napak

Iz grafa kumulativne porazdelitve verjetnosti napake (pri različnih kombinacijah oddajnikov in odmikov jakosti signalov), ki je prikazan na sliki 6.63, lahko razberemo, da je pri maksimalnih odmikih jakosti signalov okoli 18 % vseh napak manjših od 1 m oz. jih je 90 % manjših od 4,6 m.



Slika 6.63: Kumulativna porazdelitev verjetnosti napake pri različnih kombinacijah oddajnikov in odmikov jakosti signalov v točki T_2

Iz rezultatov simulacije lahko ugotovimo, da maksimalni odmiki jakosti signalov (za $2\sigma_i$ ali $3\sigma_i$) povzročijo zelo velike napake pri določanju položaja, še posebej pri uporabi bolj oddaljenih oddajnikov. Zavedati se moramo, da smo v naši simulaciji upoštevali le maksimalne odmike jakosti signalov, kar pomeni, da nam grafi kumulativne porazdelitve verjetnosti napake povedo le, kakšna je razporeditev napak pri maksimalnih odmikih jakosti signalov.

6.6 Lokalizacija z uporabo zemljevida prstnih odtisov

Lokalizacijski sistem, ki ga bomo obravnavali v tem poglavju, temelji na iskanju vektorja jakosti signalov, t.i. prstnega odtisa (ang. fingerprint), $RSSI_B$ znotraj podatkovne baze oz. zemljevida prstnih odtisov, ki je najbolj podoben trenutnemu vektorju izmerjenih jakosti signalov $RSSI_M$ iz bližnjih oddajnikov.

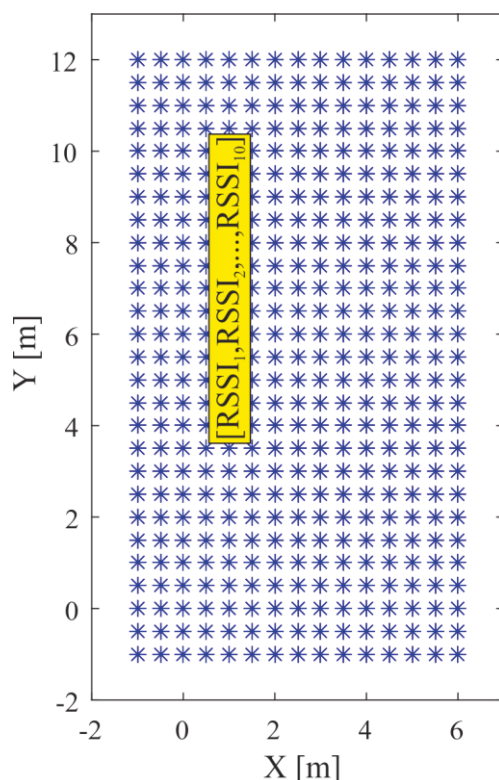
Sistem, ki temelji na uporabi »prstnih odtisov«, je običajno sestavljen iz dveh faz, in sicer faze učenja (»offline«) in faze določanja položaja (»online«). V fazi učenja je cilj zgraditi podatkovno bazo oz. zemljevid prstnih odtisov, ki vsebuje meritve jakosti Bluetooth signalov (iz vseh oddajnikov) za vse referenčne položaje v prostoru [243]. V fazi določanja položaja se trenutni vektor izmerjenih jakosti signalov $RSSI_M$ primerja z vrednostmi v bazi in glede na to, s katero referenčno točko se trenutni vektor najbolj ujema, se določi trenutni položaj (x, y) . Zgoraj omenjene probleme so v literaturi obravnavali številni avtorji [244] in eden izmed njih je tudi Jiang [245], ki je v svojem delu uporabil metodo KMLN (ang. K most likely neighbors) za iskanje najbolj podobnega »prstnega odtisa« trenutnemu vektorju meritev jakosti signalov.

6.6.1 Izgradnja zemljevida prstnih odtisov

Zemljevid prstnih odtisov (slika 6.64) je sestavljen iz enakomerno razporejenih točk po prostoru, katerim pripadajo vektorji jakosti signalov $RSSI_B = [RSSI_1, \dots, RSSI_m]$ iz vseh dostopnih oddajnikov (število vseh dostopnih oddajnikov je m). Torej baza podatkov oz. zemljevid prstnih odtisov za vsako referenčno točko prostora vsebuje informacije o jakosti signalov, oznake oddajnikov, iz katerih so bile zajete meritve v določeni točki in pa koordinate te točke (slika 6.65).

Zemljevid se v fazi učenja (»offline«) običajno zgradi tako, da se skupaj s sprejemnikom postavimo v vse referenčne točke prostora in tam izmerimo jakosti

signalov iz vseh dostopnih oddajnikov [246]. Če je v določeni točki zajetih več meritev iz istega oddajnika, se v prstnem odtisu upošteva povprečje vseh teh meritev. Za zadovoljivo natančnost lokalizacije mora biti gostota referenčnih točk dovolj velika. Veliko število referenčnih točk pa pomeni časovno dolgotrajen postopek zbiranja meritev v podatkovno bazo [247].



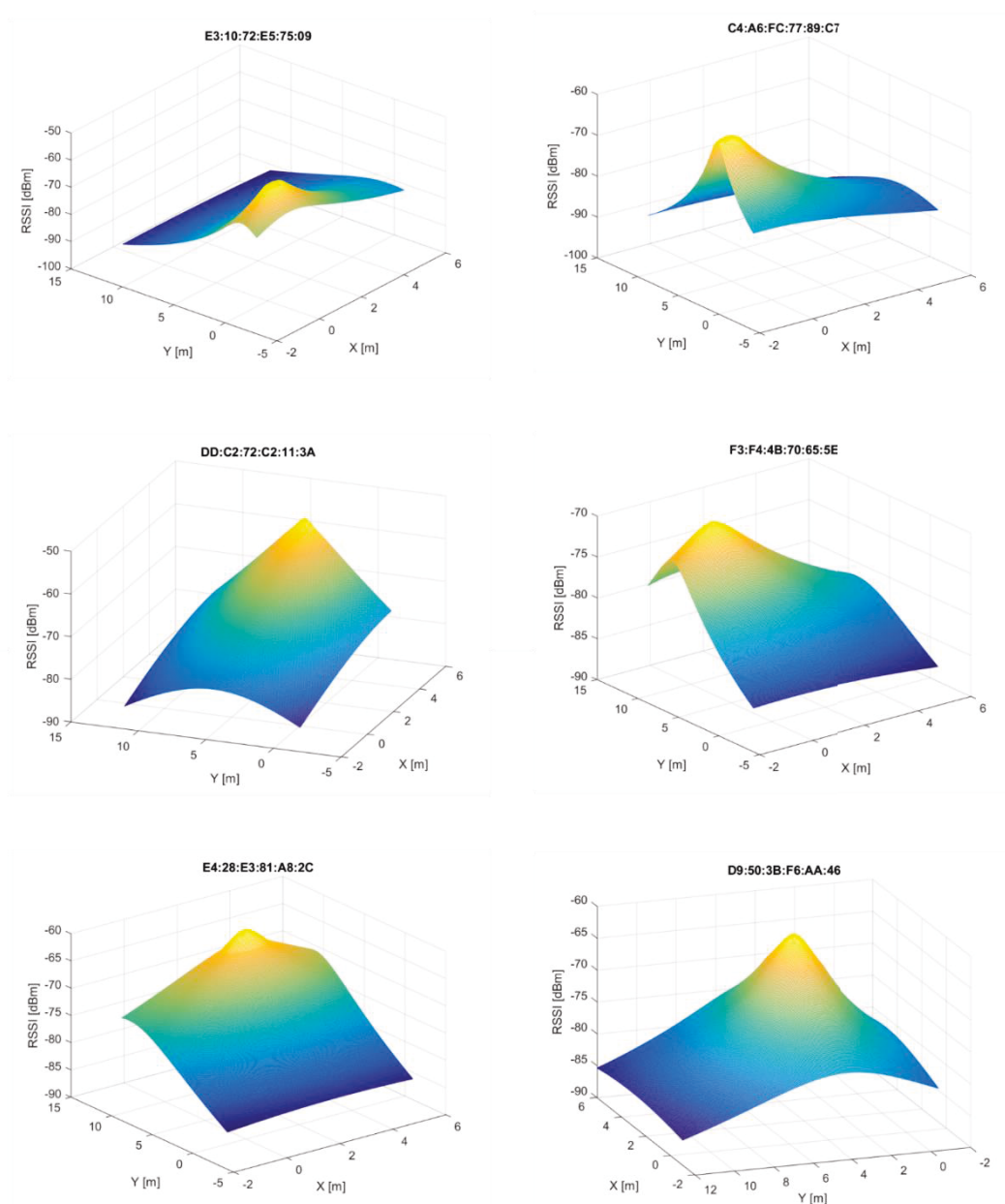
Slika 6.64: Zemljevid prstnih odtisov

Položaj (x,y)	id_1	id_2	...	id_{10}
	$RSSI_1$	$RSSI_2$	...	$RSSI_{10}$

Slika 6.65: Podatki, ki jih vsebuje posamezni prstni odtis

V našem primeru smo si gradnjo zemljevida prstnih odtisov nekoliko poenostavili (v smislu zbiranja meritev, saj pri uporabi lokalizacijskega algoritma, ki temelji na vizualni odometriji in inercialnem navigacijskem sistemu, le prehodimo izbrano pot in sproti pridobimo vse potrebne meritve za izgradnjo modelov) tako, da smo uporabili

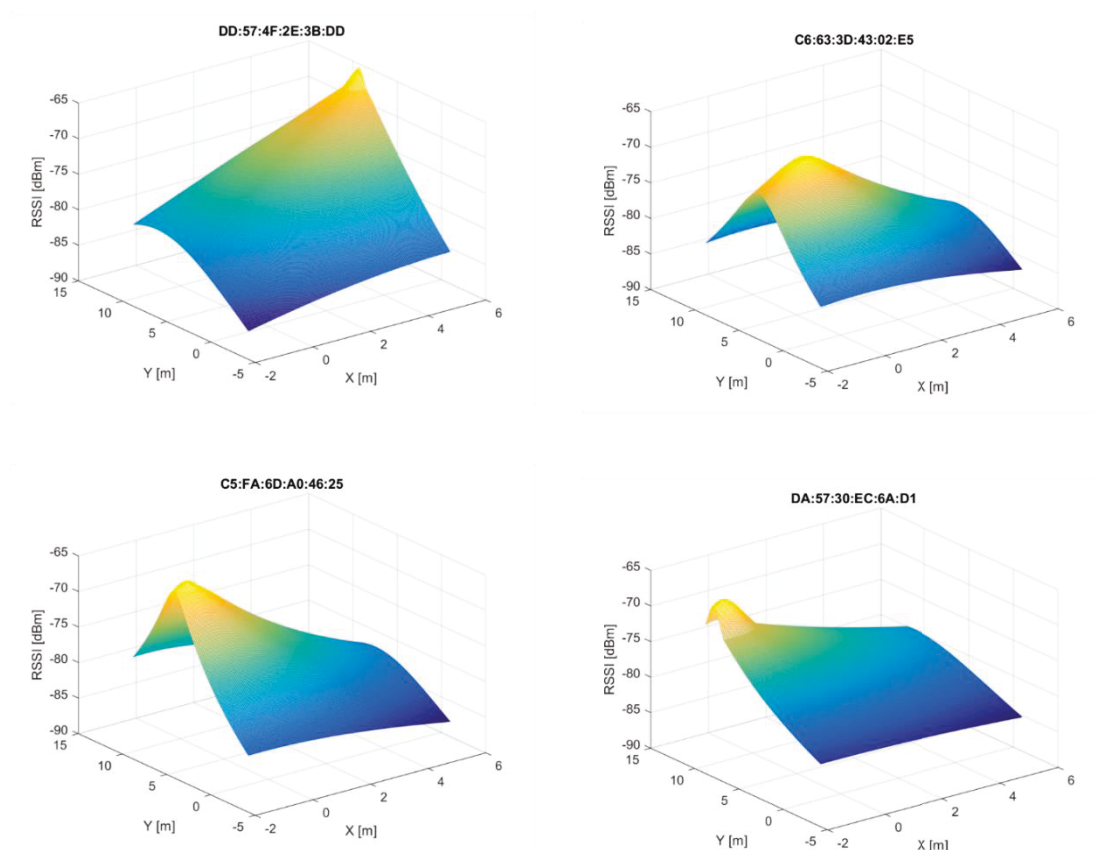
v okolju Matlab zgrajene modele širjenja jakosti signalov (6.2): $RSSI = f_{RSSI}(d)$ (sliki 6.66 in 6.67). Pri znanih razdaljah d med oddajniki (njihov položaj je znan) in referenčnimi točkami smo lahko z uporabo modelov določili jakosti signalov za vse prstne odtise $RSSI_B$ znotraj podatkovne baze. Podatkovno bazo smo shranili v obliki JSON formata, da je moč enostavno prenesti in uporabiti tudi na pametnem telefonu Android.



Slika 6.66: Prikaz širjenja Bluetooth signalov po prostoru na podlagi modelov

6.6.2 Metode iskanja najbližjih sosedov

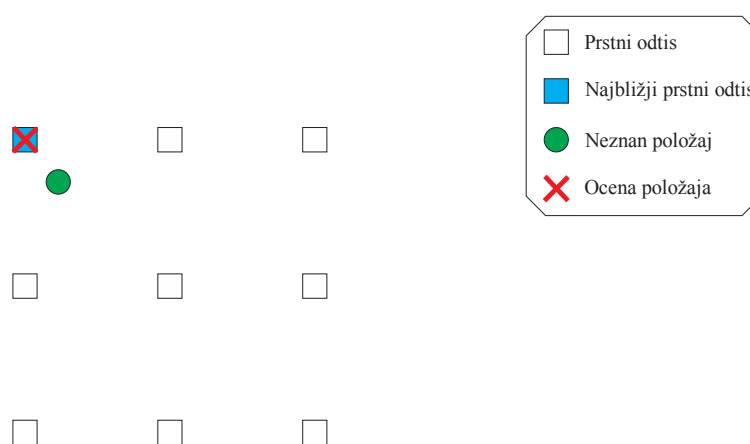
V fazi določanja položaja (»online«) je cilj poiskati (z uporabo ene izmed metod iskanja najbližjega sosedu) prstni odtis $RSSI_B$ znotraj podatkovne baze, ki je najbolj »podoben« (glede na kriterij) trenutnemu vektorju meritev jakosti signalov $RSSI_M$ [248]. Kot kriterij za določanje podobnosti med vektorji se lahko uporabi evklidska razdalja $l = \sqrt{\sum_{i=1}^n (RSSI_{B_i} - RSSI_{M_i})^2}$ (kjer je n število id-jev oddajnikov, ki so prisotni v obeh vektorjih), Kullback–Leibler divergenca itd. Pri primerjavi vektorjev se vedno upoštevajo le tisti indeksi (id-jev oddajnikov in pripadajočih jakosti signalov), ki so prisotni tako v vektorju $RSSI_B$ kot tudi $RSSI_M$.



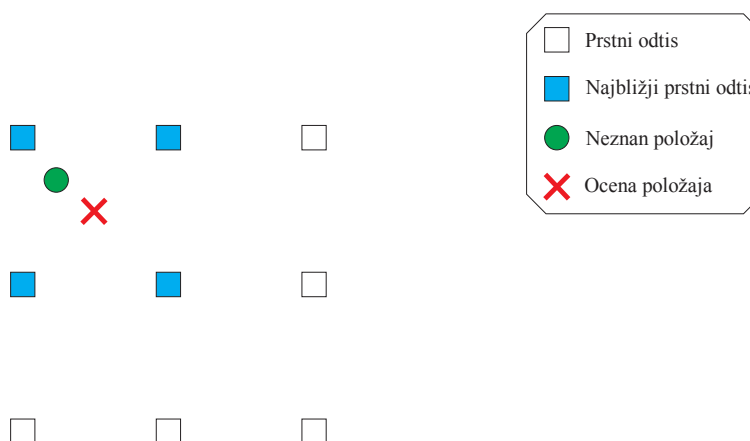
Slika 6.67: Prikaz širjenja bluetooth signalov po prostoru na podlagi modelov

Za določitev trenutnega položaja sprejemnika v prostoru je dovolj, da poiščemo le en prstni odtis, ki predstavlja najbližjega sosedu vektorju trenutnih meritev jakosti signalov (slika 6.68). Ta način določanja položaja je najbolj preprost, vendar pa je natančnost lokalizacije v tem primeru močno odvisna od gostote točk znotraj

zemljevida prstnih odtisov in varianc jakosti signalov. V praksi se zato pogosteje uporablja pristop z iskanjem K najbližjih sosedov (ang. K -nearest neighbors) znotraj podatkovne baze [249]. Na sliki 6.69 vidimo primer, pri katerem se za določanje položaja uporabljajo $K = 4$ najbližji sosedje. Če predpostavimo, da so vsi prstni odtisi enakovredni, potem ocenjeni položaj leži v središču upoštevanih najbližjih sosedov (položaj je določen kot povprečje koordinat K najbližjih sosedov). Natančnost določanja položaja se pri upoštevanju več najbližjih sosedov izboljša, saj na ta način zmanjšamo vpliv šuma, ki je prisoten v meritvah jakosti signalov.



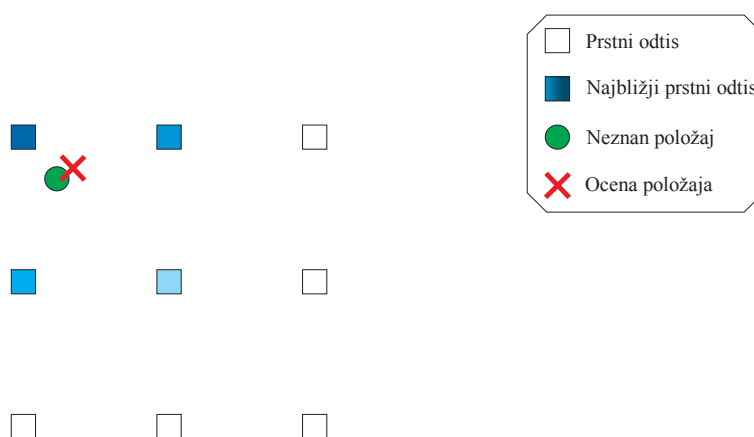
Slika 6.68: Ocena položaja z upoštevanjem najbližjega sosedu

Slika 6.69: Ocena položaja z upoštevanjem K najbližjih sosedov

Pri računanju podobnosti med vektorji iz podatkovne baze $\mathbf{RSSI}_B$ in vektorjem meritev $\mathbf{RSSI}_M$ so med K najbolj podobnimi vektorji eni vektorji bližje vektorju $\mathbf{RSSI}_B$ kot drugi. Zato je pri določanju položaja smiselno uporabiti recipročno interpolacijo K najbližjih sosedov (slika 6.70). Prispevke posameznih vektorjev lahko

utežimo z uporabo metode uteženih K najbližjih sosedov (ang. Weighted K -Nearest-Neighbor). Uteži oz. prispevke posameznih prstnih odtisov izračunamo kot: $w_k = \frac{1}{l_k}$, kjer je l_k razdalja med vektorjem iz podatkovne baze $\mathbf{RSSI}_{B_k}$ in vektorjem meritev $\mathbf{RSSI}_M$. Na ta način je prispevek (k oceni položaja) bolj oddaljenih prstnih odtisov manjši kot prispevek prstnih odtisov, ki so blizu vektorju meritev. Če je $\mathbf{x}_{B_k}$ položaj prstnega odtisa $\mathbf{RSSI}_{B_k}$, potem je ocenjeni položaj določen kot:

$$\mathbf{x}_M = \frac{\sum_{k=1}^K w_k \mathbf{x}_{B_k}}{\sum_{k=1}^K w_k}. \quad (6.108)$$



Slika 6.70: Ocena položaja z upoštevanjem uteženih K najbližjih sosedov

Drevo odločitvenih točk (Vantage point tree)

Pri določanju položaja z uporabo prstnih odtisov se največ računskega časa porabi za izračun razdalj med vektorji znotraj podatkovne baze in vektorjem trenutnih meritev ter za sortiranje teh razdalj. Da bi dosegli čim krajši računski čas (potreben za izračun položaja), smo preučili možnosti uporabe podatkovnih dreves za namen hitrega iskanja najbližjih sosedov.

Na področju podatkovnega rudarjenja (ang. data mining) sta algoritma za k -d drevo (k -d tree) in drevo odločitvenih točk (ang. Vantage point tree – vp -tree) [250] zelo priljubljena, saj omogočata hitro in učinkovito iskanje najbližjih sosedov poizvedbenega elementa v velikih podatkovnih bazah. Pri k -d drevesu se podatkovna struktura zgradi s ponovljivim postopkom razvrščanja elementov po drevesu glede na določeno dimenzijo razvrščanja. To pomeni, da v primeru, ko razvrščamo elemente iz

2D prostora, enkrat delimo elemente po prvi dimenziji X in spet drugič po drugi dimenziji Y . Prednost $k-d$ dreves (za namen hitrega iskanja najbližjih sosedov) hitro izgine pri visoko dimenzionalnih podatkih, saj je težko izbrati dimenzijo, ki bi bila najbolj primerna za razvrščanje podatkov na določenem nivoju drevesa. Posledično algoritem v fazi iskanja najbližjega sosedu preišče praktično vse elemente $k-d$ drevesa (kar je enako trivialnemu pristopu, pri katerem bi izračunali razdalje od poizvedbenega elementa do vseh elementov v bazi in nato izbrali najbližjega).

Namen vp -dreves, ki jih uvrščamo med BSP (binary space partitioning) drevesa, je hitro in natančno iskanje najbližjih sosedov v splošnem metričnem prostoru, v katerem je dana podatkovna baza z množico elementov S . Z vp -drevesom se metrični prostor razdeli na več sferičnih delov, katerih središče je določeno z izbrano odločitveno točko $\mathbf{v}$ (ang. vantage point) [250, 251]. Bistvena razlika pri vp -drevesih v primerjavi s $k-d$ drevesi je v tem, da se elementi podatkovne baze razdelijo po drevesu (tvorijo podatkovno strukturo) glede na razdalje (ang. »distance based indexing« ali tudi »metric space model indexing«) do izbranih t.i. odločitvenih točk in ne le na posamezne koordinate (dimenzije prostora). Za merjenje razdalj med elementi lahko uporabljamo npr. evklidsko razdaljo. Podatkovna struktura vp -drevesa omogoča, da shranimo elemente podatkovne baze na tak način, da v trenutku poizvedbe (pri iskanju najbližjega sosedu) lahko hitro izločimo (tj. jih ne preverjamo) velik del elementov podatkovne baze, brez da bi morali računati razdalje med temi elementi in elementom poizvedbe. Za vse razdalje med elementi vp -drevesa mora veljati trikotniška neenakost. To pomeni, da če imamo odločitveno točko $\mathbf{v} \in S$, glede na katero razvrščamo ostale elemente (npr. elementa $\mathbf{a}, \mathbf{b} \in S$) podatkovne baze, mora veljati naslednje (pri upoštevanju reverzne trikotniške neenakosti):

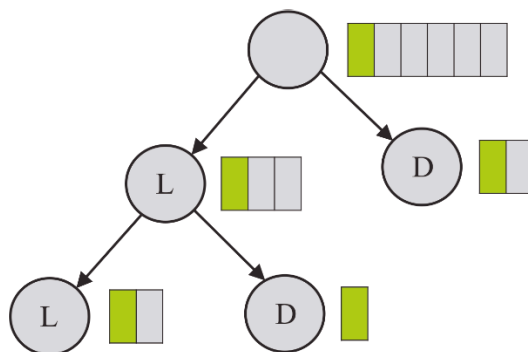
$$d(\mathbf{a}, \mathbf{b}) \geq |d(\mathbf{a}, \mathbf{v}) - d(\mathbf{b}, \mathbf{v})| = d_v(\mathbf{a}, \mathbf{b}), \quad (6.109)$$

kjer je:

$$\begin{aligned} \Pi_v(\mathbf{a}) &= d(\mathbf{a}, \mathbf{v}), \\ d_v(\mathbf{a}, \mathbf{b}) &= |\Pi_v(\mathbf{a}) - \Pi_v(\mathbf{b})| = |d(\mathbf{a}, \mathbf{v}) - d(\mathbf{b}, \mathbf{v})|. \end{aligned} \quad (6.110)$$

Na sliki 6.71 je prikazana struktura vp -drevesa, ki kaže, kako se elementi podatkovne baze med gradnjo drevesa razdelijo po posameznih vozliščih drevesa od najvišjega (prvo vozlišče drevesa je t.i. koren drevesa, ki v seznamu elementov vsebuje vse posamezne elemente iz nabora podatkov) do najnižjega nivoja (kjer so v vozliščih

le še posamezni podatkovni elementi). Pri prehajanju po drevesu navzdol se vsako vozlišče razdeli na dva podrejena vozlišča (levo in desno), kar pomeni, da se tudi niz podatkovnih elementov v vsakem vozlišču razdeli na dva dela, ki potem pripadata levemu oz. desnemu podrejenemu vozlišču.



Slika 6.71: Koncept vp-drevesa (v seznamu elementov so z zeleno označene odločitvene točke in s sivo vsi ostali elementi)

Po definiciji vsako vozlišče vp-drevesa hrani naslednje informacije [250]:

- seznam elementov, ki so del podatkovne baze oz. množice S ,
- naključno izbrano (iz zgornjega seznama elementov) odločitveno točko v (ang. vantage point),
- razdalja μ (predstavlja mediano razdalj in določa področje, ki ga pokriva to vozlišče),
- levo podrejeno vozlišče (»levi otrok« – oznaka L) in
- desno podrejeno vozlišče (»desni otrok« – oznaka D).

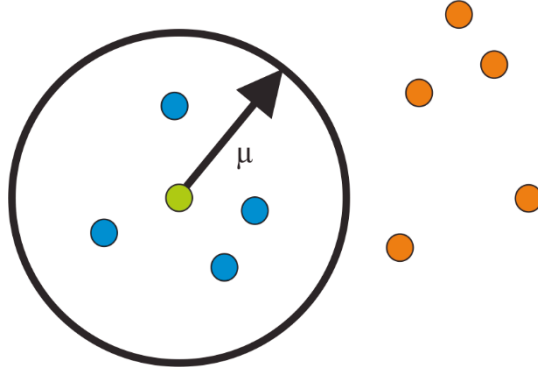
Razdaljo μ vsakega vozlišča izračunamo kot mediano razdalj med odločitveno točko v in vsemi ostalimi elementi P iz seznama elementov tega vozlišča:

$$\mu = \text{median}(\text{dist}(v, p)) \quad \forall p \in P. \quad (6.111)$$

Vse elemente, ki so od odločitvene točke oddaljeni za manj kot μ , dodelimo levemu podrejenemu vozlišču, elemente, ki pa so oddaljeni več (ali enako) kot μ , pa dodelimo desnemu podrejenemu vozlišču (slika 6.72 prikazuje primer za dvorazsežni evklidski prostor). Z vsakim novim vozliščem je njegova pripadajoča podmnožica elementov podatkovne baze manjša. Ker razdalja μ predstavlja mediano razdalj med izbrano odločitveno točko v in vsemi ostalimi elementi tega vozlišča, algoritem v naslednjem

nivoju drevesa razdeli vse elemente iz množice P , na dve enako veliki množici S_1 in S_2 , ki pripadata levemu oz. desnemu podrejenemu vozlišču:

$$\begin{aligned} S_1 &= \{p \in S \mid d(p, v) < \mu\}, \\ S_2 &= \{p \in S \mid d(p, v) \geq \mu\}. \end{aligned} \quad (6.112)$$



Slika 6.72: Območje, ki ga pokriva vozlišče (z zeleno označena odločitvena točka). Z modro označena vozlišča pripadajo levemu podrejenemu vozlišču, z oranžno označena vozlišča pa pripadajo desnemu podrejenemu vozlišču.

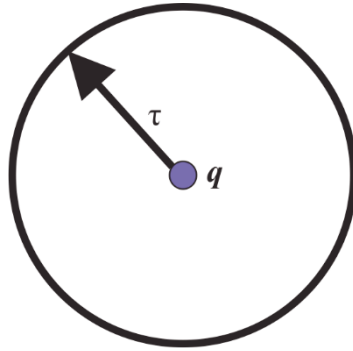
Vp -drevo se zgradi z rekurzivnim postopkom, v katerem se za vsako novo vozlišče ponovita dva ključna koraka, in sicer: 1. naključna izbira odločitvene točke v iz množice vseh elementov vozlišča in 2. razvrščanja preostalih elementov v dve množici (glede na razdaljo μ), ki pripadata levemu oz. desnemu podrejenemu vozlišču. Postopek se ponavlja, dokler ne zmanjka elementov, ki bi jih dodelili levemu oz. desnemu podrejenemu vozlišču. Vozlišča na najnižjem nivoju drevesa, ki nimajo podrejenih vozlišč, se imenujejo končna vozlišča (ang. »leaf nodes«).

Iskanje K najbližjih sosedov z uporabo vp -drevesa

Ko imamo podatke iz podatkovne baze shranjene v obliki vp -drevesa, lahko hitro in učinkovito poiščemo K najbližjih sosedov k poizvedbenemu elementu q , brez da bi računali razdalje do vseh posameznih elementov baze [251, 252]. Iskanje najbližjega sosedu v množici elementov S označimo kot $NN|_{\tau}(q, S)$. Predpostavimo lahko, da za vsak poizvedbeni element q obstaja taka pragovna razdalja τ , pri kateri se vseh K najbližjih sosedov elementa q nahaja na razdalji manjši od tega praga. Iz enačbe (6.109) lahko zaključimo, da velja naslednje:

$$d_v(a, b) \geq \tau \Rightarrow d(a, b) \geq \tau. \quad (6.113)$$

To področje, znotraj katerega se mora nahajati K najbližjih sosedov, si lahko predstavljamo kot krog (to velja za 2D prostor, pri večdimenzionalnih podatkovnih elementih pa kroge zamenjajo hiperkrogle), ki je prikazan na sliki 6.73. Z uporabo vp -drevesa je mogoče najti najbližjega soseda v času $O(\log(n))$ oz. največ $M \cdot [\log_2(n) + 1]$ izračunov (evklidskih) razdalj je potrebno, da najdemo najbližjega soseda $NN|_\tau(\mathbf{q}, S)$ k poizvedbenemu elementu $\mathbf{q}$. Parameter $M > 1$ pove, kolikokrat smo šli skozi vse nivoje drevesa pri iskanju najbližjega soseda.



Slika 6.73: Področje okoli poizvedbenega elementa $\mathbf{q}$, znotraj katerega se morajo nahajati najbližji sosedje

Glede na razdaljo poizvedbenega elementa $\mathbf{q}$ do določenega vozlišča znotraj vp -drevesa in velikost področja okoli poizvedbenega elementa $\mathbf{q}$ (ki ga določa polmer τ) imamo lahko tri scenarije. V prvem scenariju, ki ga lahko vidimo na sliki 6.74, velja, da področje (ki ga določa razdalja τ) okoli poizvedbenega elementa $\mathbf{q}$, leži povsem izven področja (ki ga določa razdalja μ) okoli odločitvene točke $\mathbf{v}$. To pomeni, da velja:

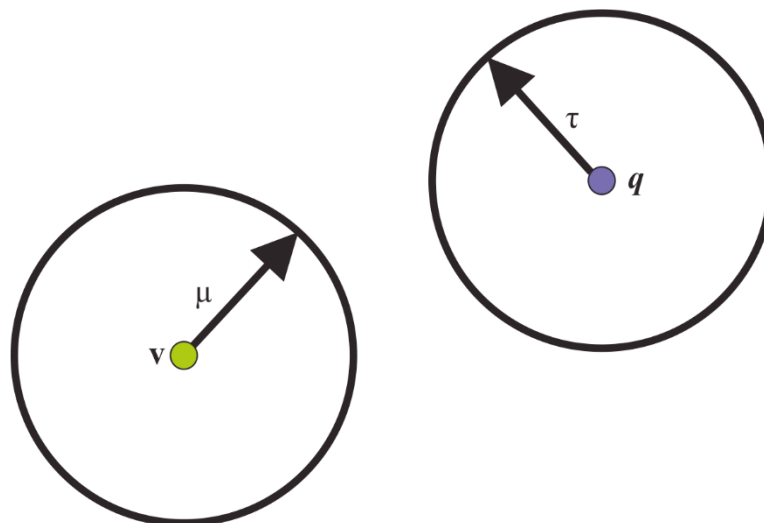
$$\Pi_{\mathbf{v}}(\mathbf{q}) = d(\mathbf{v}, \mathbf{q}) \geq \mu + \tau, \quad (6.114)$$

kar nadalje pomeni, da lahko v postopku iskanja najbližjih sosedov izpustimo vse elemente, ki se nahajajo v levem podrejenem vozlišču oz. znotraj področja, določenega z razdaljo μ okoli odločitvene točke $\mathbf{v}$.

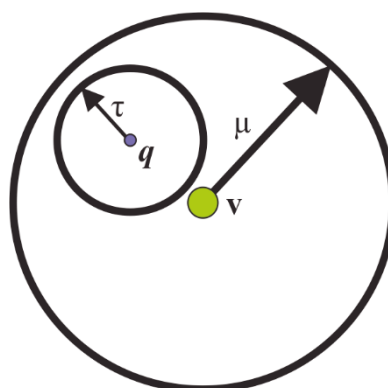
V drugem scenariju, ki ga vidimo na sliki 6.75, se področje (ki ga definira krog z radijem τ) okoli poizvedbene točke $\mathbf{q}$ nahaja povsem znotraj področja (ki ga definira krog z radijem μ) okoli odločitvene točke $\mathbf{v}$. V tem primeru velja:

$$\Pi_{\mathbf{v}}(\mathbf{q}) = d(\mathbf{v}, \mathbf{q}) \leq \mu - \tau, \quad (6.115)$$

kar pomeni, da lahko v taki situaciji izpustimo vse elemente, ki se nahajajo v desnem podrejenem vozlišču oz. izven področja, določenega z razdaljo μ okoli odločitvene točke.



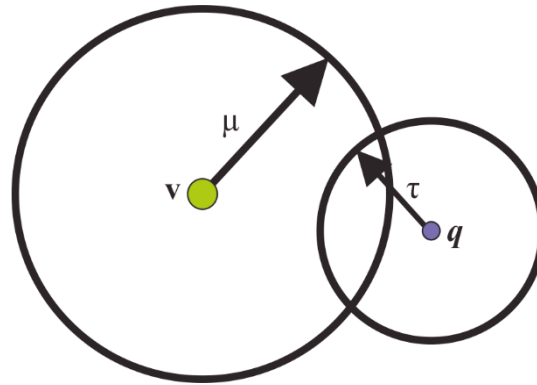
Slika 6.74: Scenarij, pri katerem je poizvedbena točka q oddaljena od odločitvene točke v za več kot $(\mu + \tau)$



Slika 6.75: Scenarij, pri katerem je poizvedbena točka q oddaljena od odločitvene točke v za manj kot $(\mu - \tau)$

Tretji scenarij, ki pa je od vseh treh najmanj zaželen, predstavlja situacijo, pri kateri področje okoli poizvedbene točke q delno prekriva področje okoli odločitvene točke v (slika 6.76). To pomeni, da moramo v tem primeru upoštevati tako elemente levega kot tudi desnega podrejenega vozlišča. Poizvedbena točka q se v tem primeru nahaja na razdalji, ki je določena z intervalom:

$$\mu - \tau \leq d(v, q) \leq \mu + \tau. \quad (6.116)$$



Slika 6.76: Scenarij, pri katerem je poizvedbena točka q oddaljena od odločitvene točke v za več kot $(\mu - \tau)$ in manj kot $(\mu + \tau)$

Algoritem iskanja K najbližjih sosedov z uporabo vp -drevesa:

VP_KNN_Iskanje(q, K , koren)

Vhod: poizvedbena točka q , zahtevano število najbližjih sosedov K , korensko vozlišče vp -drevesa.

Izhod: množica W , ki vsebuje K najbližjih sosedov za točko q .

Start:

Lokalna spremenljivka τ : začetna maks. razdalja točke q do najbližjega sosedu.

Iskanje(q, K , koren, τ, W);

vrni W ;

Konec

Iskanje(q, K , koren, τ, W)

Vhod: poizvedbena točka q , zahtevano število najbližjih sosedov K , korensko vozlišče vp -drevesa in množica W , ki vsebuje K do sedaj najdenih najbližjih sosedov.

Izhod: posodobljene vrednosti za τ in W

Start:

dodaj koren drevesa v vrsto_vozlišč (nodes_queue)

dokler ni vrsta_vozlišč prazna

node = zadnje dodano vozlišče v vrsto_vozlišč;

razd. = $d(\text{node}, v, q)$, kjer je v odločitvena točka trenutno izbranega vozlišča;

odstrani vozlišče z odločitveno točko v iz vrste_vozlišč;

če je razd. $\leq \tau$

postavi node. v na pravilen položaj znotraj W (množica W je vedno sortirana v naraščajočem vrstnem redu glede na razdaljo $d(w, q)$, kjer je $w \in W$).

če velja $|W| = K$ potem je $\tau = d(w_K, q)$;

če je razd. $\leq \text{node}.\mu + \tau$

dodaj node. *left_child* v vrsto_vozlišč;

če je razd. $\geq \text{node}.\mu - \tau$

dodaj node. *right_child* v vrsto_vozlišč;

Konec

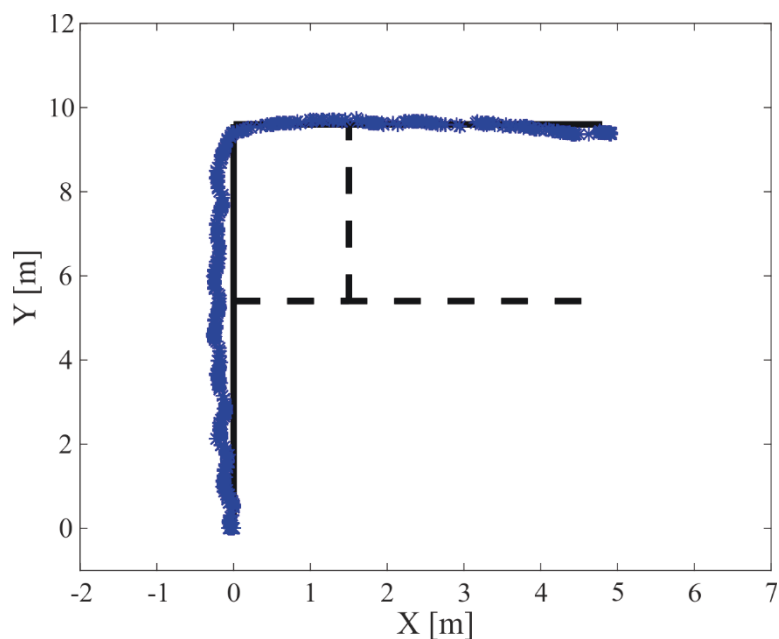
Začetno vrednost parametra τ nastavimo na neskončno vrednost ali poljubno končno vrednost, ki bo določala maksimalen odmik najbližjih sosedov od poizvedbene točke q . Od trenutka dalje, ko imamo najdenih K kandidatov, ki so od poizvedbenega elementa oddaljeni za manj kot je začetna vrednost parametra τ , je slednji določen z

razdaljo najbolj odmaknjenega kandidata (najslabši kandidat na seznamu najbližjih K sosedov) od poizvedbenega elementa q . Torej se pri rekurzivnem postopku (tj. pri pomikanju po vp -drevesu navzdol) z vsakim na novo dodanim kandidatom na seznam najbližjih sosedov zmanjša vrednost τ .

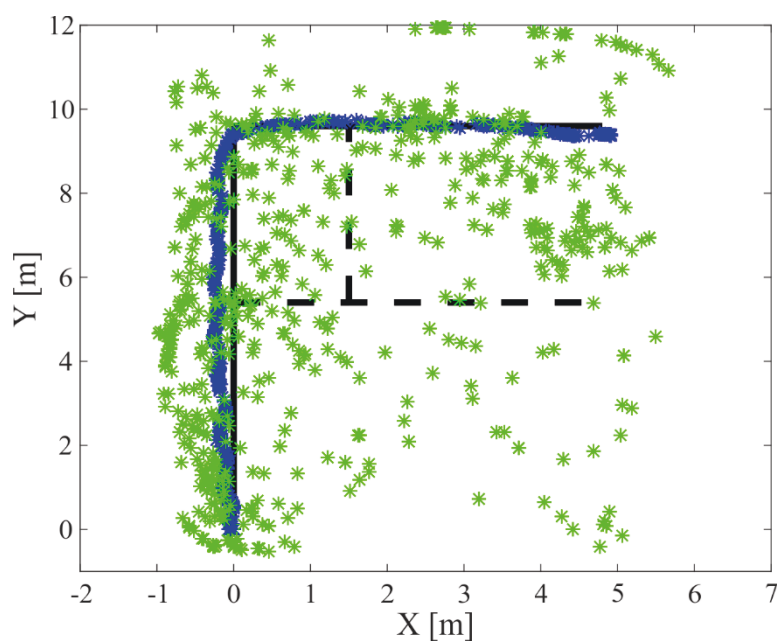
6.6.3 Rezultati lokalizacije na osnovi prstnih odtisov

Za namen preizkusa lokalizacijskega algoritma, ki temelji na uporabi podatkovne baze prstnih odtisov, smo s pametnim telefonom Android opravili meritve jakosti signalov po poti, ki je prikazana na sliki 6.77. Modre točke prikazujejo vse položaje, v katerih je bil kreiran nov vektor jakosti signalov $\mathbf{RSSI}_M$. Ta je kreiran vsakič, ko sprejemnik izmeri vsaj eno novo jakost signala, ki je večja od -90 dBm. V našem primeru se to zgodi v povprečju vsakih 30 ms (pri čemer vsak oddajnik pošlje paketek vsakih 100 ms). Da bi lahko določili natančnost lokalizacijskega algoritma, smo pri kreiranju vsakega vektorja $\mathbf{RSSI}_M$ zabeležili tudi dejanski položaj, v katerem je bil vektor kreiran. Položaje sprejemnika (pametnega telefona) smo določali z uporabo algoritma, ki temelji na vizualni odometriji in inercialnem navigacijskem sistemu. Natančnost tega algoritma je večja od natančnosti (20~30 cm), ki jo doseže pešec pri sproščeni hoji po krivulji, označeni na tleh (glej sliko 6.77). Vse vektorje jakosti signalov in položaje, v katerih so bili kreirani (slika 6.77), smo na pametnem telefonu shranili v datoteko in jo prenesli v okolje Matlab, kjer smo izvedli več eksperimentov z izbranim lokalizacijskim algoritmom.

Pri preizkusu lokalizacijskega algoritma, ki temelji na uporabi prstnih odtisov, smo najprej upoštevali vektorje meritev jakosti signalov $\mathbf{RSSI}_M$ iz treh različnih oddajnikov (dolžina vektorja $\mathbf{RSSI}_M$ je 3). V okolju Matlab smo z uporabo funkcije »knnsearch« za vsak vektor $\mathbf{RSSI}_M$ poiskali $K = 16$ (pri tej vrednosti so bili rezultati lokalizacije najboljši) najbližjih sosedov (iz množice prstnih odtisov, shranjenih v podatkovni bazi). Položaje, ki pripadajo vektorjem $\mathbf{RSSI}_B$, smo utežili z recipročnimi vrednostmi evklidskih razdalj med najbližjimi sosedi in vektorjem $\mathbf{RSSI}_M$ in tako po enačbi (6.108) dobili položaj, kjer naj bi bil ustvarjen vektor meritev jakosti signalov $\mathbf{RSSI}_M$. Na sliki 6.78 je prikazan rezultat določanja položaja. Kot lahko opazimo, je rezultat lokalizacije precej slab, kar pomeni, da vektorji $\mathbf{RSSI}_M$ dolžine 3 slabo definirajo položaj.

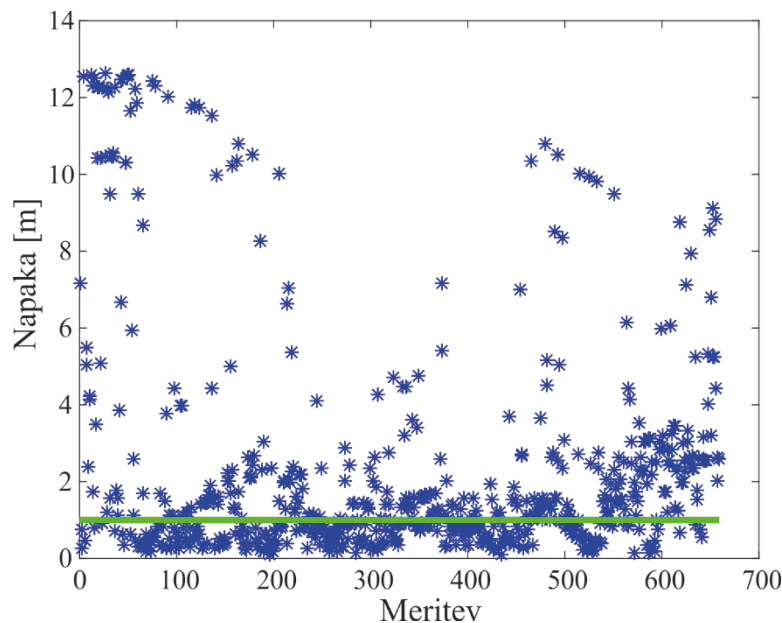


Slika 6.77: Položaji (določeni z uporabo vizualne odometrije in inercialnega navigacijskega sistema), v katerih so bile zajete meritve jakosti signalov $RSSI_M$



Slika 6.78: Določanje položaja v prostoru pri uporabi meritev jakosti signalov iz treh različnih oddajnikov (dolžina vektorja $RSSI_M$ je 3) in upoštevanju 16 najbližjih sosedov ($K=16$)

Iz slike 6.79 lahko razberemo, da napake (razdalje med izračunanim in dejanskim položajem) v tem primeru segajo tudi preko 12 m. S povečevanjem števila upoštevanih najbližjih sosedov rezultatov lokalizacije v tem primeru ne moremo kaj dosti izboljšati.



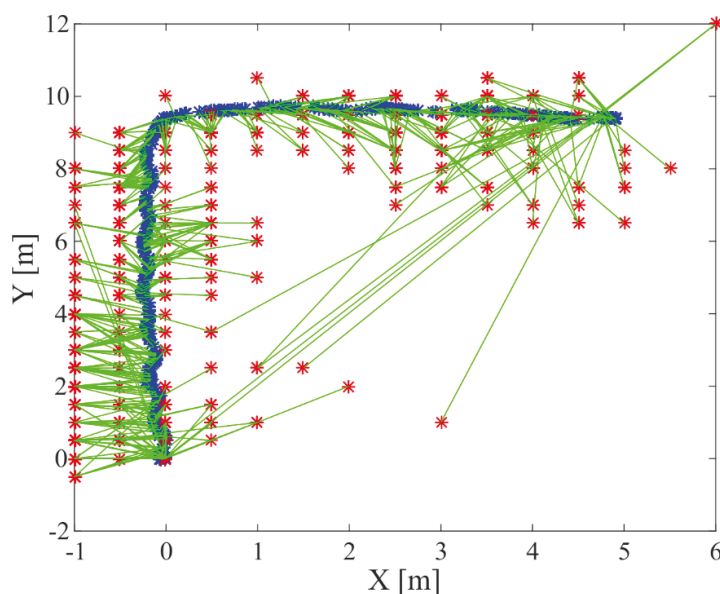
Slika 6.79: Napake pri določanju položaja v primeru, ko so upoštevane meritve jakosti signalov iz treh različnih oddajnikov in je $K=16$

Ker vektorji $\mathbf{RSSI}_M$ dolžine 3 slabo definirajo trenutni položaj, smo za nadaljnje teste uporabili vektorje meritev jakosti signalov, ki imajo dolžino 5 (to pomeni, da vektor $\mathbf{RSSI}_M$ vsebuje meritve iz petih različnih oddajnikov). Namreč izkazalo se je, da se pri upoštevanju več kot petih meritev iz različnih oddajnikov natančnost lokalizacije več ne izboljšuje.

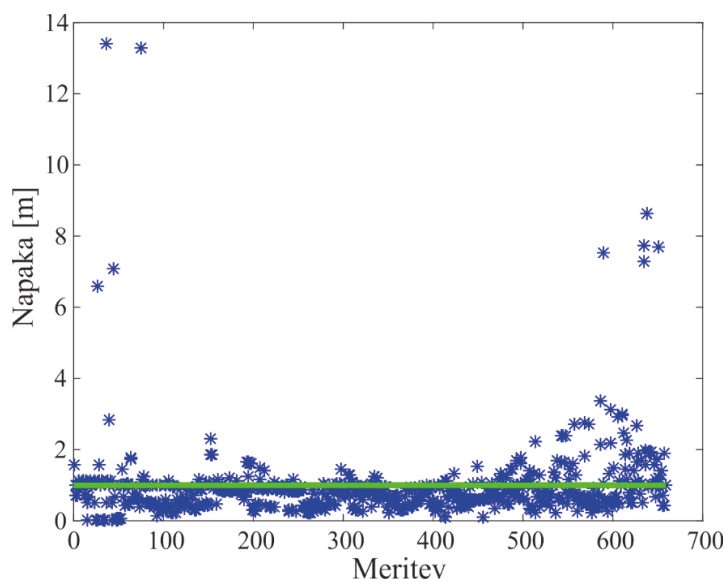
V naslednjem testu lokalizacijskega algoritma, katerega rezultat vidimo na sliki 6.80, smo položaje določili z iskanjem enega najbližjega sosesa ($K = 1$) znotraj podatkovne baze prstnih odtisov. Dolžine zelenih črt na sliki 6.80 prikazujejo razdalje med dejanskimi položaji, kjer je bil kreiran posamezen vektor $\mathbf{RSSI}_M$, in položaji, ki smo jih dobili z iskanjem najbližjega sosesa znotraj podatkovne baze. Na sliki 6.81 vidimo, da so v tem primeru položaji veliko bolj natančno določeni (večina napak je manjših od 1 m) kot v primeru, ko so upoštevane le tri meritve jakosti signalov.

Z namenom, da bi ugotovili, kako število upoštevanih najbližjih sosesov vpliva na natančnost lokalizacije, smo izvedli simulacijo, pri kateri smo spreminjali parameter K od 1 do 40 in določali položaje po enačbi (6.108) za vse meritve opravljene na poti, ki je prikazana na sliki 6.77. Natančnost lokalizacije smo v tem primeru ovrednotili z vsoto napak za celotno pot (prikazano na sliki 6.77). Iz rezultatov simulacije, ki so prikazani na sliki 6.82, lahko vidimo, da je natančnost lokalizacije najboljša pri upoštevanju 23-ih najbližjih sosesov. To pomeni, da v primeru, če bi bili vsi najbližji

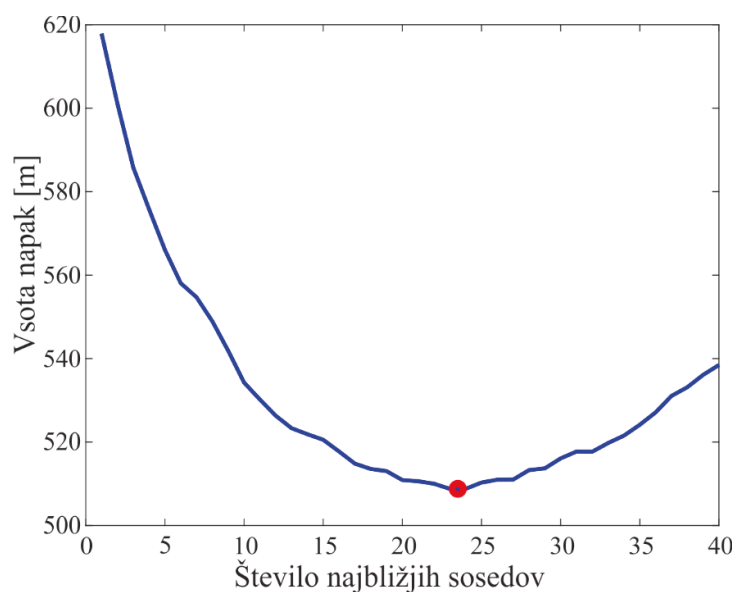
sosejje enakomerno razporejeni okoli točke, kjer so bile zajete meritve jakosti signalov, bi bilo področje najbližjih sosedov veliko približno 2×2 m (mreža zemljevida prstnih odtisov ima okenca velikosti $0,5 \times 0,5$ m). Glede na to velikost področja (najbližjih sosedov) pa bi lahko zaključili, da so napake določanja položaja v povprečju manjše od 1 m.



Slika 6.80: Določanje položaja v prostoru pri uporabi meritev jakosti signalov iz petih različnih oddajnikov (dolžina vektorja $RSSI_M$ je 5) in upoštevanju enega najbližjega soseda

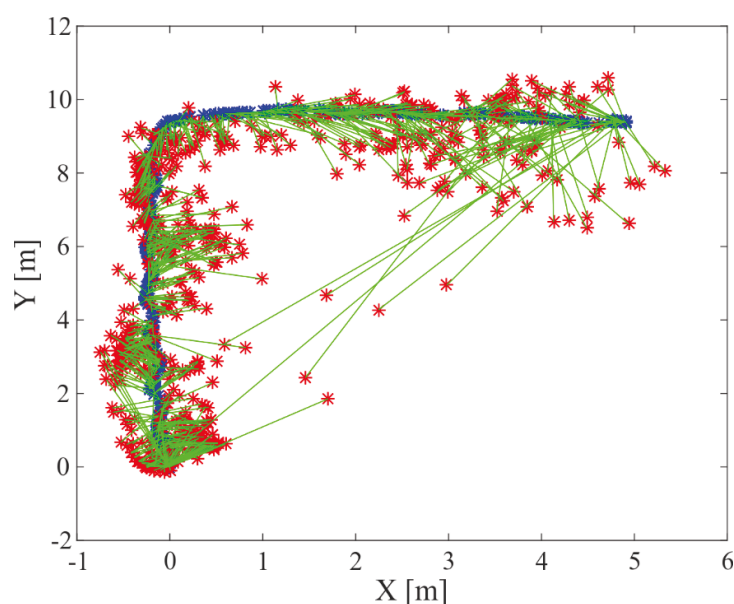


Slika 6.81: Napake (razdalje med izračunanim položajem in dejanskim položajem) pri določanju položaja v primeru, ko so upoštevane meritve jakosti signalov iz petih različnih oddajnikov in je $K=1$

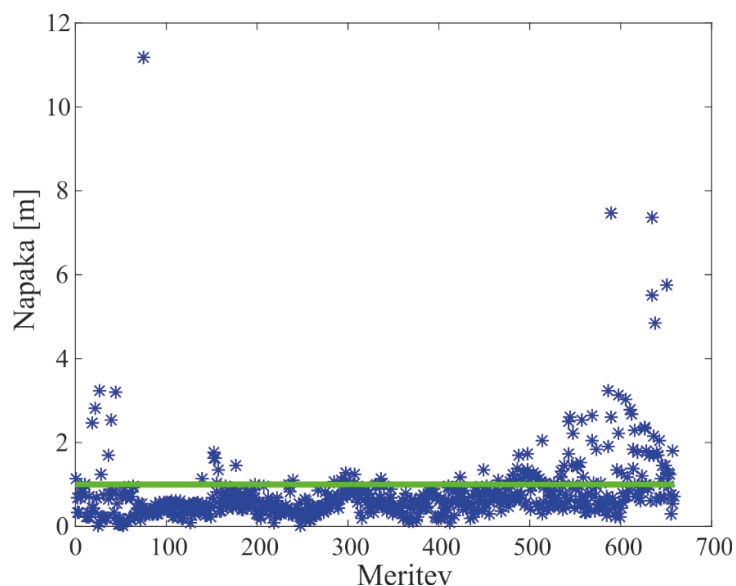


Slika 6.82: Vpliv parametra K (število najbližjih sosedov) na natančnost lokalizacije

Na sliki 6.83 je prikazan rezultat določanja položaja v prostoru pri uporabi meritev jakosti signalov iz petih različnih oddajnikov in upoštevanju optimalnega števila najbližjih sosedov ($K = 23$). Vidimo lahko, da se pri upoštevanju več najbližjih sosedov natančnost določanja položaja v veliki meri izboljša, saj vsota napak pade iz 618 m (pri $K = 1$) na 509 m (pri $K = 23$). Vse napake (za primer, ko je $K = 23$) so prikazane na sliki 6.84.



Slika 6.83: Določanje položaja v prostoru pri uporabi meritev jakosti signalov iz petih različnih oddajnikov in upoštevanju optimalnega števila najbližjih sosedov ($K=23$)

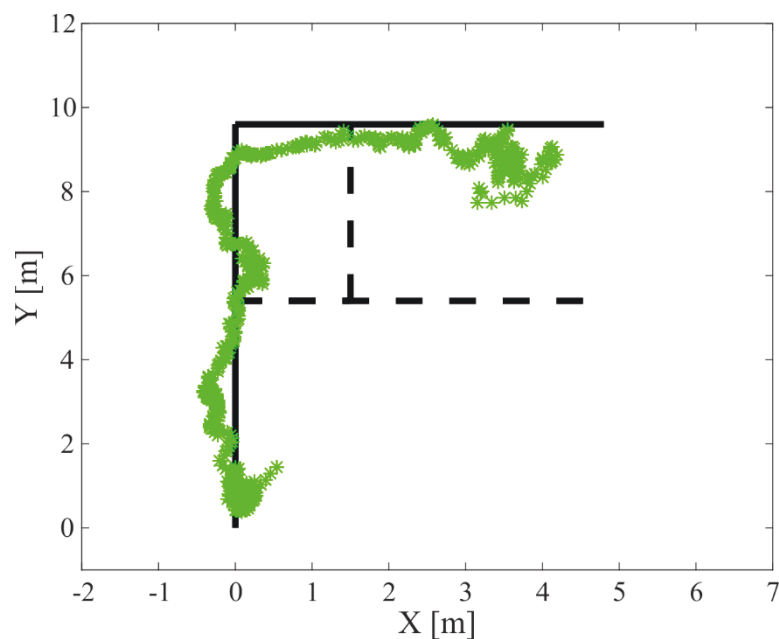


Slika 6.84: Napake (razdalje med izračunanim položajem in dejanskim položajem) pri določanju položaja v primeru, ko so upoštevane meritve jakosti signalov iz petih različnih oddajnikov in je $K=23$

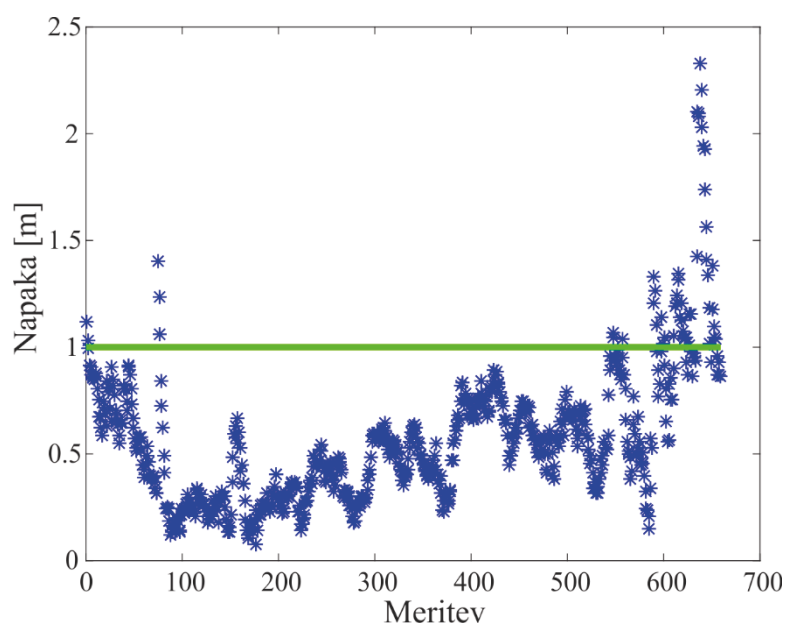
Uporaba nizkopasovnega filtra za glajenje rezultatov

Pri določanju položajev (z uporabo podatkovne baze prstnih odtisov) glede na meritve jakosti signalov zajete po poti, ki je prikazana na sliki 6.77, lahko opazimo, da dobljene točke delajo velike skoke od enega vektorja meritev do naslednjega. Ker vemo, da morajo biti pripadajoči položaji sosednjih vektorjev meritev jakosti signalov blizu skupaj, smo uporabili nizkopasovni filter za x in y koordinati točk: $x_{LP}(k) = 0,9x_{LP}(k-1) + 0,1x(k)$ in $y_{LP}(k) = 0,9y_{LP}(k-1) + 0,1y(k)$. Na sliki 6.85 vidimo, da z uporabo nizkopasovnega filtra dobimo veliko bolj gladko pot in tudi napake pri določanju položaja so veliko manjše.

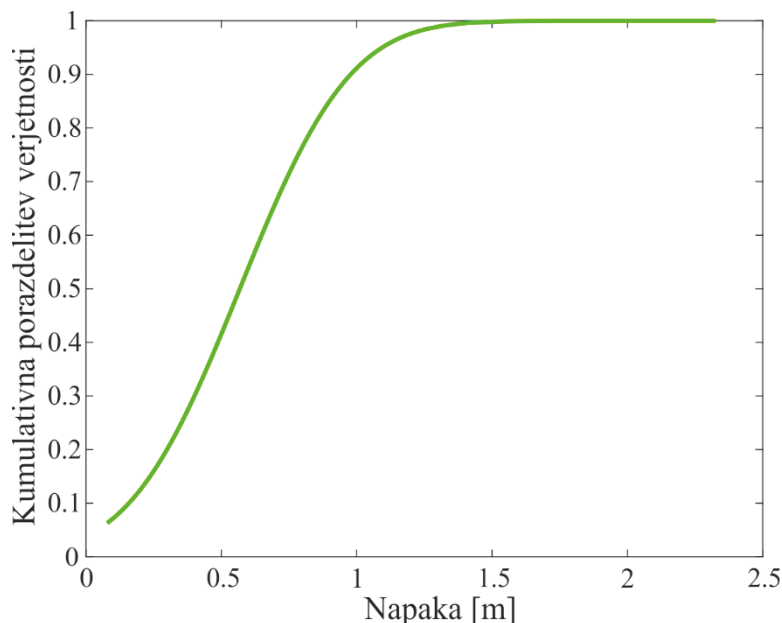
Na sliki 6.86 so prikazane napake (razdalje med izračunanim položajem in dejanskim položajem) pri določanju položaja v primeru, ko uporabimo nizkopasovni filter za obe koordinati izračunanih položajev (točk) v prostoru. Opazimo lahko, da je večina napak manjših od 1 m. Za podrobnejšo analizo natančnosti lokalizacijskega algoritma smo na sliki 6.87 prikazali še kumulativno porazdelitev verjetnosti napake. Ta nam pove, da je 92 % časa napaka manjša od 1 m oz. 42 % časa napaka manjša od 0,5 m.



Slika 6.85: Določanje položaja v prostoru pri uporabi meritev jakosti signalov iz petih različnih oddajnikov, upoštevanju optimalnega števila najbližjih sosedov ($K=23$) in uporabi nizkopasovnega filtra za glajenje poti



Slika 6.86: Napake pri določanju položaja v primeru, ko so upoštevane meritve jakosti signalov iz petih različnih oddajnikov, $K=23$ in je uporabljen nizkopasovni filter



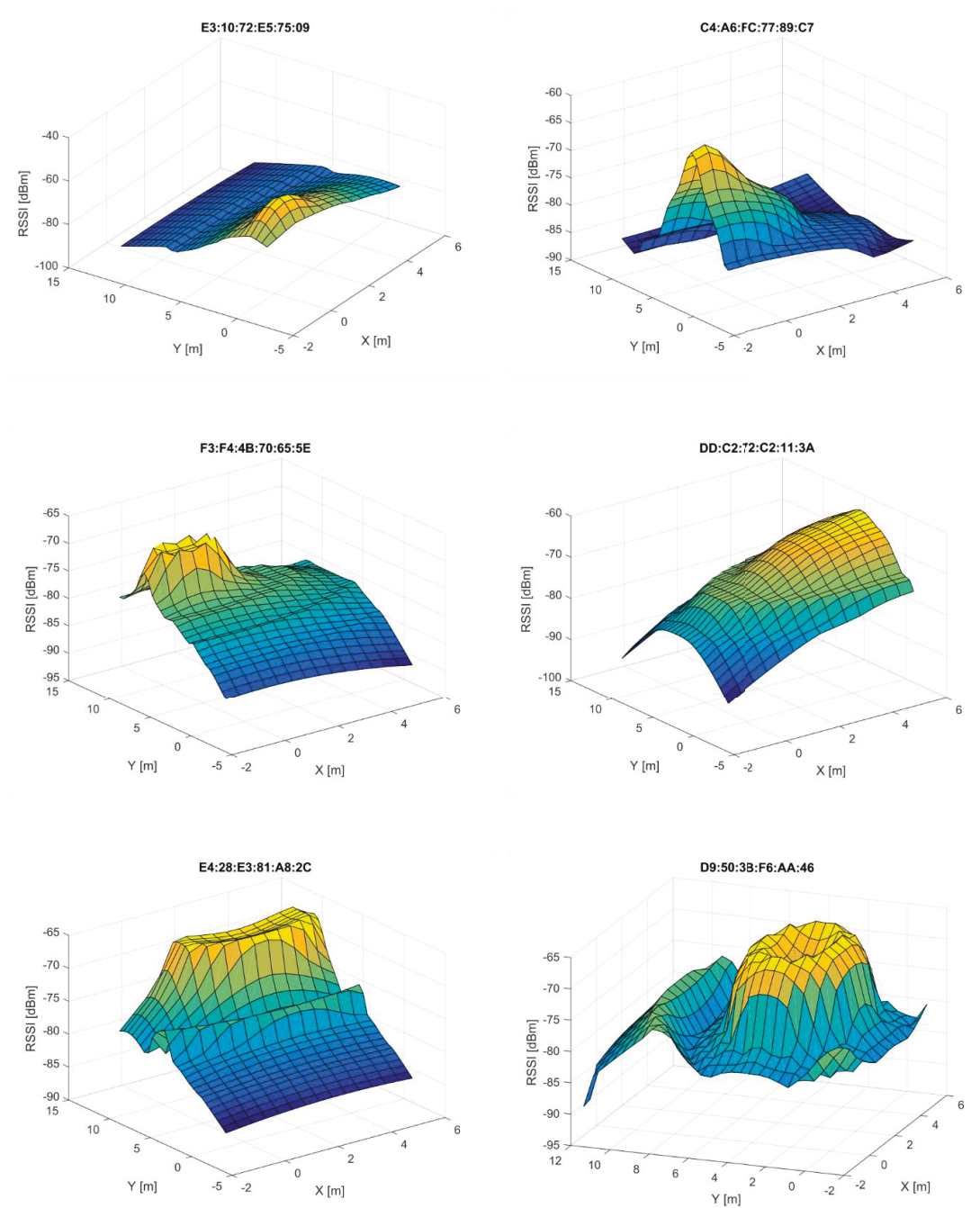
Slika 6.87: Kumulativna porazdelitev napake v primeru, ko je uporabljen nizkopasovni filter za glajenje poti (92 % časa je napaka manjša od 1 m)

Uporaba mehkih modelov za izgradnjo zemljevida prstnih odtisov

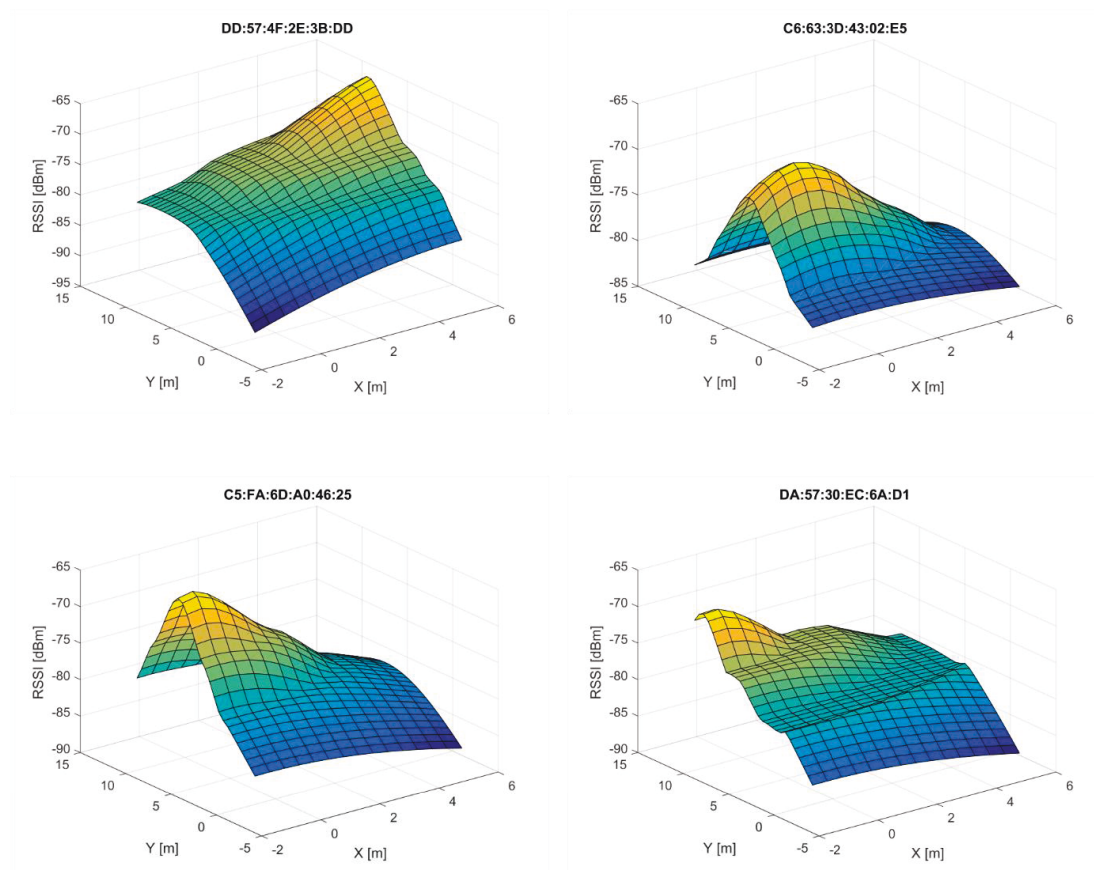
Zaradi ovir (stene, omare, pregrade itd.), ki se pri gibanju po zaprtem prostoru nahajajo med oddajniki in sprejemnikom, se padanje jakosti signalov z razdaljo ne da natančno opisati z nelinearnimi modeli (6.2). Zato smo širjenje signalov po prostoru opisali z mehкими identifikacijskimi modeli, ki so bili definirani z uporabo algoritma SUHICLUST. Na ta način smo dobili bolj sofisticirane ploskve, ki bolj natančno opisujejo jakosti signalov za vse oddajnike v prostoru (sliki 6.88 in 6.89). Ploskve, ki so prikazane na slikah 6.88 in 6.89, imajo zelo različne oblike, na kar vpliva položaj oddajnika in razporeditev ovir v prostoru.

Z uporabo mehkih modelov smo generirali mrežo (z velikostjo okenc $0,5 \times 0,5$ m) prstnih odtisov $RSSI_B$ (kot je prikazano na sliki 6.64) in jih shranili v podatkovno bazo. V okolju Matlab smo uporabili iste meritve jakosti signalov kot v prejšnjih eksperimentih in znotraj podatkovne baze poiskali K najbližjih sosedov za vse vektorje $RSSI_M$ dolžine 5. Iz položajev najbližjih sosedov smo po enačbi (6.108) izračunali trenutne položaje v prostoru. Izračunani položaji glede na meritve jakosti signalov po celotni poti (prikazani na sliki 6.77) so prikazani na sliki 6.90. Pri izračunu položaja je

bilo upoštevanih $K = 23$ najbližjih sosedov, saj v tem primeru dobimo najbolj natančno določene položaje (glede na sliko 6.82).



Slika 6.88: Prikaz širjenja Bluetooth signalov po prostoru na podlagi mehkih modelov

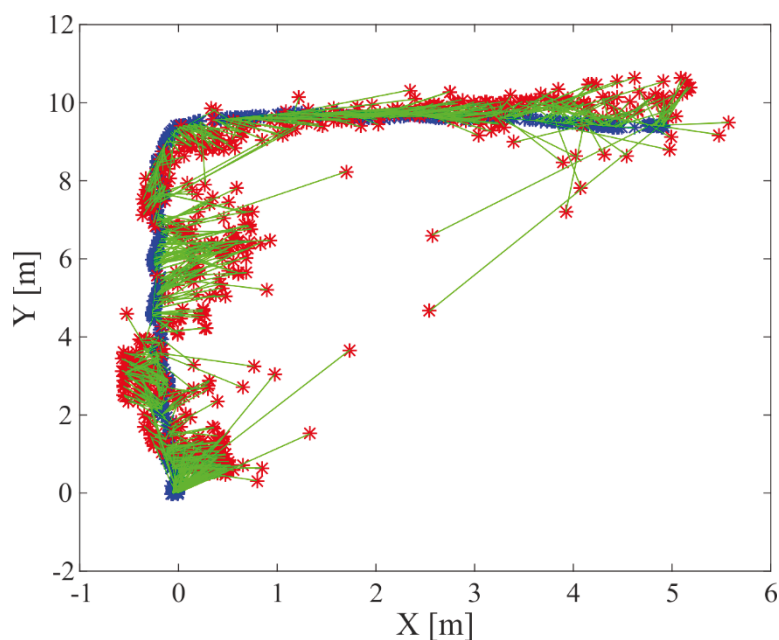


Slika 6.89: Prikaz širjenja Bluetooth signalov po prostoru na podlagi mehkih modelov

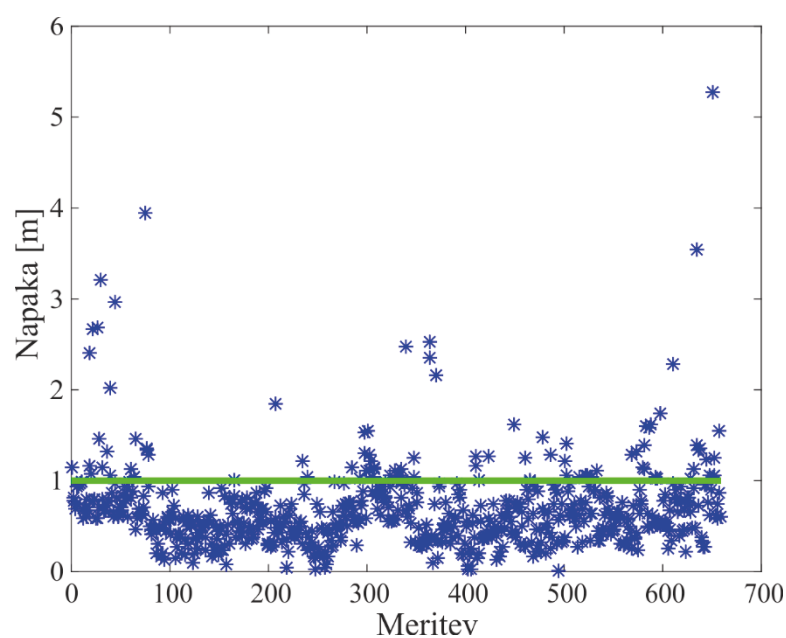
Na sliki 6.90 lahko opazimo, da so pri uporabi prstnih odtisov, ki so generirani z uporabo mehkih modelov, položaji bolj natančno določeni kot v primeru, ko smo prstne odtise generirali z uporabo običajnih nelinearnih modelov (opisani s splošno nelinearno enačbo). Na sliki 6.91, kjer so prikazane napake pri določanju položaja z uporabo prstnih odtisov, ki so bili generirani na podlagi mehkih modelov, vidimo, da je večina napak manjših od 1 m in tudi maksimalna napaka je za polovico manjša kot pri uporabi nelinearnih modelov (slika 6.84).

Z namenom, da bi izračunane položaje (prikazane na sliki 6.90) povezali in dobili bolj gladko pot, smo uporabili nizkopasovni filter za obe koordinati točk: $x_{LP}(k) = 0,9x_{LP}(k-1) + 0,1x(k)$ in $y_{LP}(k) = 0,9y_{LP}(k-1) + 0,1y(k)$. Rezultat glajenja poti je prikazan na sliki 6.92, kjer lahko vidimo, da so se napake (določanja položaja) z uporabo nizkopasovnega filtra še dodatno zmanjšale. Na sliki 6.93, kjer so prikazane velikosti napak, lahko opazimo, da je večina napak manjših od 1 m

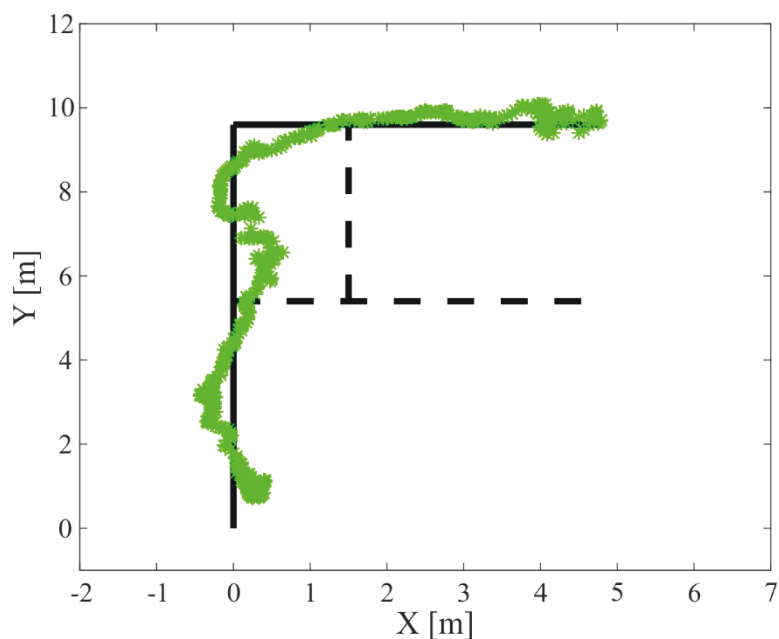
(povprečje napak je 0,5 m). Pri uporabi nizkopasovnega filtra je vsota napak enaka 206,46 m medtem ko je bila pri uporabi nelinearnih modelov ta enaka 280,09 m.



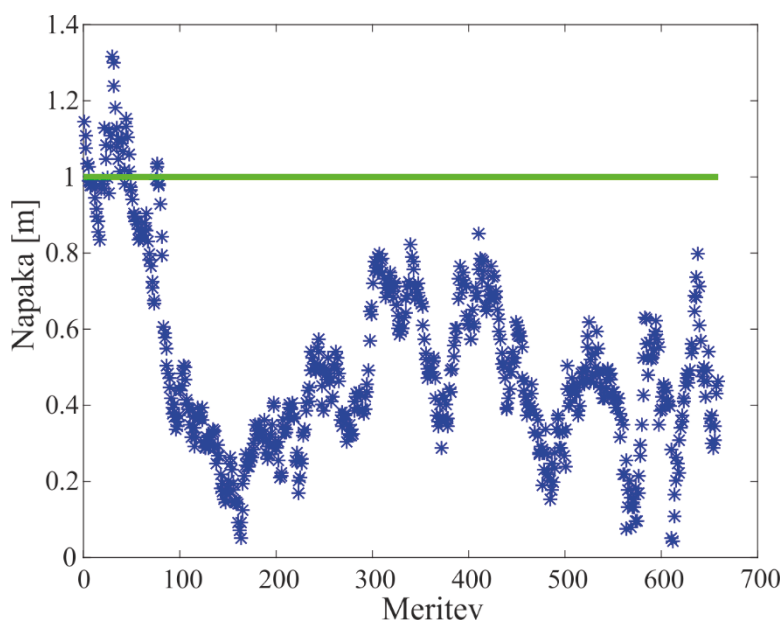
Slika 6.90: Določanje položaja v prostoru pri uporabi prstnih odtisov, ki so bili generirani na podlagi mehkih modelov (dolžina vektorjev meritev $RSSI_M$ je 5 in $K=23$)



Slika 6.91: Napake pri določanju položaja z uporabo prstnih odtisov, ki so bili generirani na podlagi mehkih modelov (dolžina vektorjev meritev $RSSI_M$ je 5 in $K=23$)



Slika 6.92: Določanje položaja v prostoru pri uporabi prstnih odtisov, ki so bili generirani na podlagi mehkih modelov (dolžina vektorjev meritev $RSSI_M$ je 5 in $K=23$) in upoštevanju nizkopasovnega filtra za glajenje poti

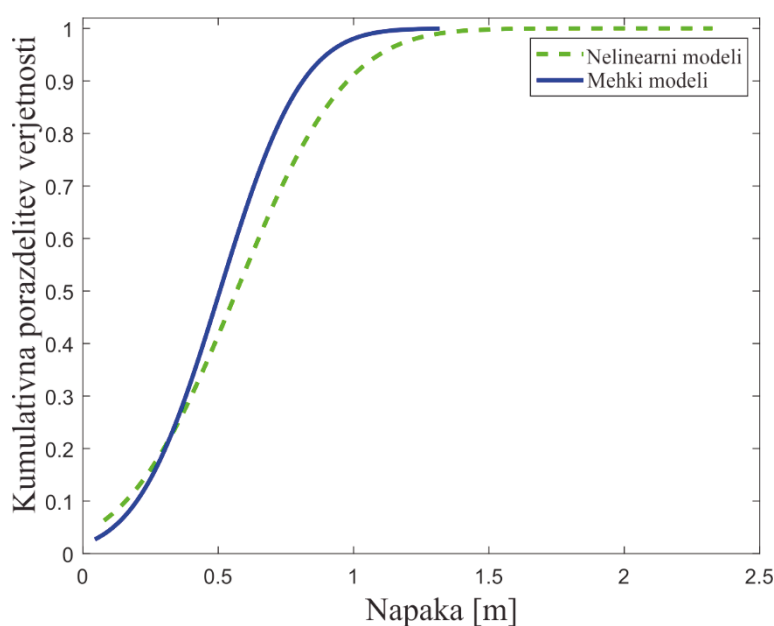


Slika 6.93: Napake pri določanju položaja z uporabo prstnih odtisov, ki so bili generirani na podlagi mehkih modelov (dolžina vektorjev meritev $RSSI_M$ je 5 in $K=23$). Za glajenje poti je bil uporabljen nizkopasovni filter.

Primerjavo natančnosti lokalizacije pri uporabi splošnih nelinearnih modelov in mehkih modelov za generiranje baze prstnih odtisov lahko naredimo z grafom kumulativnih porazdelitev verjetnosti napak, ki je prikazan na sliki 6.94. Opazimo

lahko, da ima modra krivulja (dobljena na podlagi napak pri določanju položajev z uporabo mehkih modelov) večji naklon kot zelena krivulja (dobljena na podlagi napak pri določanju položajev z uporabo nelinearnih modelov), kar pomeni, da je več napak zbranih pri nižjih vrednostih kot pri zeleni krivulji.

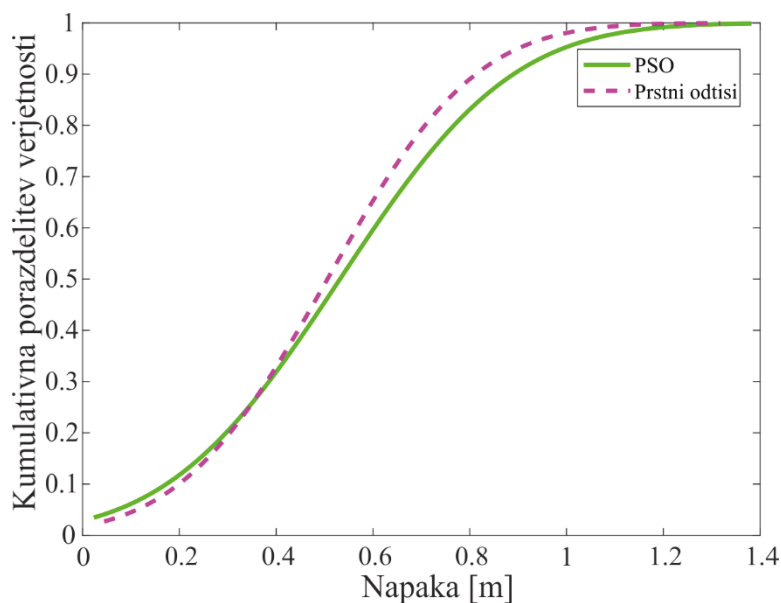
Iz slike 6.94 lahko preberemo, da je pri uporabi mehkih modelov kar v 98 % ocen položajev napaka manjša od 1 m oz. v 49 % ocen manjša od 0,5 m. Pri uporabi nelinearnih modelov pa je bila napaka manjša od 1 m v 92 % ocen položajev oz. manjša od 0,5 m v 42 % ocen položajev.



Slika 6.94: Primerjava kumulativnih porazdelitev verjetnosti napak pri določanju položaja z uporabo prstnih odtisov, ki so bili generirani z nelinearnimi modeli oz. mehki modeli

Z uporabo metode prstnih odtisov, mehkih modelov in dodatnega nizkopasovnega filtra smo dobili najboljše rezultate lokalizacije na osnovi jakosti Bluetooth signalov. Kot druga najboljša metoda pa se je izkazala metoda PSO v kombinaciji z mehki modeli in dodatnim nizkopasovnim filtrom. Slika 6.95 kaže primerjavo kumulativnih porazdelitev verjetnosti napak za ti dve metodi, ki sta se izkazali kot najboljši. Opazimo lahko, da krivulja kumulativne porazdelitve napak, ki pripada metodi prstnih odtisov, nekoliko hitreje doseže vrednost ena kot krivulja, ki pripada metodi PSO. Vsota kvadratov napak je pri metodi prstnih odtisov enaka 206, pri metodi PSO pa 252. Iz grafa na sliki 6.95 lahko preberemo, da je pri metodi prstnih odtisov v 98 %

ocenjenih položajev napaka manjša od 1 m, pri metodi PSO pa to drži v 95 % ocen. Čeprav se metodi z vidika natančnosti ne razlikujeta prav veliko, pa moramo omeniti še faktor, ki je ključen pri izbiri algoritma za lokalizacijo (ki mora delovati v realnem času tudi na računsko manj zmogljivi napravi). Metoda PSO je v okolju Matlab kljub minimalnemu številu iteracij (teh mora biti vsaj 10, da so rezultati primerljivi z metodo prstnih odtisov) in uporabi le štirih delcev porabila dvakrat več časa kot metoda prstnih odtisov za izračun posameznega položaja v prostoru.



Slika 6.95: Primerjava kumulativnih porazdelitev verjetnosti napak pri določanju položaja z uporabo metode PSO oz. metode prstnih odtisov (v obeh primerih so bili uporabljeni mehki modeli in dodatno filtriranje z nizkopasovnim filtrom)

Za vse prstne odtise, ki se nahajajo v bazi podatkov, lahko glede na meje mehkih modelov (slika 6.20) določimo območja zaupanja jakosti signalov RSSI vseh oddajnikov (vseh oddajnikov v bazi je $m = 10$). Za vsako vrednost $RSSI_{B_i} = f_i(d_i)$ ($i = 1, \dots, m$), ki se nahaja znotraj posameznega prstnega odtisa $RSSI_B$, lahko glede na spodnjo mehko mejo $\underline{f}_i(d_i)$ in zgornjo mehko mejo $\overline{f}_i(d_i)$ i -tega mehkega modela določimo širino območja zaupanja $\omega_{pi} = \overline{f}_i(d_i) - \underline{f}_i(d_i)$ pri izbranem vhodu d_i (to je razdalja med položajem i -tega oddajnika in položajem prstnega odtisa). Na ta način vsakemu prstnemu odtisu, ki je shranjen v bazi podatkov, dodelimo dodaten vektor $\omega_p = [\omega_{p1}, \dots, \omega_{pm}]^T$, ki ima enako dolžino kot vektor prstnega odtisa in vsebuje informacije o širini območja zaupanja za posamezno jakost signala $RSSI_{B_i}$ v točki

prstnega odtisa. Na podlagi širine območja zaupanja ω_{pi} lahko izračunamo uteži za vse elemente $RSSI_{B_i}$ znotraj prstnega odtisa: $\mathbf{w} = [w_1, \dots, w_m]^T$, kjer je $w_i = 1/\omega_{pi}$ ($i = 1, \dots, m$) in jih upoštevamo pri izračunu razdalje l_w (v postopku iskanja K najbližjih sosedov) med posameznim prstnim odtisom $RSSI_B$ in trenutnim vektorjem meritev jakosti signalov $RSSI_M$:

$$l_w = \sqrt{\sum_{i=1}^n w_i (RSSI_{B_i} - RSSI_{M_i})^2} \text{ (tj. utežena evklidska razdalja, kjer je } n \text{ število}$$

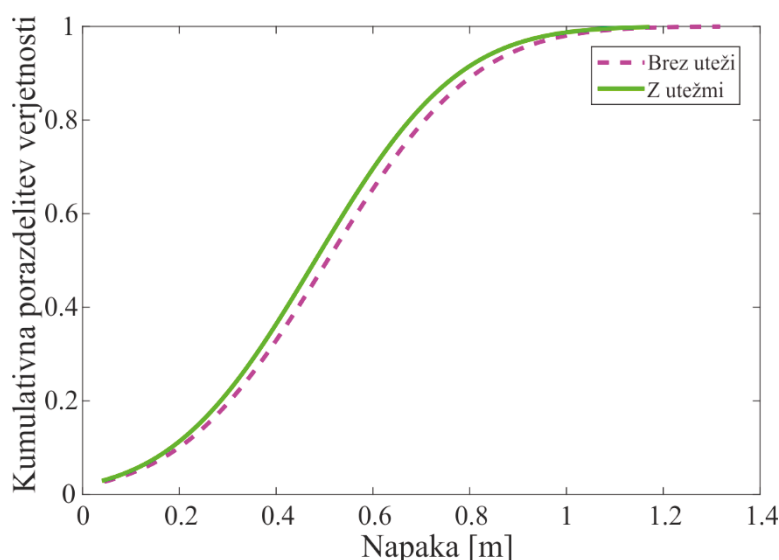
id-jev oddajnikov, ki so prisotni v obeh vektorjih). V tem primeru smo torej imeli v bazi podatkov (tj. JSON datoteka) poleg prstnih odtisov shranjene tudi pripadajoče uteži $\mathbf{w}$. Razlike $(RSSI_{B_i} - RSSI_{M_i})$ imajo večji prispevek k razdalji l_w , če je območje zaupanja ω_{pi} ožje, saj je s tem utež w_i večja (meritev $RSSI_{M_i}$ ima večji vpliv).

V naslednjem preizkusu smo torej želeli preveriti, ali lahko napako lokalizacije še dodatno zmanjšamo, če upoštevamo uteži w_i pri izračunu razdalj l_w med vektorji $RSSI_B$ in $RSSI_M$ v postopku iskanja K najbližjih sosedov. Vse ostale nastavitve in parametri so ostali enaki kot pri prejšnjem preizkusu lokalizacije s prstnimi odtisi (dolžina vektorjev meritev $RSSI_M$ je 5 in $K = 23$). Pri primerjavi kumulativnih porazdelitev verjetnosti napak, ki sta prikazani na sliki 6.96, lahko vidimo, da z upoštevanjem uteži pri izračunu evklidskih razdalj v postopku iskanja K najbližjih sosedov lahko še nekoliko zmanjšamo napako lokalizacije (vsota kvadratov napak se je zmanjšala iz 206 na 187). V tem primeru smo za 53 % ocen položajev dobili napako manjšo od 0,5 m (brez uteži je bila napaka manjša od 0,5 m pri 49 % ocen) oz. za 99 % ocen položajev napako manjšo od 1 m (brez uteži je bila napaka manjša od 1 m pri 98 % ocen položajev).

6.7 Diskusija

Preden smo se lotili te študije, ki obravnava lokalizacijo na osnovi jakosti Bluetooth signalov, smo iz do tedaj objavljene dotične literature dobili vtis, da je ta pristop k lokalizaciji kljub številnim pomanjkljivostim in težavam od vseh še najbolj primeren v primeru, ko potrebujemo nizkocenovni globalni lokalizacijski sistem v notranjem okolju. Čeprav so prve študije kazale na to, da lahko s tem lokalizacijskim pristopom določimo položaj le na nekaj metrov natančno, pa se je s spoznavanjem Bluetooth

sistema in metod radijske lokalizacije kasneje izkazalo, da je pri uporabi ustreznih algoritmov možno veliko bolje izkoristiti njegov potencial. Tudi zadnji trend naraščanja števila Bluetooth oddajnikov v javnih zgradbah, kot so trgovski centri, letališča, muzeji itd., kaže na to, da se bo ta pristop uveljavil kot nizkocenovna lokalizacija za notranje okolje, tako kot se je GPS uveljavil oz. postal sinonim za lokalizacijo v zunanjem okolju. Namreč Bluetooth oddajniki pri uporabi sofisticiranih algoritmov omogočajo relativno visoko natančnost lokalizacije ob zelo majhni porabi energije, kar je ključno pri napravah z omejenim virom energije, kot so npr. pametni telefoni.



Slika 6.96: Primerjava kumulativnih porazdelitev verjetnosti napak pri določanju položaja z uporabo metode prstnih odtisov. V prvem primeru (vijolična krivulja) je bila v postopku iskanja najbližjih sosedov uporabljena evklidska razdalja, v drugem primeru (zelena krivulja) pa utežena evklidska razdalja.

Iz rezultatov naše študije Bluetooth lokalizacijskega sistema smo ugotovili, da so za dosego visoke natančnosti lokalizacije ključni natančni modeli širjenja signalov, za izgradnjo katerih pa je potreben dovolj velik nabor meritev jakosti signalov s podanim položajem, kjer so bile meritve zajete. Za obsežen zajem meritev je bila ključna uporaba natančnega relativnega lokalizacijskega sistema, ki združuje vizualno odometrijo in inercialni navigacijski sistem. Z uporabo tega sistema smo namreč lahko enostavno določili položaje, kjer so bile meritve jakosti signalov zajete. Modele jakosti signalov smo izboljšali z uporabo algoritma SUHICLUST, ki model sestavi iz več

lokalnih modelov, kar pomeni, da so jakosti signalov pri vseh oddaljenostih od oddajnika bolj natančno opisane ne glede na ovire, ki povzročajo odboje in absorpcije signalov. Za zmanjšanje odbojev in absorpcij signalov (predvsem zaradi dinamičnih ovir, tj. prisotnost ljudi v prostoru) pa je pri nameščanju oddajnikov v prostor potrebno upoštevati določena priporočila glede postavitve oddajnikov (priporočena je enakomerna razporeditev ter dovolj velika gostota oddajnikov in pa ustrezna višina pritrditve). Glede na gostoto oddajnikov v prostoru pa je potrebno znotraj lokalizacijskega algoritma prilagoditi tudi število upoštevanih oddajnikov pri določanju posameznega položaja. Za natančno določanje položaja je namreč pomembno, da upoštevamo nazadnje osvežene meritve iz oddajnikov, ki so sprejemniku dovolj blizu.

Pri preizkusu različnih lokalizacijskih metod, ki za svoje delovanje uporabljajo jakosti signalov, se je izkazalo, da z uporabo trilateracije dobimo najmanj natančne rezultate lokalizacije. Če trilateracijo uporabljamo v kombinaciji z omejeno nelinearno optimizacijo, pa je algoritem tudi računsko zahteven. Pri uporabi metode PSO smo dobili veliko boljše rezultate lokalizacije, vendar so ti lahko suboptimalni (zaradi metahevrističnega pristopa). Na delovanje metode imajo velik vpliv parametri (predvsem je pomembna izbira velikosti iskalnega območja), ki morajo biti ustrezno določeni. Med vsemi preizkušeni metodami se je metoda prstnih odtisov izkazala za najbolj robustno, natančno, računsko nezahtevno in v kombinaciji s podatkovnimi drevesi tudi za najhitrejšo. Pri naši implementaciji algoritma, ki temelji na uporabi prstnih odtisov, je bil ključen napredek narejen v prvi fazi, tj. pri gradnji zemljevida prstnih odtisov. Zemljevid smo namreč zgradili z uporabo natančnih modelov, kar predstavlja prednost v smislu hitrejše gradnje zemljevida in možnost spreminjanja gostote prstnih odtisov. Pri določanju položaja (pri postopku iskanja najbližjih sosedov) pa je bistveno, da upoštevamo več uteženih najbližjih sosedov, saj s tem zmanjšamo vpliv šuma in posledično povečamo natančnost lokalizacije.

Zaključimo lahko, da smo pri tej študiji dosegli zastavljeni cilj, tj. natančnost lokalizacije na 1 m, saj smo z uporabo metode prstnih odtisov in mehkih modelov dobili rezultat, pri katerem je bila napaka manjša od enega metra kar v 99 % ocenjenih položajev.

7 Fuzija vizualne odometrije, inercialnega navigacijskega sistema in Bluetooth lokalizacije

V tem poglavju obravnavamo fuzijo vizualne odometrije, inercialnega navigacijskega sistema ter Bluetooth lokalizacije z uporabo Kalmanovega filtra in modela konstantne hitrosti. V nadaljevanju sta naprej opisana Kalmanov filter in model konstantne hitrosti, zatem sam postopek fuzije lokalizacijskih metod in na koncu še rezultati preizkusa novega lokalizacijskega sistema na pametnem telefonu.

7.1 Kalmanov filter

Kalmanov filter, ki se uporablja za ocenjevanje parametrov (ali stanj) linearnega dinamičnega sistema, sta razvila R. Kalman in R. Bucy [253, 254] sredi šestdesetih let prejšnjega stoletja. Kalmanov filter predstavlja optimalni opazovalnik stanj, saj so ocene optimalne v smislu minimizacije pričakovane srednje kvadratne napake med oceno stanja in dejanskim stanjem [255].

Pri Kalmanovem filtru obravnavamo splošen problem optimalnega ocenjevanja n -dimenzionalnega vektorja stanj $\hat{\mathbf{x}} \in \mathbb{R}^n$ časovno diskretnega stohastičnega sistema. Vektor stanj se ocenjuje le na podlagi meritev $\mathbf{z} \in \mathbb{R}^m$, za katere predpostavimo, da so motene z Gaussovim belim šumom.

Algoritem Kalmanovega filtra je sestavljen iz dveh faz, in sicer faze predikcije (ang. predict) in faze korekcije (ang. update):

Predikcija:

V fazi predikcije se najprej izračuna napoved vektorja stanj kot:

$$\hat{\mathbf{x}}_k^- = \mathbf{F}_k \hat{\mathbf{x}}_{k-1} + \mathbf{G}_k \mathbf{u}_k, \quad (5.1)$$

kjer je $\mathbf{F}_k$ matrika prehajanja stanj, $\mathbf{G}_k$ vhodna matrika, $\mathbf{u}_k$ vektor vhodnih signalov, $\hat{\mathbf{x}}_k^-$ a priori ocena stanj tik pred meritvijo ter $\hat{\mathbf{x}}_{k-1}$ a posteriori (trenutna) ocena stanj v trenutku $k - 1$. A priori (napovedano) oceno kovariance napake predikcije izračunamo kot:

$$\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k, \quad (7.2)$$

kjer $\mathbf{P}_{k-1}$ predstavlja a posteriori kovariančno matriko napake (ki nam pove, kako natančno so stanja ocenjena v trenutku $k - 1$) in $\mathbf{Q}_k$ kovariančno matriko šuma procesa.

Korekcija:

V fazi korekcije je potrebno najprej izračunati vektor $\mathbf{y}_k$, ki ga imenujemo tudi merilna inovacija (ang. innovation):

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-, \quad (7.3)$$

kjer je $\mathbf{H}_k$ izhodna matrika. Nato v drugem koraku korekcije lahko izračunamo a posteriori oceno stanj $\hat{\mathbf{x}}_k$ po meritvi $\mathbf{z}_k$:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{y}_k, \quad (7.4)$$

kjer matrika $\mathbf{K}_k$ predstavlja Kalmanovo ojačanje, ki pove, v kolikšni meri naj se upošteva merilna inovacija $\mathbf{y}_k$. Kalmanovo ojačanje izračunamo z uporabo a priori ocene kovariačne matrike $\mathbf{P}_k^-$ ter kovariančne matrike inovacije $\mathbf{S}_k = \text{cov}(\mathbf{y}_k)$ kot:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{S}_k^{-1}, \quad (7.5)$$

kjer je kovariančna matrika inovacije enaka $\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k$. V slednji enačbi $\mathbf{R}_k$ predstavlja kovariančno matriko merilnega šuma. Za ojačanje $\mathbf{K}_k$ velja, da pada s kovariančno matriko $\mathbf{R}_k$ in narašča s kovariančno matriko $\mathbf{Q}_k$.

V zadnjem koraku korekcije je potrebno izračunati še a posteriori kovariančno matriko napake a posteriori ocene $\mathbf{P}_k = E[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T]$ kot:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-, \quad (7.6)$$

kjer je $\mathbf{I}$ enotska matrika (v našem primeru velikosti 4×4).

7.1.1 Model konstantne hitrosti

Imamo linearni dinamični sistem, tj. model konstantne hitrosti [255, 256] podan v prostoru stanj:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\tilde{\mathbf{w}}(t), \quad (7.7)$$

$$\mathbf{z}(t) = \mathbf{H}\mathbf{x}(t) + \tilde{\mathbf{v}}(t), \quad (7.8)$$

$$\text{kjer so: } \mathbf{x}(t) = [x \ y \ \dot{x} \ \dot{y}]^T, \mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \text{ in } \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

ter $\tilde{\mathbf{w}}(t) = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix}$. Vektor stanj $\mathbf{x}(t)$ je sestavljen iz trenutnega položaja (x, y) in hitrosti v smeri x in y .

Pri gibanju s konstantno hitrostjo bi moralo veljati, da je pospešek enak nič: $\ddot{x} = 0$ in $\ddot{y} = 0$. Izkaže se, da na hitrost vplivajo določene motnje, za katere privzamemo, da imajo naključno porazdelitev. Zato nam pospešek (ki je različen od nič) predstavlja motnjo (beli šum s srednjo vrednostjo enako nič) $\tilde{\mathbf{w}}(t) = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix}$.

V diskretnem prostoru lahko model konstantne hitrosti [255, 256] zapišemo kot:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{w}_k \quad \mathbf{w}_k \sim N(0, \mathbf{Q}), \quad (7.9)$$

$$\begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} + \begin{bmatrix} w_{k,1} \\ w_{k,2} \\ w_{k,3} \\ w_{k,4} \end{bmatrix}, \quad (7.10)$$

kjer je $\mathbf{x}_k = [x_k \ y_k \ \dot{x}_k \ \dot{y}_k]^T$ vektor stanj, Δt časovni interval med korakoma k in $k - 1$ in $\mathbf{F}_k$ matrika prehajanja stanj:

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (7.11)$$

Vektor $\mathbf{w}_k = [w_{k,1} \ w_{k,2} \ w_{k,3} \ w_{k,4}]^T$ predstavlja (sistemski) Gaussov beli šum s srednjo vrednostjo nič in kovariančno matriko $\mathbf{Q}_n$ ($\tilde{q}$ je gostota spektralne moči, izpeljava je v dodatku B):

$$\mathbf{Q}_n = \tilde{q} \begin{bmatrix} \frac{(\Delta t)^3}{3} & 0 & \frac{(\Delta t)^2}{2} & 0 \\ 0 & \frac{(\Delta t)^3}{3} & 0 & \frac{(\Delta t)^2}{2} \\ \frac{(\Delta t)^2}{2} & 0 & \Delta t & 0 \\ 0 & \frac{(\Delta t)^2}{2} & 0 & \Delta t \end{bmatrix}. \quad (7.12)$$

Izhod modela $\mathbf{z}_k$ v našem primeru določa položaj (x_k, y_k) , ki ga dobimo iz vektorja stanj:

$$\mathbf{z}_k = [x_k \ y_k]^T = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad \mathbf{v}_k \sim N(0, \mathbf{R}), \quad (7.13)$$

kjer je $\mathbf{H}_k$ izhodna matrika:

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \quad (7.14)$$

Vektor $\mathbf{v}_k = [v_{k,1} \ v_{k,2}]^T$ predstavlja (merilni) Gaussov beli šum s srednjo vrednostjo nič in kovariančno matriko $\mathbf{R}_k$ (σ_v predstavlja standardno deviacijo šuma):

$$\mathbf{R}_k = \sigma_v^2 \mathbf{I}, \quad (7.15)$$

kjer je $\mathbf{I}$ enotska matrika dimenzije $q \times q$ (q je dolžina izhodnega vektorja oz. vektorja meritev $\mathbf{z}_k$).

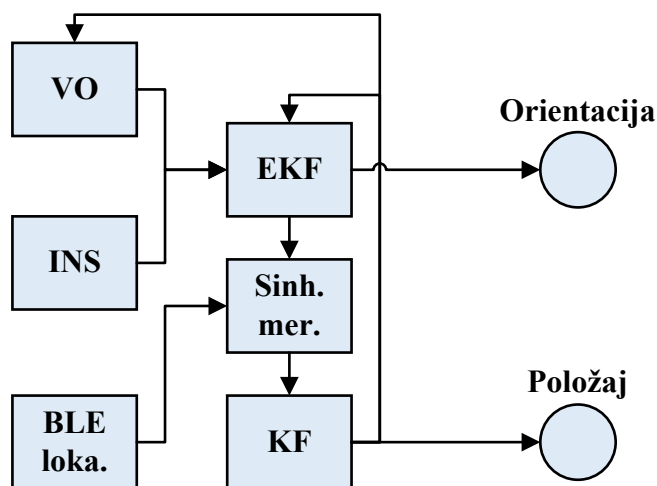
7.2 Postopek fuzije lokalizacijskih metod

Tekom razvoja in testiranja lokalizacijskih algoritmov, ki temeljijo na vizualni odometriji, inercialnem navigacijskem sistemu in radijski (Bluetooth oz. BLE) lokalizaciji, smo ugotovili, da pri izbranih pogojih dosežemo (zadovoljivo) natančno določanje položajev že pri uporabi enega lokalizacijskega pristopa. Vendar ker ima vsak pristop določene pomanjkljivosti, ki jih želimo odpraviti, smo se lotili združevanja (fuzije) vseh razvitih lokalizacijskih metod, saj želimo, da bi naš lokalizacijski sistem deloval v praksi zanesljivo pri najrazličnejših pogojih skozi daljše časovno obdobje (oz. časovno neodvisno). Skupaj z vizualno odometrijo in inercialnim navigacijskim sistemom smo razvili tudi njuno fuzijo na osnovi razširjenega Kalmanovega filtra, s katero smo dobili natančen relativen lokalizacijski sistem. Z njo lahko natančno določamo inkrementalne premike in omejeno časovno obdobje tudi globalni položaj pri znanem začetnem položaju, vendar se nam pri tem akumulira napaka (s časom postaja vse večja).

Pri preizkušanju razvitega globalnega lokalizacijskega sistema, ki temelji na merjenju jakosti Bluetooth signalov (in metodi najmanjših kvadratov), smo ugotovili, da lahko s tem pristopom relativno natančno (99 % napak je manjših od 1 m) določamo absolutni položaj v prostoru, pri čemer njegova napaka ni časovno odvisna (se s časom ne povečuje), kar je bistveno. Napaka je lahko le krajevno odvisna, kar pomeni, da imamo lahko na enem delu zemljevida prstnih odtisov večjo napako položaja kot na drugem. V nadaljevanju bomo torej pojasnili prednosti združitve natančnega relativnega lokalizacijskega sistema in časovno stabilnega globalnega lokalizacijskega sistema.

Osnovni princip združitve vseh treh lokalizacijskih algoritmov (VO, INS in BLE) je preprost in temelji na uporabi dveh Kalmanovih filtrov, kot lahko vidimo na sliki 7.1. Razširjeni Kalmanov filter skrbi za fuzijo vizualne odometrije in inercialnega navigacijskega sistema in nam vrača inkrementalne premike med posameznimi koraki pešca. Za združitev relativnega in globalnega lokalizacijskega sistema skrbi Kalmanov filter, ki uporablja model konstantne hitrosti, saj lahko predpostavimo, da se pešec giblje s približno konstantno hitrostjo. V predikcijskem delu Kalmanovega filtra upoštevamo informacije o relativnih premikih, ki jih dobimo iz razširjenega Kalmanovega filtra, v korekcijskem delu pa uporabimo rezultat globalne lokalizacije na osnovi merjenja jakosti Bluetooth signalov. Kalmanov filter skrbi le za popravek globalnega položaja, orientacije pa ne spreminja, saj je ta že natančno določena skozi fuzijo vizualne odometrije in digitalnega kompasa.

Na sliki 7.1 vidimo, da je potrebno meritve iz globalnega in relativnega lokalizacijskega sistema medsebojno časovno sinhronizirati še preden jih združimo s Kalmanovim filtrom. Globalni lokalizacijski sistem, ki temelji na merjenju jakosti Bluetooth signalov, ima namreč variabilno zakasnitev, katere velikost je odvisna od dejstva, kako hitro pridobivamo meritve iz Bluetooth oddajnikov.



Slika 7.1: Fuzija vizualne odometrije, inercialnega navigacijskega sistema in Bluetooth lokalizacije

Kot lahko vidimo na sliki 7.1, se izhod iz Kalmanovega filtra (popravljen položaj) upošteva pri vizualni odometriji, saj je potrebno popraviti transformacijo iz ključnega

K.S. (vizualne odometrije) v K.S. zemljevida (Bluetooth oddajnikov in prstnih odtisov), glede na katero se v naslednjem koraku izračuna inkrementalni premik. Nov položaj, ki je bil izračunan s Kalmanovim filtrom, pa se upošteva tudi v razširjenem Kalmanovem filtru, kjer je še pred detekcijo novega koraka potrebno popraviti vektor stanj.

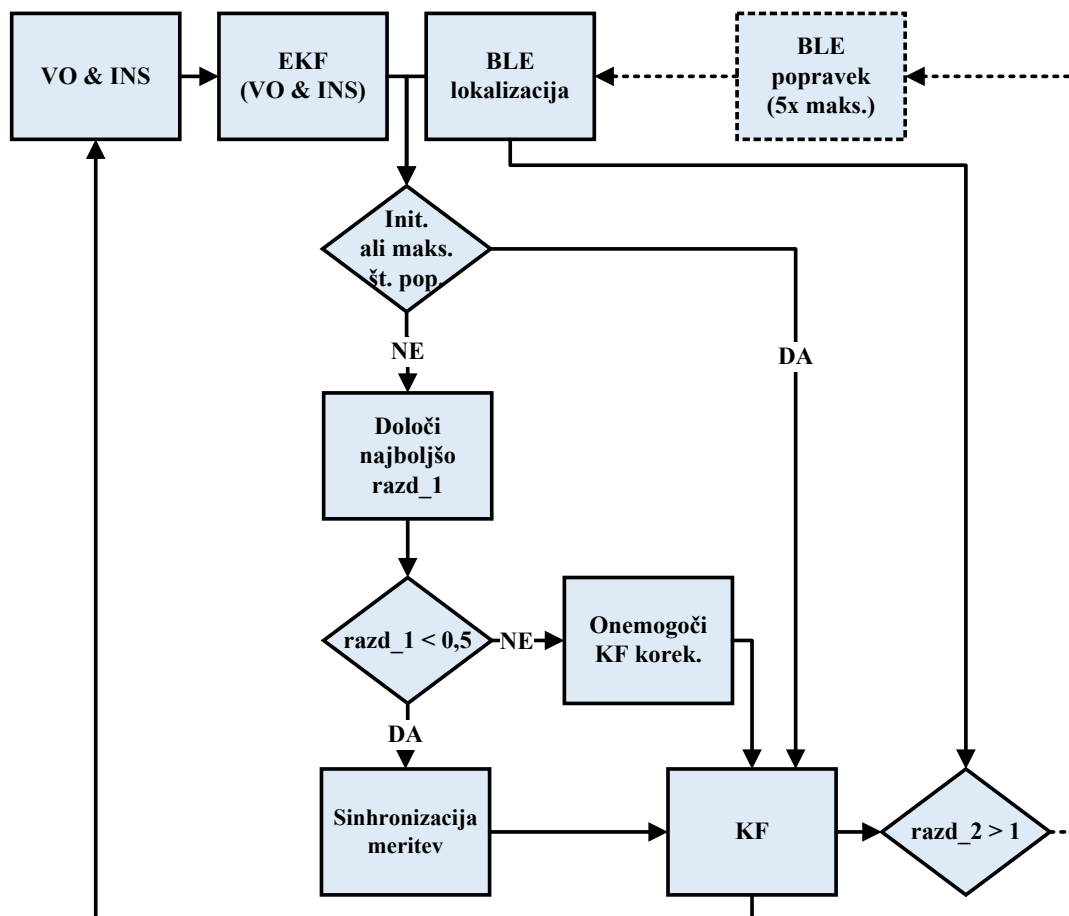
Pri fuziji vseh treh lokalizacijskih algoritmov smo naleteli na nekatere (nepričakovane) težave, ki smo jih rešili z dodatno logiko, ki je prikazana na sliki 7.2. Namreč poleg problema časovne sinhronizacije, smo naleteli tudi na problem, pri katerem Bluetooth lokalizacija na določenem delu zemljevida ne vrača dovolj točnega položaja (problem ni natančnost ampak točnost) in posledično prihaja do večjih odstopanj med napovedjo položaja iz vizualne odometrije in inercialnega navigacijskega sistema ter položajem Bluetooth lokalizacije. V takih primerih Kalmanov filter skoraj več ne upošteva predikcije (relativna lokalizacija se posledično počasi približa položaju, ki ga vrača Bluetooth lokalizacija), kar je običajno posledica slabega modela. Da bi izboljšali natančnost lokalizacije, si želimo v takih primerih kratkotrajno zmanjšati ali celo onemogočiti prispevek Bluetooth lokalizacije. Ker imamo hkrati prisotno še variabilno zakasnitev, se problem fuzije še nekoliko bolj zaplete.

Ob zagonu lokalizacijskega sistema, ki združuje vizualno odometrijo, inercialni navigacijski sistem ter Bluetooth lokalizacijo, se najprej z uporabo metode prstnih odtisov določi začetni globalni položaj in z uporabo digitalnega kompasa začetna orientacija pešca. Določitev začetnega položaja in orientacije je pomembna zato, ker se inkrementalni premiki (na osnovi fuzije vizualne odometrije in inercialnega navigacijskega sistema) pričnejo računati glede na začetni položaj in orientacijo. Ker je začetni položaj določen le približno (le z uporabo Bluetooth lokalizacije), je lahko napoved položaja v fazi predikcije prvih nekaj korakov zelo oddaljena od meritve položaja v fazi korekcije. Zato ob zagonu celotnega lokalizacijskega sistema izklopimo časovno sinhronizacijo podatkov in prvih pet korakov le združujemo rezultate iz razširjenega Kalmanovega filtra in Bluetooth lokalizacije. Na ta način se položaj iz faze predikcije počasi približa položaju iz faze korekcije. Nato lahko omogočimo časovno sinhronizacijo napovedi (VO in INS) in korekcije (BLE) položaja. Ker Bluetooth lokalizacija vsebuje zakasnitev napram izhodu iz razširjenega

Kalmanovega filtra, slednjega shranjujemo in ga uporabimo v fazi časovne sinhronizacije. Zakasnitev Bluetooth lokalizacije ni znana vnaprej (se spreminja) in jo zato poskušamo ugotoviti sproti tako, da računamo razdalje med shranjenimi položaji iz razširjenega Kalmanovega filtra in položajem, ki ga vrne Bluetooth lokalizacija. Pričnemo z izračunom razdalje med najnovejšim izhodom iz razširjenega Kalmanovega filtra in meritvijo iz faze korekcije. Če je ta razdalja manjša od 0,2 m, lahko zaključimo, da v tem trenutku nimamo zakasnitve. V primeru, da je razdalja večja od 0,2 m, izračunamo novo razdaljo med izhodom iz razširjenega Kalmanovega filtra (ob času $k - 1$) in trenutno meritvijo iz faze korekcije. Glede na to novo izračunano razdaljo zaključimo, da je naša zakasnitev enaka enemu koraku (oz. času, ki je pretekel med detekcijo predzadnjega in zadnjega koraka), če je razdalja manjša od 0,2 m. Postopek lahko nadaljujemo, vendar če ne najdemo razdalje, ki bi bila manjša od 0,2 m, potem izberemo tisto, ki je najmanjša. Na ta način imamo približno določeno zakasnitev (v korakih). Upoštevanje zakasnitve za en korak nazaj pomeni, da s Kalmanovim filtrom izračunamo popravljen položaj ob času $k - 1$. Tega nato upoštevamo tako, da popravimo transformacijo iz ključnega K.S. (fuzije VO in INS) v K.S. zemljevida, ki je ob času (koraku) $k - 1$ predstavljal izhod razširjenega Kalmanovega filtra. Transformacijo iz ključnega K.S. ob času k v ključni K.S. ob času $k - 1$ (to je inkrementalni premik določen s fuzijo VO in INS) pustimo nespremenjeno. Na ta način dobimo popravljen transformacijo iz ključnega K.S. (fuzije VO in INS) v K.S. zemljevida, ki ob času (koraku) k določa položaj pešca v K.S. zemljevida. V primeru, če bi ugotovili, da je zakasnitev ekvivalentna pomiku za dva koraka nazaj, bi s Kalmanovim filtrom popravili transformacijo iz ključnega K.S. v K.S. zemljevida, dobljeno kot izhod iz razširjenega Kalmanovega filtra ob času $k - 2$, transformacijo iz ključnega K.S. ob času k v ključni K.S. ob času $k - 2$ pa bi pustili nespremenjeno (to sta zadnja dva inkrementalna premika določena s fuzijo VO in INS).

Torej po končani začetni inicializaciji poskušamo s časovno sinhronizacijo uskladiti napoved in meritev položaja (slika 7.2), kot smo pojasnili v prejšnjem odstavku. Če smo pri tem neuspešni, kar v našem primeru določimo tako, da preverimo, ali je razdalja med napovedanim in izmerjenim položajem (*»razd_1«* na sliki 7.1) ob nekem koraku k večja od 0,5 m, potem moramo onemogočiti korak

korekcije znotraj Kalmanovega filtra, saj je prišlo do napake znotraj Bluetooth lokalizacijskega sistema. Za kratko časovno obdobje namreč veliko bolj zaupamo fuziji vizualne odometrije in inercialnega navigacijskega sistema kot pa Bluetooth lokalizaciji.



Slika 7.2: Izpopolnjena fuzija vizualne odometrije, inercialnega navigacijskega sistema in Bluetooth lokalizacije

Pri določanju globalnega položaja z uporabo jakosti Bluetooth signalov lahko na določenih delih zemljevida dobimo več zaporednih položajev, ki so (v večji meri) nenatančno določeni le po eni osi (npr. po x). Da bi zmanjšali vpliv teh napak, po vsakem izračunu položaja s Kalmanovim filtrom preverimo odstopanje položaja, določenega z uporabo Bluetooth signalov, od položaja, ki ga dobimo s Kalmanovim filtrom. V primeru, da je ta razdalja (razd_2 na sliki 7.2) večja od enega metra, potem v naslednjem koraku upoštevamo to napako in popravimo položaj, ki smo ga dobili z metodo prstnih odtisov. Tovrstno dodatno predikcijo lahko upoštevamo le pri nekaj

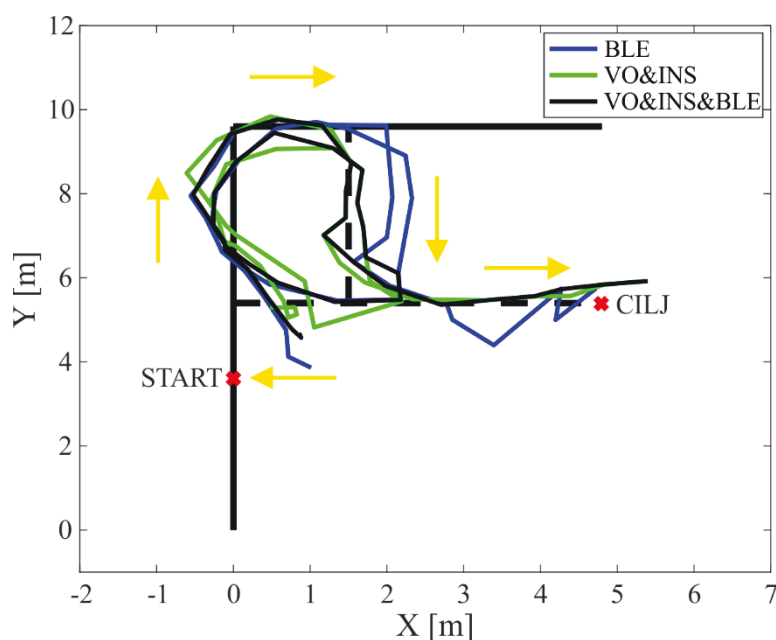
zaporednih korakih (v našem primeru smo jo omejili na največ pet korakov) in le v primeru, ko je več zaporednih položajev odmaknjenih v isto smer. Namreč, če bi se zaporedno izmenjavali dve nasprotujoči se smeri odmikov položajev, bi dobili vedno bolj napačen položaj po upoštevanju dodatni korekciji. Da do tega pojava ne pride, skrbi pogoj, ki preverja razdaljo med položajema iz faze predikcije in korekcije. Če je ta razdalja večja od 0,5 m, se korekcija znotraj Kalmanovega filtra izključi, upošteva pa se le predikcija (kar je smiselno v primeru, ko je položaj pridobljen z uporabo Bluetooth lokalizacije preveč oddaljen od ocenjenega položaja). Če se dodatna predikcija izvede petkrat zapored, potem zahtevamo ponovno inicializacijo položaja, kar pomeni, da se Kalmanov filter izvaja pri izklopljeni časovni sinhronizaciji toliko časa (dovolj je pet korakov), dokler se položaja iz faze predikcije in korekcije ne zblížata. Zatem ponovno omogočimo časovno sinhronizacijo, izklapljanje korekcije in dodatno predikcijo. Z opisano logiko povečamo vlogo relativne lokalizacije (VO in INS) pri njeni fuziji z radijsko lokalizacijo in s tem na določenih lokalnih območjih izboljšamo natančnost določanja položajev.

7.3 Rezultati fuzije lokalizacijskih metod

V notranjem okolju, ki smo ga opisali že pri prejšnjih preizkusih lokalizacijskih algoritmov, smo izvedli test, s katerim smo želeli preveriti, ali je s fuzijo vseh razvitih algoritmov (to so vizualna odometrija, inercialni navigacijski sistem in Bluetooth lokalizacija) mogoče doseči večjo robustnost lokalizacijskega sistema ter še dodatno izboljšati natančnost lokalizacije (skozi daljše časovno obdobje). Pešec, ki je izvajal preizkus, je bil opremljen s pametnim telefonom (Samsung Galaxy S6), na katerem so se vzporedno izvajali vsi lokalizacijski algoritmi ter tudi njihova fuzija na osnovi Kalmanovega filtra.

Na sliki 7.3 so prikazani rezultati lokalizacijskega preizkusa, pri katerem je pešec hodil po referenčni poti od štarta do cilja v smeri rumenih puščic, pri čemer je osrednji pravokotnik poti obhodil dvakrat. Približen začetni položaj pešca ($x = 0,6$; $y = 4,4$) je bil določen z uporabo globalne lokalizacije (ki temelji na metodi prstnih odtisov), in sicer kot mediana več zaporednih meritev. Dejanski položaj, v katerem je pešec začel hoditi, je bil pri $x = 0$, $y = 3,6$, kar pomeni, da je bila začetna napaka ravno 1 m. Ker relativna lokalizacija (fuzija VO in INS) zaradi približno ocenjenega

začetnega položaja začetnih nekaj korakov slabo napoveduje položaj v fazi predikcije Kalmanovega filtra, je brezpredmetno ugotavljati zakasnitev Bluetooth sistema že na začetku. Tako prvih pet korakov Kalmanov filter (ki temelji na modelu konstantne hitrosti) le združuje izhoda iz razširjenega Kalmanovega filtra (fuzija VO in INS) in Bluetooth lokalizacije. Korekcijski del Kalmanovega filtra mora biti medtem ves čas vklopljen, da se predikcija položaja (fuzija VO in INS) približa korekciji položaja (»meritvi« na osnovi Bluetooth lokalizacije).

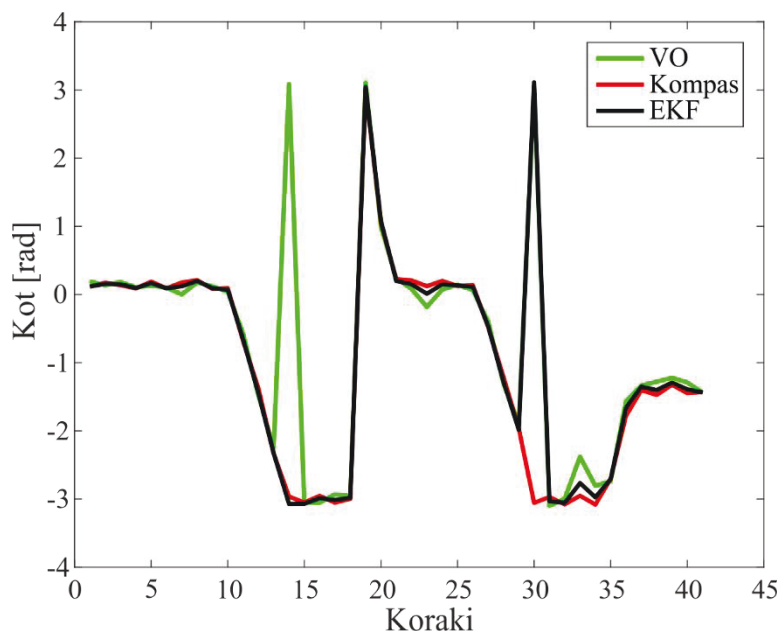


Slika 7.3: Rezultati lokalizacije pri uporabi posameznih metod (BLE, VO in INS) ter fuzije vseh metod. Pešec je prehodil pot od štarta to cilja v smeri rumenih puščic, pri čemer je osrednji pravokotnik obhodil dvakrat.

Orientacija pešca, ki je za celotno prehojeno pot prikazana na sliki 7.4, je bila določena kot fuzija vizualne odometrije in digitalnega kompasa z razširjenim Kalmanovim filtrom. Opazimo lahko, da se na sliki 7.4 pojavljajo posamezne špice, ki pa ne predstavljajo nobene napake, saj so prisotne le zaradi nezveznosti kota zasuka na območju od $-\pi$ do π .

Na sliki 7.5, kjer je med drugim prikazana razdalja (zelena krivulja) med položajem, ki predstavlja »meritev« v fazi korekcije in položajem, ki je rezultat Kalmanovega filtra, lahko vidimo tudi stopnjo zakasnitve (črtkana rdeča krivulja) Bluetooth lokalizacije. Vrednost 0 pomeni, da rezultat Bluetooth lokalizacije vsebuje zanemarljivo zakasnitev (glede na fuzijo VO in INS), vrednost 1 pa pomeni zakasnitev,

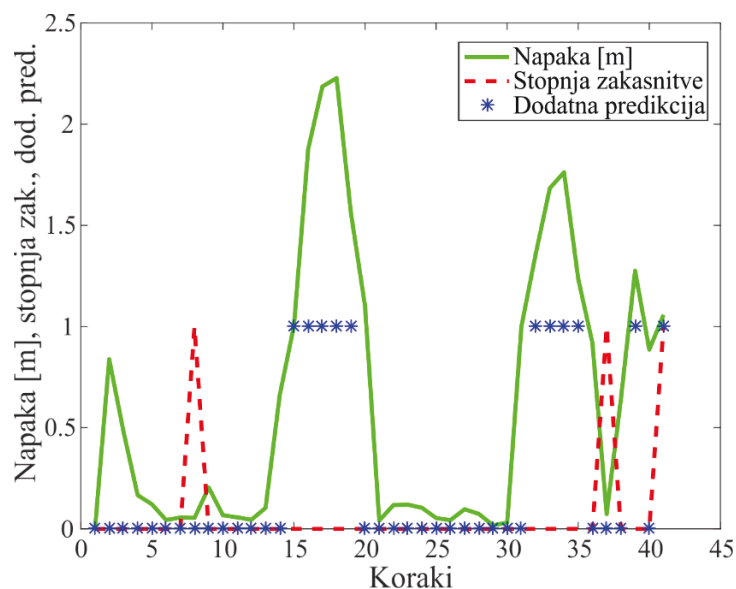
ki je ekvivalentna enemu koraku. Na sliki 7.5 vidimo, da lokalizacijski sistem ugotovi le tri zakasnitve Bluetooth lokalizacije glede na fuzijo VO in INS. Dejansko je prisotnih veliko več zakasnitev, vendar se jih en del zanemari (manjše zakasnitve), drugi del (večje zakasnitve) pa jih pokrije dodatna predikcija, ki se vključi v primeru, ko je razdalja med položajem iz faze korekcije in položajem, ki ga vrne Kalmanov filter, večja od 1 m. Pri dodatni predikciji se torej popravlja (meritev iz faze korekcije) položaj iz Bluetooth lokalizacije glede na njegov odmik (je bil večji od 1 m) v prejšnjem koraku od položaja, ki ga je vrnil Kalmanov filter. Ta odmik je prikazan na sliki 7.5 z zeleno krivuljo, z modrimi zvezdicami pa je prikazano stanje dodatne predikcije. Vrednost 0 pomeni, da je dodatna predikcija izključena, vrednost 1 pa pomeni, da je dodatna predikcija vključena.



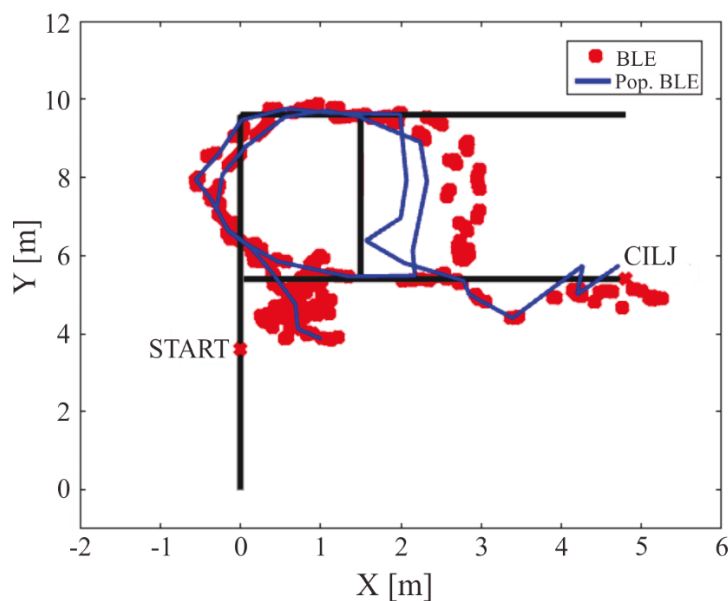
Slika 7.4: Orientacija pešca med hojo po referenčni poti, ki je prikazana na sliki 7.3

Potreba po dodatni predikciji se pojavi na delu poti, kjer pešec zamenja smer hoje (zato se pojavijo zakasnitve) in hkrati Bluetooth lokalizacija vrača položaj, ki je več kot 1 m oddaljen od dejanskega položaja (slika 7.6). Namesto dodatne predikcije bi lahko preprosto izključili korekcijski del Kalmanovega filtra, dokler se položaj iz Bluetooth lokalizacije in položaj iz fuzije VO in INS ne zblížata. Vendar, da bi zmanjšali občutljivost algoritma na potencialno napako iz fuzije VO in INS, rezultat iz Bluetooth lokalizacije še vedno upoštevamo (saj je položaj običajno napačen le

zaradi ene koordinate), vendar ga delno popravljamo z dodatno predikcijo. Na sliki 7.6 lahko vidimo položaje, določene z Bluetooth lokalizacijo (rdeče pike) in popravljene položaje z upoštevanjem dodatne predikcije (modra krivulja), ki so bili uporabljeni v korekcijskem delu Kalmanovega filtra. Na začetku poti je mogoče opaziti nekoliko večji raztros položajev (pridobljenih z Bluetooth lokalizacijo), saj je pešec nekaj trenutkov stal na mestu, preden je pričel hoditi.

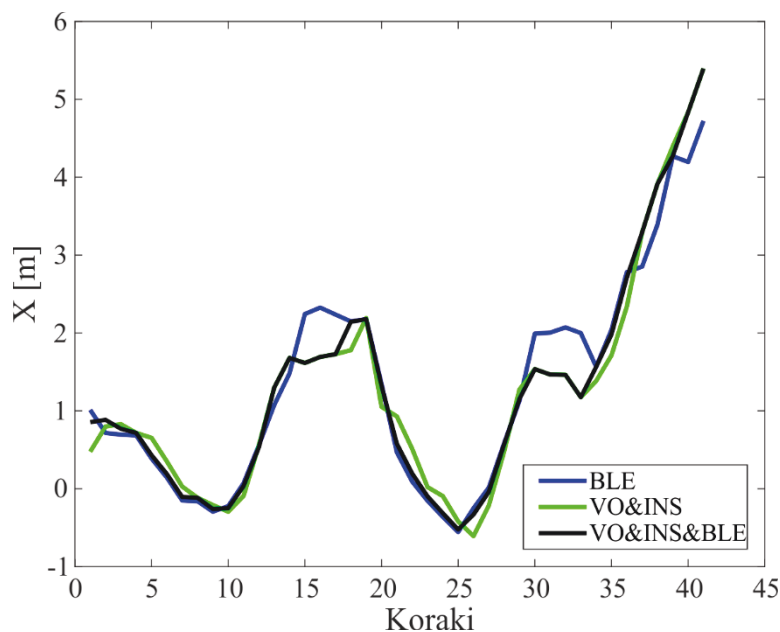


Slika 7.5: Razdalja med položajem, pridobljenim z Bluetooth lokalizacijo in izhodom iz Kalmanovega filtra, ter stopnja zakasnitve in stanje dodatne predikcije



Slika 7.6: Položaji določeni z Bluetooth lokalizacijo (rdeče pike) in popravljeni položaji pri upoštevanju dodatni predikciji (modra krivulja)

Na sliki 7.7, kjer je prikazan rezultat fuzije po osi X , lahko vidimo, da gre izhod Kalmanovega filtra pri korakih 15, 16 in 17 povsem s predikcijo na osnovi vizualne odometrije in inercialnega navigacijskega sistema. To pomeni, da je bila pri teh korakih razdalja med položajema iz faze predikcije in korekcije (kljub dodatnemu popravku položajev iz Bluetooth lokalizacije) večja od 0,5 m in posledično je lokalizacijski sistem izključil korekcijski del Kalmanovega filtra.

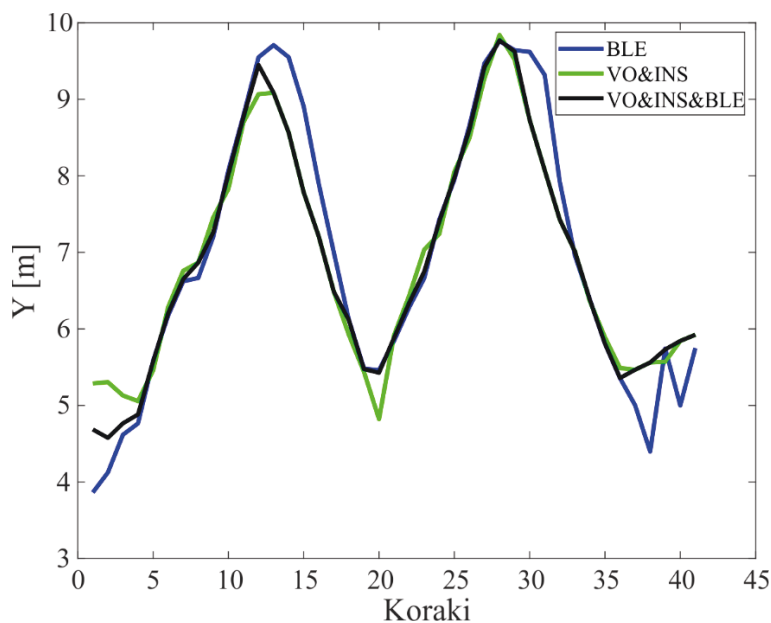


Slika 7.7: Fuzija po osi X

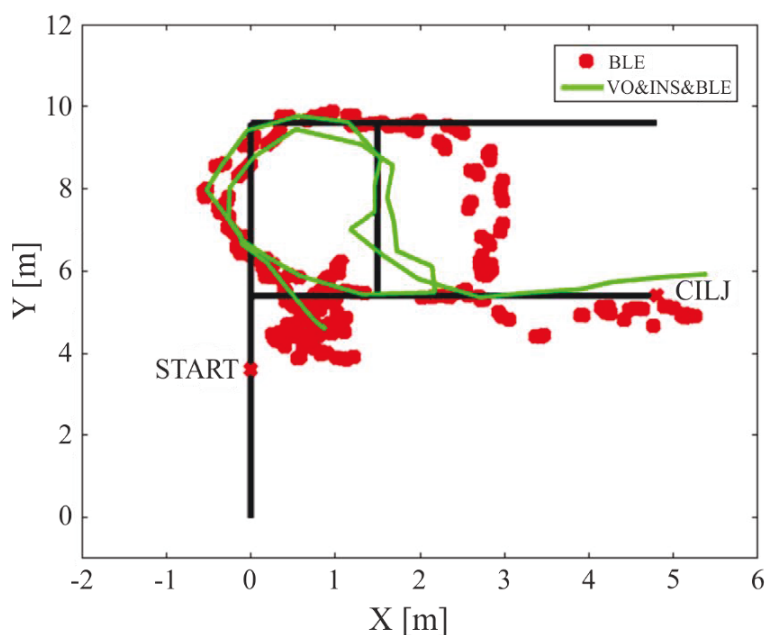
Od 18. koraka dalje pa se ponovno vključi korekcijski del Kalmanovega filtra in posledično se izhod Kalmanovega filtra spet približa meritvam položaja na osnovi Bluetooth lokalizacije. Na sliki 7.8, kjer je prikazan rezultat fuzije po osi Y , se lepo vidijo zakasnitve Bluetooth lokalizacije (od koraka 12 do koraka 18) glede na fuzijo VO in INS. Največje zakasnitve se pojavijo pri menjavi smeri hoje.

Na sliki 7.9, kjer so prikazani položaji, določeni z Bluetooth lokalizacijo (rdeče pike) in položaji določeni s fuzijo vizualne odometrije, inercialnega navigacijskega sistema ter Bluetooth lokalizacije (zelena krivulja), lahko vidimo, da je prispevek relativne lokalizacije zelo pomemben za (lokalno) izboljšavo globalne lokalizacije, predvsem v primeru, ko ta na nekem delu zemljevida postane manj natančna. Prispevek globalnega sistema pa je pomemben za ohranjanje lokalizacijskih napak znotraj določenih meja (ki so vezane na natančnost Bluetooth lokalizacije), ki jih pri

uporabi samo relativne lokalizacije ni mogoče določiti (saj napaka s časom samo narašča).



Slika 7.8: Fuzija po osi Y



Slika 7.9: Položaji, določeni z Bluetooth lokalizacijo ter fuzijo vizualne odometrije, inercialnega navigacijskega sistema ter Bluetooth lokalizacije

Ker pri vsakem narejenem koraku pešca nimamo podanega referenčnega položaja, lahko napako le ocenimo, tako da seštejemo razdalje med izbrano točko (položaja) in

njeno projekcijo na načrtano referenčno pot, po kateri je pešec hodil. Tako dobimo v primeru uporabe Bluetooth lokalizacije napako (normirano glede na število vseh točk) 0,86 m, v primeru fuzije vseh omenjenih lokalizacijskih metod pa 0,26 m. Na ta način lahko okvirno ocenimo izboljšavo natančnosti lokalizacije s fuzijo metod v primerjavi z Bluetooth lokalizacijo, ki je v našem primeru enaka 70 %. Če izračunamo kumulativno porazdelitev verjetnosti napak, dobimo za kar 88 % ocenjenih položajev napako manjšo od 0,5 m (vendar se moramo zavedati, da je v resnici nekoliko manjši delež napak manjših od 0,5 m, saj smo v našem primeru napake izračunali s projekcijo točk na referenčno pot).

Zaključimo lahko, da se je v našem primeru fuzija vseh razvitih lokalizacijskih metod pokazala za zelo koristno, saj smo z njeno uporabo zagotovili relativno visoko natančnost lokalizacije (glede na to, da uporabljeni senzorji in komunikacijski moduli niso neposredno namenjeni lokalizaciji v notranjem okolju), pri kateri napaka s časom ne narašča.

8 Zaključek

V doktorski disertaciji smo predstavili več različnih pristopov k lokalizaciji v notranjem okolju z uporabo pametnega telefona, ki ima vgrajene številne senzorje in komunikacijske module. Pri razvoju lokalizacijskega sistema smo se osredotočili na analizo informacij iz kamere, inercialnih senzorjev in Bluetooth modula, saj je bilo za njih v številnih študijah pokazano, da skrivajo največji potencial za uporabo pri lokalizaciji v notranjem okolju. Skladno z izbranimi senzorji in komunikacijskim modulom smo razvili tri različne algoritme, in sicer vizualno odometrijo, inercialni navigacijski sistem ter Bluetooth lokalizacijo, ki so v pričujočem delu natančno opisani in podprti z rezultati številnih eksperimentov. Z namenom, da bi dosegli kar se da robustno in natančno določanje položajev in zasukov, smo že pri razvoju posameznih lokalizacijskih algoritmov veliko časa posvetili fuziji senzorjev. Ta je še posebej potrebna pri uporabi inercialnih senzorjev, saj so vsi trije senzorji podvrženi napakam in zunanjim motnjam.

Za vizualno odometrijo se je izkazalo, da je njena prednost natančno določanje inkrementalnih premikov v primeru, da na poti ni ostrih zavojev. Težave vizualne odometrije nastopijo predvsem v primeru monotone teksture podlage ali ob slabi osvetlitvi prostora. Prednost inercialnega navigacijskega sistema se kaže predvsem v možnosti natančnega določanja zasukov z uporabo digitalnega kompasa. Pedometer, ki je del inercialnega navigacijskega sistema, se je izkazal za zelo zanesljivega, saj pri testiranju ni prihajalo do napačnih detekcij korakov. Izpopolnjeni digitalni kompas prav tako deluje zanesljivo v primeru, da ni izpostavljen dolgotrajni magnetni motnji. V nasprotnem primeru se zasuk nekoliko odmakne od prave vrednosti, vendar je to napako možno izločiti s fuzijo. Da bi dobili še bolj robustno in natančno relativno lokalizacijo, smo prednosti posameznih relativnih metod združili z uporabo

razširjenega Kalmanovega filtra, ki izloči večino napak, do katerih pride pri posamezni metodi. Rezultati eksperimentov so pokazali, da lahko z dobljenim relativnim lokalizacijskim sistemom zelo natančno določamo inkrementalne premike pešca, saj je napaka le okoli 3 % dolžine poti.

Slabost vizualne odometrije kot tudi inercialnega navigacijskega sistema je ta, da lahko ti dve metodi določata absolutne položaje v prostoru le v primeru, da poznamo položaj začetne točke. Zato smo razvili tudi globalni lokalizacijski sistem, ki temelji na uporabi meritev jakosti Bluetooth signalov. Z uporabo mehkih modelov in metode prstnih odtisov smo dobili najboljše rezultate, saj je bila pri 99 % ocenjenih položajev napaka manjša od 1 m oz. pri 53 % ocenjenih položajev manjša od 0,5 m. Metoda prstnih odtisov se je izkazala za bolj robustno kot metodi trilateracije in optimizacije z rojem delcev, saj sta slednji zelo občutljivi na šum, ki je prisoten v meritvah jakosti signalov, hkrati pa sta tudi bolj računsko potratni. Da bi napake globalnega lokalizacijskega sistema še nekoliko zmanjšali, smo implementirali fuzijo Bluetooth lokalizacije in natančnega relativnega lokalizacijskega sistema, ki temelji na vizualni odometriji in inercialnem navigacijskem sistemu. Rezultati eksperimenta so pokazali, da s to fuzijo uspešno zmanjšamo napake položajev, saj je bila pri več kot 80 % ocenjenih položajev napaka manjša od 0,5 m.

Celoten lokalizacijski sistem je bil implementiran na pametnem telefonu Android, kjer so bili algoritmi razdeljeni na več niti, ki se lahko izvajajo simultano. Na ta način je bilo doseženo izvajanje v realnem času, kar je še posebej pomembno za normalno delovanje vizualne odometrije, ki lahko v našem primeru obdela do 30 sličic na sekundo. Z implementacijo globalnega lokalizacijskega algoritma smo na pametnem telefonu dobili nov virtualni senzor, ki lahko določa inkrementalne premike ali globalni položaj v prostoru. Ta se lahko uporabi v aplikacijah na pametnem telefonu (npr. za vodenje slepih) ali kot samostojna ali dodatna senzorska enota na mobilnih sistemih.

Zaključimo lahko, da smo z razvojem in implementacijo posameznih lokalizacijskih algoritmov ter njihovo fuzijo dobili natančen globalni lokalizacijski sistem, ki je primeren za lokalizacijo oseb ali mobilnih sistemov v notranjem okolju. V okviru nadaljnjega dela bi lahko lokalizacijski sistem še nadgradili tako, da bi v oblaku implementirali sprotno gradnjo modelov in osveževanje zemljevida prstnih

odtisov glede na trenutno zajete meritve jakosti signalov. S tem bi se lokalizacija dodatno izboljšala predvsem v delih stavbe, kjer meritve niso bile zajete v prvi fazi gradnje zemljevida. S sprotnim osveževanjem prstnih odtisov pa bi zmanjšali tudi vpliv zunanje motnje, kot jo npr. predstavlja prisotnost večjega števila ljudi v prostoru.

9 Doprinos k znanosti

- Fuzija metod za relativno lokalizacijo na osnovi vizualne odometrije in inercialnega navigacijskega sistema ter metode za globalno lokalizacijo na osnovi Bluetooth radijskih signalov.
- Razvoj algoritma monokularne vizualne odometrije, ki je primeren za izvajanje v realnem času na pametnem telefonu.
- Izvedba globalnega lokalizacijskega sistema za določanje položaja v prostoru z uporabo meritev jakosti radijskih signalov Bluetooth ali WiFi.
- Izboljšava inercialnega navigacijskega sistema z uporabo digitalnega kompasa in pedometra, ki temeljita na zlivanju meritev inercialnih senzorjev.

Dodatek

A Singularni razcep

Singularni razcep – SVD [166, 167] (ang. Singular value decomposition) je matrična dekompozicija realne ali kompleksne matrike, ki se uporablja v statistiki in pri obdelavi signalov za najrazličnejše aplikacije. Singularni razcep obstaja za vsako matriko $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m \geq n$:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (\text{A.1})$$

kjer sta $\mathbf{U} \in \mathbb{R}^{m \times n}$ in $\mathbf{V} \in \mathbb{R}^{m \times n}$ ortogonalni matriki ter $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ diagonalna matrika:

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix}, \quad (\text{A.2})$$

kjer so $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$ singularne vrednosti matrike $\mathbf{A}$. Stolpci $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_m]$ so levi in $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$ desni singularni vektorji. Stolpci $\mathbf{u}_1, \dots, \mathbf{u}_m$ in $\mathbf{v}_1, \dots, \mathbf{v}_n$ tvorijo množico ortonormalnih vektorjev (ki so hkrati ortogonalni in enotski). σ_i je singularna vrednost matrike $\mathbf{A}$, če in samo če obstajata enotska vektorja $\mathbf{u}_i$ in $\mathbf{v}_i$ tako, da velja:

$$\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{u}_i \text{ in } \mathbf{A}^T\mathbf{u}_i = \sigma_i\mathbf{v}_i \quad i = 1, \dots, n. \quad (\text{A.3})$$

Za singularni razcep veljajo naslednje lastnosti:

- Lastne vrednosti $\mathbf{A}^T\mathbf{A}$ so $\sigma_1^2, \dots, \sigma_n^2$. Lastni vektorji $\mathbf{A}^T\mathbf{A}$ so desni singularni vektorji $\mathbf{v}_1, \dots, \mathbf{v}_n$.
- Lastne vrednosti $\mathbf{A}\mathbf{A}^T$ so $\sigma_1^2, \dots, \sigma_n^2, \underbrace{0, \dots, 0}_{m-n}$. Lastni vektorji $\mathbf{A}\mathbf{A}^T$ so levi singularni vektorji $\mathbf{u}_1, \dots, \mathbf{u}_m$.
- Če je $\mathbf{A} = \mathbf{A}^T$ ($\mathbf{A}$ je simetrična matrika), potem se da $\mathbf{A}$ diagonalizirati kot $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^T$, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. SVD za $\mathbf{A}$ je $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ za $\sigma_i = |\lambda_i|$ in $\mathbf{v}_i = \text{sign}(\lambda_i)\mathbf{u}_i$ ($\text{sign}(0) = 1$), kjer so λ_i lastne vrednosti matrike $\mathbf{A}$.

V primeru, ko je matrika $\mathbf{A}$ kvadratna matrika dimenzije $m \times m$ s pozitivno determinanto, so tudi matrike $\mathbf{U}$, $\mathbf{V}^T$ in $\mathbf{\Sigma}$ dimenzije $m \times m$, pri čemer matriki $\mathbf{U}$ in $\mathbf{V}^T$ predstavljata rotacijski matriki ter $\mathbf{\Sigma}$ skalirno matriko. Torej si lahko v tem primeru izraz $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ predstavljamo kot sekvenco treh geometrijskih transformacij, in sicer rotacije, skaliranja in še ene rotacije.

SVD se uporablja za reševanje predoločenih sistemov, saj velja naslednje; če je $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m \geq n$, $\text{rang}(\mathbf{A}) = n$, potem je minimum $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$ dosežen pri $\mathbf{x} = \sum_{i=1}^n \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i$.

B Kovariančna matrika šuma procesa

Imamo linearen dinamičen sistem tj. model konstantne hitrosti podan v prostoru stanj:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\tilde{\mathbf{w}}(t), \quad (\text{B.1})$$

$$\mathbf{z}(t) = \mathbf{H}\mathbf{x}(t) + \tilde{\mathbf{v}}(t), \quad (\text{B.2})$$

kjer so: $\mathbf{x}(t) = [x \ y \ \dot{x} \ \dot{y}]^T$, $\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ ter

$\tilde{\mathbf{w}}(t) = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix}$. Ker je $\tilde{\mathbf{w}}(t)$ beli šum velja:

$$E(\tilde{\mathbf{w}}(t)) = 0, \quad (\text{B.3})$$

$$E(\tilde{\mathbf{w}}(t)\tilde{\mathbf{w}}(\tau)) = \tilde{q}(t)\delta(t - \tau), \quad (\text{B.4})$$

kjer je $\tilde{q}$ je gostota spektralne moči in δ Diracova delta funkcija (B.4 je avtokorelacijska funkcija).

V diskretnem prostoru je matrika prehajanja stanj določena kot:

$$\mathbf{F}_k = e^{A\Delta t} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.5})$$

in šum procesa kot:

$$\mathbf{w}_k = \int_0^{\Delta t} e^{A(\Delta t - \tau)} \mathbf{B}\tilde{\mathbf{w}}(k\Delta t + \tau) d\tau. \quad (\text{B.6})$$

Kovariančno matriko šuma procesa dobimo kot:

$$\mathbf{Q} = E[\mathbf{w}_k \mathbf{w}_k^T] \quad (\text{B.7})$$

$$\begin{aligned}
\mathbf{Q} &= E \left[\int_0^{\Delta t} \begin{pmatrix} 1 & 0 & \Delta t - u & 0 \\ 0 & 1 & 0 & \Delta t - u \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tilde{\mathbf{w}}(k\Delta t + u) du \right. \\
&\quad \cdot \int_0^{\Delta t} \begin{pmatrix} 1 & 0 & \Delta t - v & 0 \\ 0 & 1 & 0 & \Delta t - v \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tilde{\mathbf{w}}(k\Delta t + v) dv)^T \Big] \\
&= \int_0^{\Delta t} \int_0^{\Delta t} \begin{bmatrix} \Delta t - u & 0 \\ 0 & \Delta t - u \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta t - v & 0 & 1 & 0 \\ 0 & \Delta t - v & 0 & 1 \end{bmatrix} \\
&\quad \cdot E[\tilde{\mathbf{w}}(k\Delta t + u) \tilde{\mathbf{w}}(k\Delta t + v)^T] dudv \\
&= \int_0^{\Delta t} \int_0^{\Delta t} \begin{bmatrix} \Delta t - u & 0 \\ 0 & \Delta t - u \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta t - v & 0 & 1 & 0 \\ 0 & \Delta t - v & 0 & 1 \end{bmatrix} \cdot \tilde{q}(k\Delta t + u) \delta(u - v) dudv \\
&= \int_0^{\Delta t} \begin{bmatrix} (\Delta t - v)^2 & 0 & \Delta t - v & 0 \\ 0 & (\Delta t - v)^2 & 0 & \Delta t - v \\ \Delta t - v & 0 & 1 & 0 \\ 0 & \Delta t - v & 0 & 1 \end{bmatrix} \tilde{q}(k\Delta t + v) dv = \\
&= \tilde{q} \begin{bmatrix} \frac{(\Delta t)^3}{3} & 0 & \frac{(\Delta t)^2}{2} & 0 \\ 0 & \frac{(\Delta t)^3}{3} & 0 & \frac{(\Delta t)^2}{2} \\ \frac{(\Delta t)^2}{2} & 0 & \Delta t & 0 \\ 0 & \frac{(\Delta t)^2}{2} & 0 & \Delta t \end{bmatrix},
\end{aligned}$$

kjer je $\tilde{q}$ konstantna vrednost, ki predstavlja gostoto spektralne moči. Pri izpeljavi (B.7) se v četrtem koraku upošteva: $\int_{-\infty}^{\infty} f(t) \delta(t - \tau) dt = f(\tau)$.

Literatura

- [1] J. Straub, S. Hilsenbeck, G. Schroth, R. Huitl, A. Möller, E. Steinbach: »Fast Relocalization For Visual Odometry Using Binary Features«. *In proceeding of: IEEE International Conference on Image Processing (ICIP)*, 2013.
- [2] G. Klein, D. Murray: »Parallel tracking and mapping for small AR workspaces«. *In IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 1–10, Nara, Japan, 2007.
- [3] C. Arth, D. Wagner, M. Klopschitz, A. Irschara, D. Schmalstieg: »Wide area localization on mobile phones«. *In IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 73–82, Orlando, Florida, USA, 2009.
- [4] W. V. Rossem: »An Extensible Framework for Indoor Positioning on Mobile Devices«. *Thesis, Faculty of Science, Department of Computer Science*, Vrije Universiteit Brussel, 2012.
- [5] G. Klein, D. Murray: »Parallel tracking and mapping on a camera phone«. *In IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 83–86, Orlando, Florida, USA, 2009.
- [6] R. Munguia, A. Grau: »Monocular SLAM for Visual Odometry«. *IEEE International Symposium on Intelligent Signal Processing, WISP 2007*, pp. 1–6, 2007.
- [7] M. Klopschitz, A. Irschara, G. Reitmayr, D. Schmalstieg: »Robust Incremental Structure from Motion«. *In 3DPVT*, 2010.
- [8] E. Eade, T. Drummond: »Scalable monocular SLAM«. *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 469–476, New York, NY, USA, 2006.
- [9] D. Nister, O. Naroditsky, J. Bergen: »Visual odometry«. *In proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 652–659, 2004.
- [10] C. Arth, M. Klopschitz, G. Reitmayr, D. Schmalstieg: »Real-Time Self-Localization from Panoramic Images on Mobile Devices«. *In IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 37–46, 2011.

- [11] J. Campbell, R. Sukthankar, I. Nourbakhsh and A. Pahwa: »A Robust Visual Odometry and Precipice Detection System Using Consumergrade Monocular Vision«. *In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA*, pp. 3421–3427, 2005.
- [12] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, F. Zhao: »A reliable and accurate indoor localization method using phone inertial sensors«. *In Proceedings of the ACM Conference on Ubiquitous Computing (UbiComp '12)*, ACM, New York, NY, USA, 2012.
- [13] V. Honkavirta, T. Perala, S. Ali-Loytty and R. Pich'e: »A comparative survey of wlan location fingerprinting methods«. *In Positioning, Navigation and Communication, WPNC 2009. 6th Workshop on. IEEE*, pp. 243–251, 2009.
- [14] Z. Jianyong, C. Zili, L. Haiyong, L. Zhaohui: »RSSI based Bluetooth Low Energy Indoor Positioning«. *IPIN 2014*, 2014.
- [15] S. Mazuelas, A. Bahillo, R. Lorenzo, P. Fernandez, F. Lago, E. Garcia, J. Blas, E. Abril: »Robust indoor positioning provided by real-time RSSI values in unmodified WLAN networks«. *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, pp. 821–831, 2009.
- [16] J. Li: »Characterization of WLAN Location Fingerprinting Systems«. *Master of Science by Research Institute of Computing Systems Architecture, School of Informatics*, University of Edinburgh, 2012.
- [17] S. Sirtkaya, B. Seymen, A.A. Alatan: »Loosely coupled Kalman filtering for fusion of Visual Odometry and inertial navigation«. *In proceeding of 16th International Conference on Information Fusion (FUSION)*, pp. 219–226, 2013.
- [18] T. Lupton, S. Sukkarieh: »Visual-inertial-aided navigation for highdynamic motion in built environments without initial conditions«. *IEEE Transactions on Robotics*, 28(1), pp. 61–76, 2012.
- [19] S. Chan, G. Sohn: »Indoor localization using WI-FI based fingerprinting and trilateration techniques for LBS applications«. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2012.
- [20] O. J. Woodman: »An introduction to inertial navigation«. *Technical Report, University of Cambridge, Computer laboratory*, 2007.
- [21] G. Klančar, I. Škrjanc: »Robot localization using inertial and Wi-Fi signal strength sensors«. *In proceeding of: 10th IFAC Symposium on robot control (SYROCO 2012)*, pp. 139–144, Dubrovnik, 2012.
- [22] GNSS, <<http://www.gnss.si/kako-deluje/globalni-navigacijski-satelitski-sistemi>>, 3.9.2016
- [23] Število uporabnikov pametnih telefonov: <<https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>>, 3.9.2016

-
- [24] Google Indoor maps: <<http://maps.google.com/help/maps/indoormap/>>, 2013
 - [25] Microsoft Indoor Navigation: <<http://research.microsoft.com/en-us/projects/indoorloc/>>, 2013
 - [26] Micello: <<https://www.micello.com/>>, 3.9.2016
 - [27] Nokia indoor navigation: <<http://research.nokia.com/news/12425>>, 2013
 - [28] Meridian indoor positioning: <http://www.meridianapps.com/howitworks#option_b>, 2013
 - [29] IndoorAtlas: <<http://www.indooratlas.com>>, 2013
 - [30] B. Micusik and H. Wildenauer: »Descriptor free visual indoor localization with line segments«. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3165-3173, Boston, MA, 2015.
 - [31] Igra z obogateno resničnostjo: <<http://www.pokemongo.com/>> , 20.9.2016
 - [32] M. Bošnjak, D. Matko, S. Blažič: »Quadrocopter Hovering Using Position-estimation Information from Inertial Sensors and a High-delay Video System«, *Journal of Intelligent and Robotic Systems*, 67(1), pp. 43-60, July 2012.
 - [33] A. Zdešar, I. Škrjanc and G. Klančar: »Visual trajectory-tracking model-based control for mobile robots«. *Int. Journal of Advanced Robotic Systems*, 10(1), pp. 1-11, 2013.
 - [34] A. Zdešar: »Vodenje avtonomnega mobilnega sistema z robotsko roko na osnovi digitalne slike«, doktorska disertacija, Ljubljana, 2015.
 - [35] D. G. Lowe: »Object Recognition from Local Scale-Invariant Features«. In *Proceedings of the International Conference on Computer Vision*, (ICCV '99), vol. 2. IEEE Computer Society, pp. 1150--, Washington, DC, USA, 1999.
 - [36] H. Bay, A. Ess, T. Tuytelaars and L. V. Gool: »Speeded-Up Robust Features (SURF)«, *Comput. Vis. Image Underst.*, 110(3), pp. 346-359, 2008.
 - [37] M. Calonder, V. Lepetit, C. Strecha and P. Fua: »BRIEF: Binary Robust Independent Elementary Features«. *11th European Conference on Computer Vision (ECCV)*, Heraklion, Crete. LNCS Springer, September 2010.
 - [38] M. Calonder, V. Lepetit, C. Strecha, P. Fua: »BRIEF: Binary robust independent elementary features«. *European Conference on Computer Vision (ECCV)*, pp. 778-792, 2010.
 - [39] H. Bay, T. Tuytelaars, L. Van Gool: »SURF: Speeded up robust features«. *European Conference on Computer Vision (ECCV)*, pp. 404-417, 2006.
 - [40] P.L. Rosin: »Measuring corner properties«. *Computer Vision and Image Understanding*, 73(2), pp. 291-307, 1999.

- [41] S. Leutenegger, M. Chli, R.Y. Siegwart: »BRISK: Binary robust invariant scalable keypoints«. *In IEEE International Conference on Computer Vision (ICCV)*, pp. 2548-2555, Barcelona, Spain, 2011.
- [42] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, D. Schmalstieg: »Pose tracking from natural features on mobile phones«. *In IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, IEEE Computer Society, pp. 125-134, Cambridge, UK, 2008.
- [43] A. Alahi, R. Ortiz, P. Vandergheynst: »FREAK: Fast retina keypoint«. *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (To Appear), Providence, Rhode Island, 2012.
- [44] E. Rosten, T. Drummond: »Fusing points and lines for high performance tracking«. *In IEEE International Conference on Computer Vision (ICCV)*, vol. 2, pp. 1508-1515, 2005.
- [45] K. M. Saipullah, N. A. Ismail, A. Anuar, N. Sarimin: »Comparison of Feature Extractors for Real-Time Object Detection on Android Smartphone«. *Journal of Theoretical and Applied Information Technology*, pp. 135-142, 2013.
- [46] J. Straub: »Visual Localization based on Binary Features«. *Diplomarbeit*, Technische Universitat Munchen, Munchen 2012.
- [47] Navvis TUM: <<http://navvis.de/>>, 2016
- [48] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, E. Steinbach: »TUMIndoor: An extensive image and point cloud dataset for visual indoor localization and mapping«. *In IEEE ICIP*, Orlando, USA, Sept. 2012.
- [49] C. Forster, M. Pizzoli and D. Scaramuzza: »SVO: Fast semi-direct monocular visual odometry«. *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 2014, pp. 15-22, 2014.
- [50] L. Porzi, E. Ricci, T. A. Ciarfuglia: »Visual-inertial Tracking on Android for Augmented Reality Applications«. *Environmental Energy and Structural Monitoring Systems (EESMS)*, 2012, pp. 35-41, 2012.
- [51] E.S. Jones, S. Soatto: »Visual-inertial navigation, mapping and localization: A scalable real-time causal approach«. *The International Journal of Robotics Research (IJRR)*, 30(4), pp. 407-430, 2011.
- [52] B. M. Kitt, J. Rehder, A. D. Chambers, M. Schonbein, H. Lategahn and S. Singh: »Monocular Visual Odometry using a Planar Road Model to Solve Scale Ambiguity«. *Proc. European Conference on Mobile Robots*, September, 2011.
- [53] S. Hilsenbeck, A. Moller, R. Huitl, G. Schroth, M. Kranz, E. Steinbach: »Scale-preserving long-term visual odometry for indoor navigation«. *Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1-10, 2012.
- [54] H. Strasdat, J. M. M. Montiel, A. J. Davison: »Scale drift-aware large scale monocular SLAM«. *In Robotics: Science and Systems*. The MIT Press, 2010.

-
- [55] A.J. Davison: »Real-time simultaneous localisation and mapping with a single camera«. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 1403-1410, Nice, France, 2003.
 - [56] A. J. Davison, I.D. Reid, N.D. Molton, O. Stasse: »MonoSLAM: Real-time single camera SLAM«. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(6), pp. 1052-1067, 2007
 - [57] R. Sim, P. Elinas, M. Grin, J.J. Little and oth.: »Vision-based SLAM using the rao-blackwellised particle filter«. In *IJCAI Workshop on Reasoning with Uncertainty in Robotics*, vol. 14, pp. 9-16, Edinburgh, Scotland, UK, 2005.
 - [58] Eigen: <[http://eigen.tuxfamily.org/index.php?title= Main_Page](http://eigen.tuxfamily.org/index.php?title=Main_Page)>, 2016.
 - [59] T. Schöps, J. Engel, D. Cremers: »Semi-Dense Visual Odometry for AR on a Smartphone«, *ISMAR*, 2014.
 - [60] R. Mur-Artal, J. M. M. Montiel and J. D. Tardós: »ORB-SLAM: A Versatile and Accurate Monocular SLAM System«. *IEEE Transactions on Robotics*, 31(5), pp. 1147-1163, October 2015.
 - [61] F. Dellaert, W. Burgard, D. Fox, S. Thrun: »Using the condensation algorithm for robust, vision-based mobile robot localization«. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, Ft. Collins, CO, USA, 1999.
 - [62] G. Reitmayr, T.W. Drummond: »Going out: Robust model-based tracking for outdoor augmented reality«. In *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 109-118, Santa Barbara, CA, USA, 2006.
 - [63] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas, A. Calway: »Real-time and robust monocular SLAM using predictive multi-resolution descriptors«. *Advances in Visual Computing*, pp. 276-285, 2006.
 - [64] A. M. Fischler, R. C. Bolles: »Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography«. *Commun. ACM* 24, pp. 381-395, 1981.
 - [65] J. M. Airey, J. H. Rohlf, F. P. Brooks, Jr.: »Towards Image Realism with Interactive Update Rates in Complex Virtual Building Environments«. In *Proc. Symposium on Interactive 3D Graphics*, pp. 41-50, New York, NY, USA, 1990.
 - [66] X. Li, J. Wang: »Image matching techniques for vision-based indoor navigation systems: performance analysis for 3D map based approach«. *Indoor Positioning and Indoor Navigation (IPIN)*, 2012 International Conference on , pp. 1-8, 2012.
 - [67] G. Zhang, Z. Dong, J. Jia, T. T. Wong, H. Bao: »Efficient non-consecutive feature tracking for structure-from-motion«. In *Proceedings of the European Conference on Computer Vision (ECCV '10)*, 2010.

- [68] J. Sivic, A. Zisserman: »Video google: A text retrieval approach to object matching in videos«. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 1470-1477, Nice, France, 2003.
- [69] D. Galvez-Lopez, J.D. Tardos: »Bags of binary words for fast place recognition in image sequences«. *IEEE Transactions on Robotics*, 2012.
- [70] D. Galvez-Lopez, J.D. Tardos: »Bags of binary words for fast place recognition in image sequences«. *IEEE Transactions on Robotics*, 2012.
- [71] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, E. Steinbach: »Virtual reference view generation for CBIR-based visual pose estimation«. In *ACM Multimedia (MM)*, Nara, Japan, October 2012. ACM 2012.
- [72] V. V. Nguyen, J. W. Lee: »Self-positioning system for indoor navigation on mobile phones«. *Consumer Electronics (ICCE)*, 2012 IEEE International Conference, pp. 114-115, 2012.
- [73] S. Taylor, E. Rosten, T.W. Drummond: »Robust feature matching in 2.3 microseconds«. *IEEE Workshop on Feature Detectors and Descriptors*, 2009.
- [74] A. M. G. Pinto, A. P. Moreira, P. G. Costa: »Indoor Localization System based on Artificial Landmarks and Monocular Vision«. *TELKOMNIKA*, 10(4), pp. 609-620, 2012.
- [75] D. Zachariah, M. Jansson: »Fusing visual tags and inertial information for indoor navigation«. *Position Location and Navigation Symposium (PLANS)*, 2012 IEEE/ION, pp. 535-540, 2012.
- [76] Q. Wang, Y. Liu, J. Chen: »Accurate indoor tracking using a mobile phone and non-overlapping camera sensor networks. *Instrumentation and Measurement Technology Conference (I2MTC)*, 2012 IEEE International, pp. 2022-2027, 2012.
- [77] P. D. Groves: »Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition«. Artech House, 2013.
- [78] A. R. Pratama, Widyawan and R. Hidayat: »Smartphone-based Pedestrian Dead Reckoning as an indoor positioning system«. *System Engineering and Technology (ICSET)*, 2012 International Conference on, pp. 1-6, Bandung, 2012.
- [79] U. Shala, A. Rodriguez: »Indoor Positioning using Sensor-fusion in Android Devices«. *Bachelor thesis*, School of Health and Society, Department Computer Science, Embedded Systems, 2011.
- [80] J. Scarlet: »Enhancing the Performance of Pedometers Using a Single Accelerometer«. *Analog Devices AN-900 Application Note*, 2005.
- [81] Zemeljsko magnetno polje:
<http://www.nasa.gov/mission_pages/sunearth/news/gallery/Earths-magneticfieldlines-dipole.html>, 22.8.2016
- [82] P. Dabove, G. Ghinamo and A. M. Lingua: »Inertial Sensors for Smartphones Navigation«. *SpringerPlus* 4, pp. 834, 2015.

-
- [83] Android Developers: Sensors Overview, <http://developer.android.com/guide/topics/sensors/sensors_overview.html>, 2016
 - [84] P. Lawitzki: »Application of Dynamic Binaural Signals in Acoustic Games«. *Master Thesis*, Stuttgart Media University, Faculty Print and Media, 2012
 - [85] M. Placer, S. Kovačič: »Enhancing Indoor Inertial Pedestrian Navigation Using a Shoe-Worn Marker«. *Sensors*, 13(8), pp. 9836, 2013.
 - [86] A. Hansson, L. Tufvesson: »Using Sensor Equipped Smartphones to Localize WiFi Access Points«, *Master Thesis*, Lund University, Department of Automatic Control, 2012.
 - [87] B. Gozick, K. P. Subbu, R. Dantu, T. Maeshiro: »Magnetic Maps for Indoor Navigation. Instrumentation and Measurement«. *IEEE Transactions on Instrumentation and Measurement*, 60(12), pp. 3883-3891, 2011.
 - [88] J. Blankenbach, A. Norrdine: »Position estimation using artificial generated magnetic fields«. *Indoor Positioning and Indoor Navigation (IPIN)*, International Conference on, pp. 1-5, 2010.
 - [89] M. Ibrahim, M. Youssef: »CellSense: An Accurate Energy-Efficient GSM Positioning System«. *IEEE Transactions on Vehicular Technology*. 61(1), pp. 286–296, 2012.
 - [90] Y. Zhuang, Z. Syed, J. Georgy, N. El-Sheimy: »Autonomous smartphone-based WiFi positioning system by using access points localization and crowdsourcing«. *Pervasive and Mobile Computing*, vol. 18, pp. 118-136, April 2015.
 - [91] X. Y. Lin, T. W. Ho, C. C. Fang, Z. S. Yen, B. J. Yang and F. Ali: »A mobile indoor positioning system based on iBeacon technology«. *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4970-4973, Milan, 2015.
 - [92] Y. Chen, D. Lymberopoulos, J. Liu and B. Priyantha: »FM-based indoor localization«. *In Proc. of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*, 2012.
 - [93] B. Ozdenizci, C. Vedat and O. Kerem: »NFC Internal: An Indoor Navigation System«. *Sensors (Basel, Switzerland)*, pp. 7571–7595, 2015.
 - [94] H. Liu, H. Darabi, P. Banerjee, J. Liu: »Survey of wireless indoor positioning techniques and systems«. *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, IEEE Transactions on, 37(6), pp. 1067-1080, 2007.
 - [95] K. Curran, E. Furey, T. Lunney, J. Santos, D. Woods, A. McCaughey: »An evaluation of indoor location determination technologies«. *Journal of Location Based Services*, 5(2), pp. 61-78, 2011.
 - [96] Nokia indoor navigation: <<http://research.nokia.com/news/11809>>, 2013

- [97] N. Mahiddin: »Indoor Position Detection Using WiFi and Trilateration Technique«. *The International Conference on Informatics and Applications (ICIA2012)*, 2012.
- [98] C. Gentner, T. Jost: »Indoor positioning using time difference of arrival between multipath components«. *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, pp. 1-10, 2013.
- [99] T. Fernandes: »Indoor Localization Using Bluetooth«. *Proc FEUP Doctoral Symp. on Informatics Engineering - DSIE*, Porto, Portugal, January, 2011.
- [100] H. Liu, H. Darabi, P. Banerjee, J. Liu: »Survey of wireless indoor positioning techniques and systems«. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 37, pp. 1067–1080, 2007.
- [101] M. Robinson, I. Psaromiligkos: »Received signal strength based location estimation of a wireless LAN client«. *In Proceedings of IEEE Wireless Communications and Networking Conference*, vol. 4, pp. 2350–2354, New Orleans, LA, 2005.
- [102] A. Dempster, B. Li, I. Quader: »Errors in deterministic wireless fingerprinting systems for localisation«. *In Proceedings of 3rd International Symposium on Wireless Pervasive Computing*, pp. 111–115, Santorini, Greece, 2008.
- [103] A. Rai, K. Chintalapudi, V. Padmanabhan, R. S. Zee: »Zeroeffort crowdsourcing for indoor localization«. *In Proceedings of the 18th annual international conference on Mobile computing and networking*, pp. 293–304. ACM, 2012.
- [104] M. Ciurana, F. Barcelo-Arroyo, F. Izquierdo: »A ranging method with IEEE 802.11 data frames for indoor localization«. *In Proceedings of IEEE Wireless Communications and Networking Conference*, pp. 2092–2096, Hong Kong, March 2007.
- [105] A. Narzullaev, Y. Park, H. Jung: »Accurate signal strength prediction based positioning for indoor WLAN systems«. *In Proceedings of IEEE on Position, Location and Navigation Symposium*, pp. 685–688, Monterey, CA, 2008.
- [106] J. Zheng, C. Wu, H. Chu, P. Ji: »Localization algorithm based on RSSI and distance geometry constrain for wireless sensor network«. *In Proceedings of International Conference on Electrical and Control Engineering (ICECE)*, pp. 2836–2839, 2010.
- [107] M. Robinson, I. Psaromiligkos: »Received signal strength based location estimation of a wireless LAN client«. *In Proceedings of IEEE Wireless Communications and Networking Conference*, vol. 4, pp. 2350–2354, New Orleans, LA, 2005.
- [108] H. Nurminen, J. Talvitie, S. Ali-Loytty, P. Muller, E. Lohan, R. Piche, M. Renfors: »Statistical path loss parameter estimation and positioning using RSS measurements in indoor wireless networks«. *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, pp. 1-9, 2012.

-
- [109] Y. Ji, S. Biaz, S. Pandey, P. Agrawal: »ARIADNE: a dynamic indoor signal map construction and localization system«. In *Proceedings of ACM MobiSys*, pp. 151-164, Uppsala, Sweden, 2006.
 - [110] W. Yeung, J. Ng: »An enhanced wireless LAN positioning algorithm based on the fingerprint approach«. In *Proceedings of IEEE TENCON*, pp. 1-4, Hong Kong, 2006.
 - [111] X. Luo, W. J. O'Brien, C. Julien: »Comparative evaluation of received signal strength index (RSSI) based indoor localization techniques for construction jobsites«. *Advanced Engineering Informatics*, vol. 25, pp. 355-363, 2011.
 - [112] L. Jiang: »A WLAN fingerprinting based indoor localization technique«. *Thesis*, The Graduate College at the University of Nebraska, 2012.
 - [113] P. Bahl, V. Padmanabhan: »RADAR: an in-building RF-based user location and tracking system«. In *Proceedings of IEEE INFOCOM*, pp. 775-784, Tel Aviv, Israel, March 2000.
 - [114] V. Moghtadaiee, A. G. Dempster: »WiFi fingerprinting signal strength error modeling for short distances«. *Indoor Positioning and Indoor Navigation (IPIN)*, 2012 *International Conference on*, pp. 1-6, 2012.
 - [115] P. Castro, P. Chiu, T. Kremenek, R. R. Muntz: »A probabilistic room location service for wireless networked environments«. In *Proceedings of ACM Ubicomp*, pp. 18-34, Atlanta, Georgia, 2001.
 - [116] V. Savic, A. Poblacio, S. Zazo, M. Garcianda: »An experimental study of RSS based indoor localization using nonparametric belief propagation based on spanning trees«. In *Proceedings of Fourth International Conference on SENSORCOMM*, pp. 238-243, Venice/Mestre, Italy, 2010.
 - [117] N. Swangmuang, P. Krishnamurthy: »Location fingerprint analyses toward efficient indoor positioning«. In *Proceedings of Sixth Annual IEEE International Conference on, Pervasive Computing and Communications*, pp. 100-109, Hong Kong, 2008.
 - [118] B. Altintas, T. Serif: »Indoor location detection with a RSS-based short term memory technique« (KNN-STM). In *Proceedings of IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 794-798, Lugano, Switzerland, 2012.
 - [119] M. Ghaddar, L. Talbi, T. Denidni: »Human body modelling for prediction of effect of people on indoor propagation channel«. *Electronics Letters*, vol. 40, pp. 1592-1594, 2004.
 - [120] J. S. Lim, W. H. Jang, G. W. Yoon, D. S. Han: »Radio Map Update Automation for WiFi Positioning Systems«. *Communications Letters, IEEE*, 17(4), pp. 693-696, 2013.
 - [121] P. Mirowski, P. Whiting, H. Steck, R. Palaniappan, M. MacDonald, D. Hartmann, T. K. Ho: »Probability kernel regression for WiFi localisation«. *J. Locat. Based Serv.*, 6(2), pp. 81-100, 2012.

- [122] M. Youssef, A. Agrawala: »The Horus WLAN Location Determination System«. In Mobysys, pp 205-218, 2005.
- [123] M. Brunato, R. Battiti: »Statistical Learning Theory for Location Fingerprinting in Wireless LANs«. Computer Networks, 47(6), pp. 825-845, 2005.
- [124] R. Henniges: Current approaches of Wifi Positioning. Service-Centric Networking- Seminar WS 2011/2012, pp. 1-8, TU-Berlin, 2012.
- [125] J. M. Alonso, M. Ocana, N. Hernandez, F. Herranz, A. Llamazares, M. Sotelo, L. M. Bergasa, L. Magdalena: »Enhanced WiFi localization system based on Soft Computing techniques to deal with small-scale variations in wireless sensors«. Applied Soft Computing, 11(8), pp. 4677-4691, 2011.
- [126] N. Viol, J. A. B. Link, H. Wirtz, D. Rothe, K. Wehrle: »Hidden Markov model-based 3D path-matching using raytracing-generated Wi-Fi models«. *Indoor Positioning and Indoor Navigation (IPIN)*, 2012 International Conference on, pp. 1-10, 2012.
- [127] S. Jialin Pan, V. W. Zheng, Q. Yang, D. Hao Hu: »Transfer Learning for WiFi-based Indoor Localization«. In the Workshop on Transfer Learning for Complex Task of the 23rd AAAI Conference on Artificial Intelligence, pp. 43-48, 2008.
- [128] Z. Yang, C. Wu, Y. Liu: »Locating in fingerprint space: wireless indoor localization with little human intervention«. *In Proceedings of the 18th annual international conference on Mobile computing and networking (Mobicom '12)*. ACM, New York, NY, USA, pp. 269-280, 2012.
- [129] B. Ferris, D. Fox, N. Lawrence: »Wifi-slam using Gaussian process latent variable models«. *In Proceedings of IJCAI*, pp. 2480-2485, 2007.
- [130] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, A. Aggarwal: »Efficient, generalized indoor wifi graphslam«. *In IEEE International Conference on Robotics and Automation, ICRA 2011*, Shanghai, China, 2011, pp. 1038-1043, 2011.
- [131] M. Cypriani, P. Canalda, F. Spies, A. Dobircan: »Benchmark measurements for Wi-Fi signal strength-based positioning system«. *Indoor Positioning and Indoor Navigation (IPIN)*, 2012 International Conference on, pp. 1-8, 2012.
- [132] Ekahau: WiFi tags, < <http://www.ekahau.com/real-time-location-system/technology/wi-fi-tags>>, 2013.
- [133] R. R. Thomas: »Optimisation of Adaptive localisation techniques for cognitive radio«. *Master of Engineering*, Faculty of Engineering, Built Environment in Information Technology, University of Pretoria, 2012.
- [134] B. Kannan, F. Meneguzzi, M. B. Dias, K. Sycara: »Predictive indoor navigation using commercial smart-phones«. *In Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC '13)*, ACM, pp. 519-525, New York, NY, USA, 2013.

-
- [135] Y. Liu, Q. Wang, J. Liu, T. Wark: »MCMC-based indoor localization with a smartphone and sparse WiFi access points«. *Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2012 IEEE International Conference on, pp. 247-252, 2012.
 - [136] K. Chen, K. R. Vadde: »Design and Evaluation of an Indoor Positioning System Framework«. 2012.
 - [137] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, Y. Zhang: »Walkie-Markie: indoor pathway mapping made easy«. *In Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation (nsdi'13)*, Nick Feamster and Jeff Mogul (Eds.). USENIX Association, pp. 85-98, Berkeley, CA, USA, 2013.
 - [138] M. Sonka, V. Hlavac, R. Boyle: »Image Processing, Analysis, and Machine Vision«. 4th Edition, Cengage Learning, pp. 598-601, 2014.
 - [139] Z. Zhang: »A flexible new technique for camera calibration«. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22(11), pp. 1330–1334, 2000.
 - [140] Y. Ma, S. Soatto, J. Kosecka, S. S. Sastry: »An Invitation to 3-D Vision: From Images to Geometric Models«. SpringerVerlag, pp. 27-28, 2004.
 - [141] J. Y. Bouguet: »Camera Calibration Toolbox for Matlab«. <http://www.vision.caltech.edu/bouguetj/calib_doc/>, 1.6.2016
 - [142] BoofCV, Calibration, <http://boofcv.org/index.php?title=Tutorial_Camera_Calibration>, 1.6.2016
 - [143] Matlab 2014a, Computer Vision System Toolbox: <<http://www.mathworks.com/products/computer-vision/>>, 1.6.2016
 - [144] M. Sonka, V. Hlavac, R. Boyle: »Image Processing, Analysis, and Machine Vision«. 3rd edition, Thomson Engineering, pp. 757-760, 2007.
 - [145] B. D. Lucas, T. Kanade: »An Iterative Image Registration Technique with an Application to Stereo Vision«. *International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.
 - [146] C. Tomasi, T. Kanade: »Detection and Tracking of Point Features«. Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.
 - [147] J. Shi, C. Tomasi: »Good Features to Track«. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.
 - [148] J. K. Suhr: »Kanade-Lucas-Tomasi (KLT) Feature Tracker«. Computer Vision Laboratory, Seoul, Korea, 2009.
 - [149] J. Y. Bouguet: »Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm«. Intel Corporation Microprocessor Research Labs, 2001.

- [150] H. Moravec: »Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover«. Tech Report CMU-RI-TR-3 Carnegie-Mellon University, Robotics Institute, 1980.
- [151] R. Siegwart, I. Nourbakhsh, D. Scaramuzza: »Introduction to Autonomous Mobile Robots«. Second Edition. A Bradford Book, The MIT Press, ISBN: 978-0-262-01535-6, February, 2011.
- [152] C. Harris, M. Stephens: »A combined corner and edge detector«. Proceedings of the 4th Alvey Vision Conference. pp. 147–151, 1988.
- [153] T. Lindeberg: »Feature detection with automatic scale selection«. *International Journal of Computer Vision*, vol. 30(2), pp. 79-116, 1998.
- [154] M. Sonka, V. Hlavac, R. Boyle: »Image Processing, Analysis, and Machine Vision, 3rd edition«. Thomson Engineering, pp. 132-136, 2007
- [155] Sobel Edge Detector:
<<http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm>>, 14.4.2016
- [156] K. Engel: Real-time volume graphics, pp. 112–114, 2006
- [157] M. Sonka, V. Hlavac, R. Boyle: »Image Processing, Analysis, and Machine Vision, 3rd Edition, Thompson«, 2008, pp. 553-561.
- [158] D. A. Forsyth, J. Ponce: »Computer Vision: a modern approach«. Prentice Hall, pp. 365-375, 2003.
- [159] M.A. Fischler, R.C. Bolles: »Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography«, *Commun. ACM*, 24(6), pp. 381-395, New York, 1981.
- [160] R. I. Hartley, A. Zisserman: »Multiple View Geometry in Computer Vision«. 2nd edition, Cambridge University Press, pp. 117-127, 2004.
- [161] Y. Ma, S. Soatto, J. Kosecka, S. S. Sastry: »An Invitation to 3-D Vision: From Images to Geometric Models«. SpringerVerlag, pp. 19-37, 2004.
- [162] D.W. Eggert, A. Lorusso, R.B. Fisher: »Estimating 3-D rigid body transformations: a comparison of four major algorithms. Machine Vision and Applications«, (9), pp. 272-290, 1997.
- [163] R. Szeliski: »Computer Vision Algorithms and Applications. Springer London«, pp. 320-321, 2011.
- [164] K. S. Arun, T. S. Huang, S. D. Blostein: »Least-Squares Fitting of Two 3-D Point Sets«, *IEEE Transactions on pattern analysis and machine intelligence, PAMI* 9, (5), 1987.
- [165] R. Siegwart, I. Nourbakhsh, D. Scaramuzza: »Introduction to Autonomous Mobile Robots, Second Edition«. A Bradford Book, The MIT Press, ISBN: 978-0-262-01535-6, pp. 187-189, February, 2011.
- [166] M. Sonka, V. Hlavac, R. Boyle: »Image Processing, Analysis, and Machine Vision«, 4th Edition, Cengage Learning, 2014, pp. 73-74.
- [167] B. Plestenjak: Singularni razcep. <http://www-lp.fmf.uni-lj.si/plestenjak/vaje/NlaBol/Gradivo/10_SVD.pdf>, 16.5.2015

-
- [168] H. Leppakoski, J. Collin, J. Takala: »Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals«. *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 1569-1572, March 2012.
 - [169] Android Coordinate System:
<<http://developer.android.com/reference/android/hardware/SensorEvent.html#values>>, 2016
 - [170] Android Kalman filter implementation:
><https://android.googlesource.com/platform/frameworks/native/+/master/services/sensorservice/Fusion.cpp><, Junij, 2016
 - [171] K. Shoemake: »Animating rotation with quaternion curves«. *In Proceedings of the 12th annual conference on Computer graphics and interactive techniques* (SIGGRAPH '85), ACM, pp. 245-254., New York, USA, 1985.
 - [172] T. Zengshan, Z. Yuan, Z. Mu, L. Yu: »Pedestrian dead reckoning for MARG navigation using a smartphone«. *EURASIP Journal on Advances in Signal Processing*, vol. 65(1), Springer, 2014.
 - [173] M. Hardegger, G. Tröster, D. Roggen: »Improved action SLAM for long-term indoor tracking with wearable motion sensors«. *In Proceedings of the 2013 International Symposium on Wearable Computers* (ISWC '13). pp. 1-8, New York, USA, 2013.
 - [174] T. Gallagher, E. Wise, L. Binghao, A. G. Dempster, and other: »Indoor positioning system based on sensor fusion for the Blind and Visually Impaired«. *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, pp. 1-9, 2012.
 - [175] S. Sirtkaya, B. Seymen, A. A. Alatan: »Loosely coupled Kalman filtering for fusion of Visual Odometry and inertial navigation«. *Information Fusion (FUSION)*, 16th International Conference on, pp. 219-226, 2013.
 - [176] R. Faragher and R. Harle: »Location Fingerprinting With Bluetooth Low Energy Beacons«. *In IEEE Journal on Selected Areas in Communications*, vol. 33(11), pp. 2418-2428, Nov. 2015.
 - [177] M. Karikallio: »Techniques to enhance accuracy and automations of BLE positioning systems«, Master's thesis, University of Oulu, 2015
 - [178] S. Mazuelas, A. Bahillo, R. Lorenzo, P. Fernandez, F. Lago, E. Garcia, J. Blas, E. Abril: »Robust indoor positioning provided by real-time RSSI values in unmodified WLAN networks«. *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, pp. 821-831, 2009.
 - [179] X. Luo, W. J. O'Brien, C. Julien: »Comparative evaluation of received signal strength index (RSSI) based indoor localization techniques for construction jobsites«. *Advanced Engineering Informatics*, vol. 25, pp. 355-363, 2011.
 - [180] S. Ahlberg: »Evaluation of Different Radio-Based Indoor Positioning Methods«, Master's degree, Linköping University, 2014.

-
- [181] Bluetooth, SIG, <<https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth>>, 21.5.2016.
 - [182] Bluetooth radio interface, modulation and channels: <<http://www.radio-electronics.com/info/wireless/bluetooth/radio-interface-modulation.php>>, 21.5.2016
 - [183] Specification of the Bluetooth System, Publication date: 30 June 2010 <https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737>, 2016.
 - [184] Kontakt.io, Transmission power, Range and RSSI, <<https://support.kontakt.io/hc/en-gb/articles/201621521-Transmission-power-settings>>, 22.5.2016.
 - [185] K. Townsend, C. Cufi, Akiba and R. Davidson: »Getting Started with Bluetooth Low Energy«, O'Reilly Media, 2014.
 - [186] Bluetooth developers: <<https://developer.bluetooth.org/TechnologyOverview/Pages/GATT.aspx>>, 24.5.2016.
 - [187] R. Faragher and R. Harle: »Location Fingerprinting with Bluetooth Low Energy Beacons«, *IEEE Journal on Selected Areas in Communications*, vol. 33(11), pp. 2418 – 2428, 2015.
 - [188] Physical Web by Google: <<http://google.github.io/physical-web/>>, 27.5.2016.
 - [189] Bluetooth Low Energy Beacons, Texas Instruments, Application Report, Jan. 2015, <<http://www.ti.com.cn/cn/lit/an/swra475/swra475.pdf>>, 27.5.2016.
 - [190] iBeacon, Apple: <<https://developer.apple.com/ibeacon/>>, 28.5.2016.
 - [191] Eddystone, Google: <<https://developers.google.com/beacons/>>, 28.5.2016.
 - [192] Eddystone beacons, Estimote: <<http://developer.estimote.com/eddystone/>>, 29.5.2016.
 - [193] Eddystone beacons, Google: <<https://github.com/google/eddystone>>, 29.5.2016.
 - [194] Kontakt.io, <<https://support.kontakt.io/hc/en-gb/articles/201621521-Transmission-power-Range-and-RSSI>>, 22.3.2016.
 - [195] Kontakt.io, <<https://kontakt.io/blog/ibeacon-configuration-guide-transmission-power/>>, 22.3.2016.
 - [196] S. Tomažič, I. Škrjanc: »Fusion of visual odometry and inertial navigation system on a smartphone«, *Computers in Industry*, vol 74, pp. 119-134, 2015.
 - [197] B. Hartmann, O. Banfer, O. Nelles, A. Sodja, L. Teslic and I. Škrjanc: »Supervised Hierarchical Clustering in Fuzzy Model Identification«. *In IEEE Transactions on Fuzzy Systems*, vol. 19(6), pp. 1163-1176, Dec. 2011.
 - [198] L. Teslic, B. Hartmann, O. Nelles and I. Škrjanc: »Nonlinear System Identification by Gustafson–Kessel Fuzzy Clustering and Supervised Local

Model Network Learning for the Drug Absorption Spectra Process». *In IEEE Transactions on Neural Networks*, vol. 22(12), pp. 1941-1951, Dec. 2011.

- [199] R. Babuška and H. B. Verbruggen: »An overview of fuzzy modeling for Control«. *Control Eng. Practice*, vol. 4(11), pp. 1593-1606, 1996.
- [200] T.A. Johansen, R. Shorten, and R. Murray-Smith: »On the interpretation and identification of Takagi–Sugeno fuzzy models«. *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 297-313, Jun. 2000.
- [201] T. Takagi and M. Sugeno: »Fuzzy identification of systems and its application to modeling and control«. *IEEE Trans. Syst., Man, Cybern.*, vol. 15(1), pp. 116-132, Feb. 1985.
- [202] D. E. Gustafson and W. C. Kessel: »Fuzzy clustering with a fuzzy covariance matrix«. *In Proc. IEEE Conf. Decis. Control*, pp. 761-766, 1979.
- [203] I. Škrjanc: »Fuzzy confidence interval for pH titration curve«. *Applied Mathematical Modelling*, vol. 35(8), pp. 4083-4090, August 2011.
- [204] Y. Wang, X. Yang, Y. Zhao, Y. Liu, L. Cuthbert: »Bluetooth positioning using RSSI and triangulation methods«, *In Consumer Communications and Networking Conference (CCNC)*, 2013 IEEE , pp. 837-842, 2013.
- [205] N. A. Mahiddin, N. Safie, E. Nadia, S. Safei, F. Engku: Indoor position detection using WiFi and trilateration technique, *The International Conference on Informatics and Applications*, 2012.
- [206] C. L. Lawson and R. J. Hanson: »Solving Least Squares Problems«, Englewood Cliffs, NJ: Prentice-Hall., 1974.
- [207] H. G. Golub and C. F. Van Loan: »Matrix Computations«, 3rd edition, *Johns Hopkins University Press*, Baltimore, MD, USA. 1996.
- [208] A. Ben-Israel and T.N.E. Greville: »Generalized Inverses: Theory and Applications«, CMS Books in Mathematics, Springer, 2003.
- [209] W. S. Murphy Jr.: »Determination of a Positioning using Approximate Distances and Trilateration«, *M.S. thesis*, Faculty and the Board of Trustees of the Colorado School of Mines, Golden, Colorado, 2007.
- [210] Y.X. Yuan: »Nonlinear Optimization: Trust Region Algorithms«, < <ftp://lsec.cc.ac.cn/pub/yyx/papers/p946d.pdf> >, 10.9.2016
- [211] J.J. Moré, and D.C. Sorensen: "Computing a Trust Region Step", *SIAM Journal on Scientific and Statistical Computing*, vol. 3, pp 553–572, 1983.
- [212] R.H. Byrd, R.B. Schnabel, and G.A. Shultz: "Approximate Solution of the Trust Region Problem by Minimization over Two-Dimensional Subspaces", *Mathematical Programming*, vol. 40, pp 247–263, 1988.
- [213] T. Steihaug: "The Conjugate Gradient Method and Trust Regions in Large Scale Optimization", *SIAM Journal on Numerical Analysis*, vol. 20, pp 626–637, 1983.

- [214] M.A. Branch, T.F. Coleman and Y. Li: "A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems", *SIAM Journal on Scientific Computing*, vol. 21(1), pp 1-23, 1999.
- [215] T.F. Coleman and A. Verma: "A Preconditioned Conjugate Gradient Approach to Linear Equality Constrained Minimization", *Computational Optimization and Applications*, vol. 20(1), pp. 61-72, 2001.
- [216] D.C. Sorensen: »Minimization of a Large Scale Quadratic Function Subject to an Ellipsoidal Constraint«, Department of Computational and Applied Mathematics, Rice University, Technical Report TR., pp. 94-27, 1994.
- [217] T. F. Coleman and Y. Li: »An interior trust region approach for nonlinear minimization subject to bounds«, *SIAM Journal on Optimization*, 6(2), pp. 418-445, 1996.
- [218] J. Nocedal and S. J. Wright: »*Numerical Optimization*, Second Edition«, Springer Series in Operations Research, Springer Verlag, pp. 102-120, 2006.
- [219] R. Koenker and K. F. Hallock: »Quantile Regression«, *Journal of Economic Perspectives*, vol. 15(4), pp. 143-156, 2011.
- [220] S. Gorard: »Revisiting a 90-year-old debate: The advantages of the mean deviation«, *British Journal of Educational Studies*, vol. 53(4), Blackwell Publishing Ltd, 2005
- [221] J. Kennedy: »The particle swarm: social adaptation of knowledge«. *In proceedings of IEEE International Conference on Evolutionary Computation*, pp. 303-308, 1997.
- [222] J. Kennedy, R. Eberhart: »Particle Swarm Optimization«, *In proceedings of IEEE International Conference on Neural Networks*. pp. 1942-1948, 1995.
- [223] Y. Shi, R.C. Eberhart: »A modified particle swarm optimizer«, *In proceedings of IEEE International Conference on Evolutionary Computation*, pp. 69-73, 1998.
- [224] J. Kennedy: »The particle swarm: social adaptation of knowledge«, *In proceedings of IEEE International Conference on Evolutionary Computation*, pp. 303-308, 1997.
- [225] M. Clerc: »Standard Particle Swarm Optimisation«, HAL open access archive, >https://hal.archives-ouvertes.fr/file/index/docid/764996/filename/SPSO_descriptions.pdf<, 2012.
- [226] M. Taherkhani, R. Safabakhsh: »A novel stability-based adaptive inertia weight for particle swarm optimization«, *Applied Soft Computing*, vol. 38, pp. 281-295, 2016.
- [227] Y. Shi, R.C. Eberhart: »Parameter selection in particle swarm optimization«, *In proceedings of Evolutionary Programming VII (EP98)*, pp. 591–600, 1998.

-
- [228] R.C. Eberhart, Y. Shi: »Comparing inertia weights and constriction factors in particle swarm optimization«, *In proceedings of the Congress on Evolutionary Computation*, pp. 84-88, 2000.
 - [229] A. Carlisle, G. Dozier: »An Off-The-Shelf PSO«, *In proceedings of the Particle Swarm Optimization Workshop*, pp. 1-6, 2001.
 - [230] F. Van den Bergh: »An Analysis of Particle Swarm Optimizers«, *PhD thesis*, University of Pretoria, Faculty of Natural and Agricultural Science, 2001.
 - [231] M. Clerc, J. Kennedy: »The particle swarm - explosion, stability, and convergence in a multidimensional complex space«, *IEEE Transactions on Evolutionary Computation*, vol. 6 (1), pp. 58-73, 2002.
 - [232] I.C. Trelea: »The Particle Swarm Optimization Algorithm: convergence analysis and parameter selection«, *Information Processing Letters*, vol. 85(6), pp. 317-325, 2003.
 - [233] D. Bratton, T. Blackwell: »A Simplified Recombinant PSO«, *Journal of Artificial Evolution and Applications*, 2008.
 - [234] G. Evers: »An Automatic Regrouping Mechanism to Deal with Stagnation in Particle Swarm Optimization«, *Master's thesis*, The University of Texas - Pan American, Department of Electrical Engineering, 2009.
 - [235] M. Meissner, M. Schmuker, G. Schneider: »Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training«, *BMC Bioinformatics*, vol. 7 (1), 2006.
 - [236] M.E.H. Pedersen: »Tuning & Simplifying Heuristical Optimization«, *PhD thesis*, University of Southampton, School of Engineering Sciences, Computational Engineering and Design Group, 2010.
 - [237] M.E.H. Pedersen, A.J. Chipperfield: »Simplifying particle swarm optimization«, *Applied Soft Computing*, vol. 10 (2), pp. 618-628, 2010.
 - [238] D. Bratton, J. Kennedy: »Defining a Standard for Particle Swarm Optimization«, *In proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)*, 2007.
 - [239] M. Zambrano-Bigiarini, M. Clerc, R. Rojas: »Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements«, *Evolutionary Computation (CEC), IEEE Congress on*, 2013.
 - [240] M. Lovbjerg, T. Krink: »The LifeCycle Model: combining particle swarm optimisation, genetic algorithms and hillclimbers«, *In proceedings of Parallel Problem Solving from Nature VII (PPSN)*, pp. 621-630, 2002.
 - [241] T. Niknam, B. Amiri: »An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis«, *Applied Soft Computing*, vol. 10 (1), pp. 183-197, 2010.
 - [242] Wen-Jun Zhang, Xiao-Feng Xie: »DEPSO: hybrid particle swarm with differential evolution operator«, *IEEE International Conference on Systems*,

- Man and Cybernetics (SMCC), Washington, DC, USA, pp. 3816-3821, 2003.
- [243] Y. Ji, S. Biaz, S. Pandey, P. Agrawal: ARIADNE: a dynamic indoor signal map construction and localization system. In *Proceedings of ACM MobiSys*, pp. 151–164, Uppsala, Sweden, 2006.
 - [244] V. Honkavirta, T. Perala, S. Ali-Loytty and R. Piche: »A comparative survey of WLAN location fingerprinting methods«. In *Positioning, Navigation and Communication, WPNC 2009*, 6th Workshop on, pp. 243-251, March 2009.
 - [245] L. Jiang: A WLAN fingerprinting based indoor localization technique. Thesis, The Graduate College at the University of Nebraska, 2012.
 - [246] L. Chen, L. Pei, H. Kuusniemi, Y. Chen, T. Kroger, and R. Chen: »Bayesian fusion for indoor positioning using Bluetooth fingerprints«. *Wireless Personal Communications*, 70(4), pp. 1735-1745, 2013.
 - [247] F. Subhan, H. Hasbullah, A. Rozyyev and S. Bakhsh: »Indoor positioning in Bluetooth networks using fingerprinting and lateration approach«. In *Information Science and Applications (ICISA)*, 2011, *International Conference on*, pp. 1-9, 2011.
 - [248] T. Guan, W. Dong, D. Koutsonikolas, G. Challen and C. Qiao: »Robust, Cost-Effective and Scalable Localization in Large Indoor Areas«. *IEEE Global Communications Conference (GLOBECOM)*, San Diego, CA, pp. 1-6, 2015.
 - [249] R. Faragher and R. Harle: »Location Fingerprinting With Bluetooth Low Energy Beacons«. In *IEEE Journal on Selected Areas in Communications*, vol. 33(11), pp. 2418-2428, Nov. 2015.
 - [250] P. N. Yianilos: Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms (SODA '93)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp. 311-321, 1993.
 - [251] F. A. Wai-chee, P. M. Chan, Y. Cheung, Y. S. Moon: Dynamic vp-tree indexing for n-nearest neighbor search given pair-wise distances. *The International Journal on Very Large Data Bases*, Springer-Verlag New York, Inc. Secaucus, NJ, USA. pp. 154-173, 2000.
 - [252] F. Nielsen: Bregman vantage point trees for efficient nearest Neighbor Queries, In *Proceedings of Multimedia and Exp (ICME)*, IEEE, pp. 878-881, 2009.
 - [253] R. Kalman: »A New Approach to Linear Filtering and Prediction Problems«. *Journal of Basic Engineering*, pp. 82-35, 1960.
 - [254] R. Kalman, R. S. Bucy: »New Results in Linear Filtering and Prediction Theory«, ASME. *J. Basic Eng.*, vol. 83(1), pp. 95-108, 1961.
 - [255] N. Kovvali, M. Banavar, A. Spanias: »An Introduction to Kalman Filtering with MATLAB Examples«, *Synthesis Lectures on Signal Processing*. pp. 23-63, Morgan & Claypool Publishers, September 2013.

- [256] P. D. Groves: »Principles of Gnss, Inertial, and Multisensor Integrated Navigation Systems«, pp. 82-117, Artech House, Boston, 2013.