

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SINIŠA ŠEGVIĆ

UPORABA PROJEKCIJSKE GEOMETRIJE I AKTIVNOG VIDA U TUMAČENJU SCENA

Magistarski rad

Zagreb, 2000.

Magistarski rad je izrađen na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave.

Mentor: **Prof. dr. sc. Slobodan Ribarić**

Magistarski rad ima 100 stranica

Rad broj:

Povjerenstvo za ocjenu rada u sastavu:

1. Prof. dr. sc. Leo Budin, predsjednik;
2. Prof. dr. sc. Slobodan Ribarić, mentor;
3. Prof. dr. sc. Nikola Pavešić, Fakulteta za elektrotehniko, Ljubljana.

Povjerenstvo za obranu u sastavu:

1. Prof. dr. sc. Leo Budin, predsjednik;
2. Prof. dr. sc. Slobodan Ribarić, mentor;
3. Prof. dr. sc. Nikola Pavešić, Fakulteta za elektrotehniko, Ljubljana.

Datum obrane: 3. svibnja 2000.

Sadržaj

1	Uvod	1
1.1	Opis problema	1
1.2	Odabrani pristup	3
1.3	Moguće primjene	3
1.4	Struktura rada	4
2	Pregled pristupa samostalnoj lokalizaciji	5
2.1	Klasifikacija sustava za samostalnu navigaciju	6
2.2	Tehnike lokalizacije	7
2.2.1	Mjerenje prijednog puta	7
2.2.2	Inercijska navigacija	8
2.2.3	Određivanje apsolutne orijentacije	8
2.2.4	Lokalizacija uz pomoć aktivnih referentnih objekata	8
2.2.5	Prepoznavanje pasivnih referentnih objekata	9
2.2.6	Modeliranje strukture okoliša	10
2.3	Lokalizacija upotrebom računarskog vida	11
2.3.1	Lokalizacija prepoznavanjem referentnih objekata	11
2.3.2	Lokalizacija modeliranjem strukture okoliša	14
3	Modeli, metode i teoretske pretpostavke	16
3.1	Model kamere	16
3.2	Model scene	19
3.3	Projekcijska geometrija	19
3.3.1	Projekcijski prostor	21
3.3.2	Točke i pravci u projekcijskoj ravnini	21
3.3.3	Nedogled kao slika idealne točke prostora \mathcal{P}^3	22
3.4	Aktivni vid	23
3.4.1	Motivacija	23
3.4.2	Definicija aktivnog vida	23
3.4.3	Svojstva sustava aktivnog vida	24
4	Određivanje težinskog pravocrtnog prikaza slike	26
4.1	Pronalaženje rubova	26
4.1.1	Usporedba postojećih algoritama	27
4.1.2	Cannyjev algoritam za pronalaženje rubnih elemenata	28
4.1.3	Određivanje parametara za Cannyjev algoritam	30
4.1.4	Izvedbeni detalji algoritma	32
4.2	Povezivanje rubnih elemenata u pravocrtne segmente	35
4.2.1	Houghova transformacija za pravce	36
4.2.2	Pregled postojećih modifikacija HT za pravce	40
4.2.3	Odabrana modifikacija izvornog postupka	40
4.2.4	Analiza parametarskog prostora	42

4.2.5	Izlučivanje pravocrtnih segmenata	43
4.2.6	Eksperimentalni rezultati	45
4.2.7	Izvedbeni detalji algoritma	49
4.3	Ocjena rezultata pretprocesiranja	52
5	Lokalizacija u sustavu hodnika	53
5.1	Pronalaženje nedogleda	53
5.1.1	Pregled postojećih algoritama	54
5.1.2	Idealizirane scene hodnika	57
5.1.3	Odabir pristupa pronalaženju nedogleda	57
5.1.4	Pronalaženje dominantnog nedogleda vodoravnih bridova scene	59
5.1.5	Primjenljivost predloženog postupka za lokalizaciju	61
5.1.6	Eksperimentalni rezultati	62
5.2	Lokalizacijska procedura	69
5.3	Eksperimentalni rezultati	72
6	Rasprava	75
6.1	Mogući pristupi lokalizaciji	75
6.2	Međuprikaz	76
6.3	Odabrani pristup lokalizaciji	76
6.4	Eksperimentalni rezultati	77
6.5	Budući rad	77
A	Upravljanje kamerom i prijenos slika	78
A.1	Prioriteti oblikovanja programske podrške	78
A.1.1	Lakoća održavanja i testiranja	79
A.1.2	Mogućnost višestrukog korištenja	80
A.1.3	Prenosivost	81
A.2	Objektno orijentirano oblikovanje	82
A.3	Programski podsustav za upravljanje kamerom	83
A.3.1	Zahtjevi	83
A.3.2	Arhitektura	83
A.3.3	Izvedba	84
A.4	Programski podsustav za prijenos slike	86
A.4.1	Zahtjevi	86
A.4.2	Arhitektura	87
A.4.3	Izvedba	87
B	Uputstva za instalaciju i korištenje razvijenih programa	90
B.1	Program za obavljanje lokalizacijske procedure	90
B.2	Konačna izvedba upravljanja digitalizatorom i kamerom	91
C	Izvorni i izvršni kôd	93
	Bibliografija	93
	Kratki sažetak	99
	Summary	100
	Životopis	101

Popis slika

1.1	Koordinatni sustavi hodnika $(\mathbf{O}, \mathbf{X}, \mathbf{Y}, \mathbf{Z})$ i robota $(\mathbf{o}, \mathbf{x}, \mathbf{y}, \mathbf{z})$	2
1.2	Stupnjevi slobode kamere.	2
2.1	Klasifikacija robota prema nesigurnosti s kojom se susreću.	6
2.2	Osnovni problem trijagulacije: robot određuje svoj položaj (x_R, y_R, φ) u koordinatnom sustavu svijeta $(\mathbf{O}, \mathbf{X}, \mathbf{Y})$ na temelju azimuta $\lambda_0, \lambda_1, \lambda_2$ svjetionika S_0, S_1, S_2 u koordinatnom sustavu robota $(\mathbf{o}, \mathbf{x}, \mathbf{y})$	9
3.1	Jednostavni model kamere s koordinatnim sustavima svijeta (O, K, L, M) , kamere (C, X, Y, Z) i slike (c, x, y) . Točka T trodimenzionalnog prostora se preslikava u točku t ravnine projekcije.	17
4.1	Slike i odgovarajući histogrami sivih razina dobiveni za dva različita para kamera i digitalizatora slike.	28
4.2	Slike i odgovarajući histogrami dobiveni gladenjem slika na sl. 4.1 uz $\sigma = 1,7$	28
4.3	Iznos gradijenta u slici na sl. 4.1(b).	29
4.4	Binarna maska za stanjivanje rubova u slici na sl. 4.3. Svijetli slikovni elementi odgovaraju elementima slike na sl. 4.3 koje je u postupku stanjivanja potrebno odbaciti.	29
4.5	Konačan rezultat pronalaženja rubova u slici na sl. 4.1(b).	30
4.6	Slika neravnomjerno osvijetljene scene hodnika (a) te rezultat algoritma za pronalaženje rubova (b).	31
4.7	Alternativni prikaz rezultata obrade sa sl. 4.6 i sl. 4.5.	31
4.8	Prikaz toka obrade algoritma za pronalaženje rubova. Pravokutnici označavaju ulazne i izlazne slike te međurezultate, dok krugovi označavaju korake obrade (opširnije u tekstu).	32
4.9	Primjer upotrebe Houghove transformacije za pronalaženje pravaca: binarna slika sa slikovnim elementima A, B, C i D (a), parametarski prostor sa odgovarajućim pravcima p_A, p_B, p_C i p_D , (b) te odgovarajuće polje akumulatora nakon transformacije (c).	37
4.10	Izvorna slika (a), težinska slika rubnih elemenata (b), binarna slika rubnih elemenata (c) sa ucertanim pravcem $p(\theta_p, \rho_p)$ koji odgovara lokalnom maksimumu #5 u polju akumulatora parametarskog prostora dobivenog Houghovom transformacijom (d).	38
4.11	Uvećani detalj slike na sl. 4.10(d).	39
4.12	Tanka i debela crna krivulja označavaju doprinos točke T sa sl. 4.10 transformiranoj slici u skladu sa izvornom odnosno polariziranom HT, uz $\Delta\varphi = 10^\circ$	41
4.13	Polje akumulatora parametarskog prostora dobiveno na temelju sl. 4.10(b), težinskom polariziranom HT. Elementi parametarskog prostora koji odgovaraju markantnim pravocrtnim segmentima izvorne slike su označeni krugovima.	42

4.14	Lokalni maksimumi koji su pronađeni analizom slike na sl. 4.13(b), sortirani po vrijednostima odgovarajućih akumulatora. Treba primijetiti da područje #0 ipak obuhvaća dva akumulatora koji odgovaraju markantnim pravocrtnim segmentima na sl. 4.10(a).	43
4.15	Pravocrtni segmenti u slici na sl. 4.10(b) koji se nalaze na pravcima koji odgovaraju lokalnim maksimumima na sl. 4.14.	44
4.16	Pravocrtni segmenti u sl. 4.10(b) koji su pronađeni nakon 11. iteracije postupka. Izlučeni segmenti su označeni pridruženim težinama.	44
4.17	Sintetizirana slika koja prikazuje jednostavnu scenu zatvorenog prostora i izlučeni pravocrtni segmenti nakon 3. iteracije postupka. Tamnijim pravocrtnim segmentima pridružena je veća težina.	45
4.18	Rezultati obrade slika jednostavnih poliedara. Obrada je trajala 11 (a) odnosno 15 iteracija (b). Tamnijim pravocrtnim segmentima pridružena je veća težina.	46
4.19	Rezultati obrade slika jednostavnih poliedara. Obrada je trajala 10 (a) odnosno 12 iteracija (b). Tamnijim pravocrtnim segmentima pridružena je veća težina.	47
4.20	Rezultati obrade dnevnih slika hodnika. Obrada je trajala 16 (a), odnosno 14 iteracija (b). Tamnijim pravocrtnim segmentima pridružena je veća težina.	48
4.21	Rezultati obrade noćnih slika hodnika. Obrada je trajala 12 (a), odnosno 14 iteracija (b). Tamnijim pravocrtnim segmentima pridružena je veća težina.	48
4.22	Prikaz toka obrade u jednoj iteraciji algoritma. Pravokutnici označavaju slike i strukture podataka, dok krugovi označavaju korake obrade (opširnije u tekstu).	49
5.1	Određivanje normale interpretacijske ravnine pravocrtnog segmenta \overline{AB} na temelju koordinata njegovih krajnjih točaka, po jednadžbi (5.1).	55
5.2	Pravocrtni segmenti s_1 i s_2 odgovaraju paralelnim bridovima scene b_1 i b_2 . Slika ilustrira postupak pronalaženja nedogleda pravocrtnih segmenata na temelju normala odgovarajućih interpretacijskih ravnina, u skladu sa jednadžbom (5.2).	56
5.3	Pravocrtni prikaz idealizirane scene hodnika sa označenim horizontom i nedogledima koji odgovaraju uzdužnoj (N1), poprečnoj (N2) i uspravnoj (N3) osi hodnika.	58
5.4	Prikaz toka obrade algoritma za određivanje dominantnog nedogleda vodoravnih bridova scene. Pravokutnici označavaju ulazne i izlazne strukture podataka te međurezultate, dok su koraci obrade predstavljeni krugovima (opširnije u tekstu).	59
5.5	Slika hodnika (lijevo), te rezultati obrade (desno). U desnom dijelu prikazani su hipotetski nedogledi vodoravnih bridova scene (križići), njihove nakupine određene postupkom grupiranja (kvadratići), pravocrtni segmenti koji definiraju dominantnu nakupinu (dužine) te slika horizonta (vodoravna dužina).	63
5.6	Rezultati obrade dviju slika koje su dobivene iz istog položaja hodnika kao i slika na sl. 5.5, ali za različite smjerove gledanja.	64
5.7	Rezultati obrade triju slika koje su dobivene iz istog položaja hodnika, za različite smjerove gledanja. Pokazane su iste značajke, kao i na sl. 5.5.	65
5.8	Rezultati obrade triju slika koje su dobivene iz istog položaja hodnika, za različite smjerove gledanja. Pokazane su iste značajke, kao i na sl. 5.5.	66
5.9	Rezultati obrade triju slika koje su dobivene iz istog položaja hodnika, za različite smjerove gledanja. Pokazane su iste značajke, kao i na sl. 5.9.	67
5.10	Blok-dijagram predložene lokalizacijske procedure.	70
5.11	Međusobni odnos koordinatnih sustava robota i slike.	71

5.12	Tlocrt Zavodskog hodnika sa ucrtanim položajima na kojima su provedeni eksperimenti te odgovarajućim orijentacijama robota ($\varphi_{0a}, \varphi_{0b}, \varphi_{0c}$).	72
5.13	Rezultati primjenjivanja predložene procedure na položajima koji su ucrtani na sl. 5.12.	74
A.1	Shematski prikaz sklopovlja u sustavu.	78
A.2	Grafički prikaz relacija je jedan od (a) i tri oblika relacije upotrebljava (b) među klasama, te relacije ovisi o među komponentama (c).	83
A.3	Arhitektura programskog podsustava za upravljanje kamerom.	85
A.4	Arhitektura programskog podsustava za prijenos slike.	89

Popis tablica

3.1	Neki modeli scena zatvorenog prostora	20
5.1	Rezultati postupka za različite kutove zakreta kamere, za položaje 1, 2 i 3 . .	68
5.2	Rezultati primjene postupka na skupu slučajnih pravocrtnih segmenata . .	69
5.3	Rezultati primjenjivanja predložene lokalizacijske procedure na položajima koji su prikazani na sl. 5.12	73
A.1	Klase izvornog kôda koje su pridružene pojedinim konceptima podsustava za upravljanje kamerom	84
A.2	Klase izvornog kôda koje su pridružene pojedinim konceptima podsustava za prijenos slike	88

Poglavlje 1

Uvod

Mnoge ljudske aktivnosti su usko povezane sa snalaženjem i kretanjem u prostoru. Ljudi se kreću između strojeva u industrijskim pogonima, žure se hodnicima poslovnih zgrada, upravljaju kamionima i poljoprivrednim strojevima na cestama odnosno oranicama. Zato je izgradnja tehničkih sustava sposobnih za samostalno snalaženje u prostoru vrlo atraktivan cilj. Međutim, iako su spomenute aktivnosti vrlo raznorodne, zajedničko im je to da ih ljudi relativno uspješno obavljaju, za sada uspješnije od tehničkih sustava.

Kod većine primjena, samostalni pokretni sustav se ne može kretati unaprijed isplaniranim putem zbog neočekivanih prepreka te nesigurnosti u mjerenju prijednog puta. Stoga je promjenu položaja odnosno kretanje redovito potrebno nadzirati sensorima. Glavno osjetilo koje čovjek koristi prilikom snalaženja u prostoru jest vid. Čini se stoga da je za postizanje sposobnosti snalaženja sličnih čovjekovima, ključno tumačenje trodimenzionalnih scena iz stvarnog svijeta analizom slika u vidljivom spektru. Ta ocjena vrijedi tim više ako se snalaženje odvija u okolišu koji je prvenstveno projektiran za potrebe ljudi.

Motivacija ovog rada je istraživanje mogućnosti izrade tehničkog sustava sposobnog za samostalno snalaženje u zatvorenom prostoru korištenjem video kamere kao jedinog senzora. Osnovni zahtjev koji takav sustav mora zadovoljavati jest mogućnost određivanja vlastitog položaja u odnosu na ostale objekte u okolišu. Cilj istraživanja je zato ostvarivanje tzv. lokalizacijske procedure kojom sustav može odrediti svoj položaj u prethodno neistraženom okolišu, metodama jednookog računarskog vida.

Razmatra se prvenstveno snalaženje u sustavu hodnika jer je to arhitektonsko rješenje vrlo često u modernim zgradama. Pretpostavlja se da u hodnicima u kojima se navigacija odvija nema navigacijskih znakova tj. specijalnih oznaka za olakšavanje snalaženja. Odabrani pristup se zato zasniva na određivanju smjera pružanja uzdužne osi hodnika, odnosno na nedogledu [81] kao značajci koja je svojstvena svim hodnicima.

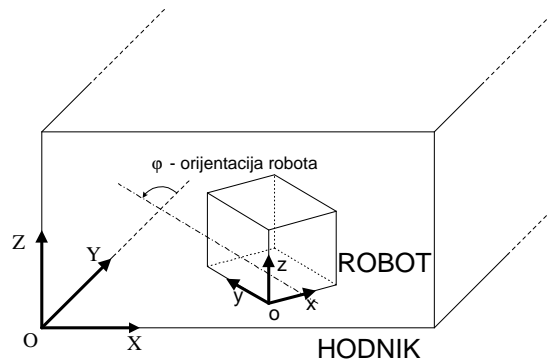
Za analizu trodimenzionalnih scena mogu se koristiti razni senzori: laserski i ultrazvučni određivači udaljenosti, infracrvene kamere, stereo parovi kamera i razni dodirni senzori. Svaki od njih ima svoja ograničenja i prednosti, međutim u ovom radu se koristi jedna video kamera kao relativno jednostavno i pristupačno rješenje koje pruža mnogo podataka uz nisku složenost upravljačkih algoritama.

1.1 Opis problema

Razmatra se samostalni pokretni robot¹ opremljen jednom video kamerom. Robot se nalazi u sustavu hodnika koji ima ortogonalnu triedarsku strukturu. Drugim riječima, granice objekata u okolišu su ravne plohe koje imaju jednu od tri međusobno okomite orijentacije. Kao referentni koordinatni sustav u takvom okolišu koristi se kartezijski koordinatni sustav

¹robot —uređj koji automatski obavlja složne i često ponavljajué radnje (Webster's dictionary of english language).

[34, 59] čije osi \mathbf{X} , \mathbf{Y} i \mathbf{Z} odgovaraju poprečnoj, uzdužnoj odnosno uspravnoj osi hodnika — sl. 1.1. Pretpostavlja se da je smjer uspravne osi hodnika u koordinatnom sustavu robota implicitno poznat jer pod hodnika podupire kućište robota pa se smjerovi uspravnih osi u sustavima robota i hodnika poklapaju. Alternativno, uspravna os hodnika se može odrediti pomoću jednostavnih mjernih uređaja kao lokalni smjer pružanja gravitacijskog polja. Pokazuje se da takve uređaje koriste i ljudi i mnoge životinje te da izmjereni smjer uspravne osi strukture okoliša koriste u postupku interpretacije scena [34].



Slika 1.1: Koordinatni sustavi hodnika ($\mathbf{O}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}$) i robota ($\mathbf{o}, \mathbf{x}, \mathbf{y}, \mathbf{z}$).

Kamera kojom je robot opremljen ima dva stupnja slobode — mogu joj se podešavati kutovi zakreta (*engl.* pan) i nagiba (*engl.* tilt) (sl. 1.2). Ti kutovi definiraju smjer gledanja kamere odnosno njenu orijentaciju u koordinatnom sustavu robota. Kamera je učvršćena



Slika 1.2: Stupnjevi slobode kamere.

na kućište robota tako da su za kutove zakreta i nagiba od 0° , osi kartezijskog koordinatnog sustava slike paralelne osima \mathbf{x} i \mathbf{z} koordinatnog sustava robota. Cilj istraživanja je razviti sustav računarskog vida koji će upotrebom takve kamere moći odrediti orijentaciju robota u hodniku čiji je model dan na sl. 1.1. Nažalost, stvarne scene u hodnicima su obično dovoljno dobro osvijetljene sa stajališta čovjeka, ali nejednoliko i loše sa stajališta moderne video kamere. Stoga se očekuje da uvjeti pribavljanja slike neće biti konstantni jer robot može biti proizvoljno orijentiran u različitim dijelovima okoliša. Samostalno snalaženje u prostoru nije interaktivna primjena pa tokom rada sustava vanjska intervencija u smislu podešavanja parametara nekog od algoritama nije prihvatljiva. Razvijeni sustav zato treba biti potpuno samostalan i robusan na slabu odnosno nepotpunu ulaznu informaciju.

1.2 Odabrani pristup

Budući da se analiziraju scene sa ravnim plohama i bridovima, kao međuprikaz između slikovnog i simboličkog opisa scene nameće se skup pravocrtnih segmenata. Promatrane scene imaju izraženu trodimenzionalnu strukturu pa se postupak dobivanja slike može modelirati perspektivnom projekcijom u skladu s kojom se paralelni bridovi scene preslikavaju u pravocrtne segmente slike koji se sijeku u nedogledu — točki slike koja je pridružena trodimenzionalnom smjeru bridova [54]. Odnose među geometrijskim značajkama u tako dobivenoj slici dobro opisuje projekcijska geometrija [22, 33].

Rezultati algoritama za prepoznavanje 3D objekata u pojedinačnim slikama [3, 28] jako ovise o kvaliteti izlučenih geometrijskih značajki u fazi pretprocesiranja ulaznih slika. Međutim, metode izlučivanja geometrijskih značajki često postižu zadovoljavajuće rezultate tek nakon prilagođavanja skupa parametara pojedinačnoj slici, pogotovo ako se radi o slikama nejednoliko osvijetljenih stvarnih scena [53]. Nemogućnost kontroliranja uvjeta pribavljanja slike i interaktivnog prilagođavanja parametara algoritama pribavljenim slikama uvelike usporavaju napredak na području samostalnih pokretnih tehničkih sustava. Rezultati istraživanja na područjima računarskog vida u kojima tih ograničenja nema (raspoznavanje pisanog teksta, analiza bioloških eksperimenata ili čak raspoznavanje izdvojenih 3D objekata [28]) su bitno bolji pa razvijeni tehnički sustavi postižu performanse bliže ljudskima.

Općenito se pokazuje da metode projektirane za primjene koje pretpostavljaju kontrolirane uvjete pribavljanja slike nije lako prilagoditi zadacima u nekontroliranom ili neistraženom okruženju. U takvim okruženjima potrebno je razviti posebne strategije za upravljanje postupcima pribavljanja i obrade slike. Primjena takvih strategija naziva se aktivnim vidom [11, 16, 64, 79, 88], a njima se osigurava slijedeće:

1. pribavljanje slike koja sadrži informaciju koja je korisna za snalaženje u prostoru (podešavanje parametara kamere — fokus, otvor objektiva, povećanje, nagib, zakret, itd.);
2. primjena prikladnog i ispravno konfiguriranog algoritma na pribavljenu sliku (podešavanje parametara algoritama — širina konvolucijske funkcije za glaćenje slike, vrijednosti praga za detektor rubova, granične udaljenosti za algoritam grupiranja, itd.);
3. sinkronizacija navedenih postupaka u cilju rješavanja tekućeg zadatka te odgovarajuća interpretacija rezultata.

Odabrani pristup stoga obuhvaća takve upravljačke strategije kako bi se zadaci mogli obavljati i u prethodno nepoznatom hodniku. Opisanim procedurom lokalizacije [50] robot stiče znanje o okolišu kojim se omogućuje bitno smanjivanje broja slika koje je potrebno analizirati kako bi se stiglo do rješenja složenog zadatka. Pored toga, pokazuje se da takvo znanje može poboljšati performansu algoritama za obradu slike i na nižim razinama sustava računarskog vida [34].

1.3 Moguće primjene

Pregledi primjena samostalnih sustava vođenih računarskim vidom su dani u [24] i [43]. U novije vrijeme, samostalno snalaženje u prostoru se primjenjuje u poljoprivredi [70], samostalnim cestovnim [57] i terenskim vozilima [60], sustavima za pomoć onesposobljenim osobama [67, 63], istraživanju nepoznatog okoliša [66], proizvodnji [18], rukovanju opasnim materijalima [46] i nadgledanju ratišta [59]. Upotreba samostalnih pokretnih sustava je posebno pogodna u primjenama kod kojih radna okolina nije prikladna za ljudsku aktivnost. Primjeri takvih okolina uključuju dubine oceana, geotermalne izvore, nuklearna postrojenja i svemir (orbitalne stanice, Mjesec, Mars) [43, 66, 82]. U skorij budućnosti

može se očekivati da će uređaji sposobni za autonomnu navigaciju doći u svakodnevnu upotrebu u obliku automatiziranih kućnih i poljoprivrednih strojeva [24].

Tehnika opisana u poglavlju 5 se može koristiti u sustavu za podešavanje optimalnog smjera vozila u cestovnom prometu [8]. Takav sustav bi mogao odrediti optimalan zakret upravljača na temelju slike snimljene kamerom koja je postavljena na prednjem kraju vozila. Primjena takvog sustava bi u početku bila usmjerena na pomoć vozačima dok su u perspektivi vozila koja bi mogla prometovati specijaliziranim prometnicama bez upravljanja od strane čovjeka [57]. Snalaženje u prostoru je nešto teže ostvariti kod terenskih vozila, jer se ona kreću u okolišu koji je manje jednoobrazan od asfaltne ceste. Baten et al. [60] su opisali primjenu u kojoj terensko vozilo, metodama stereo vida određuje vertikalni profil puta po kojem se kreće. Opisani sustav radi u stvarnom vremenu zahvaljujući specifičnom sklopovlju.

Ostvarivanje sustava za samostalno snalaženje je od centralnog značaja na području svemirskih istraživanja [66]. Sposobnost kretanja po površini Marsa bez ljudske posade je bila ključna za uspjeh NASAine misije “Mars Pathfinder” 1997. godine [82]. Tokom 83 marsovska dana koliko je trajala misija, vozilo “Sojourner rover” je obavljalo razna mjerenja i prikupljanja informacija na površini nepoznate planete. Skromne dimenzije vozila (duljina od 65 cm i masa od 10,5 kg) su uvjetovane zahtjevima svemirskog putovanja i ograničene snage napajanja. Zbog ograničenog dometa antena na Zemlji i napajanja vozila, veza sa Zemljom je ostvarivana samo jednom dnevno, kada bi astronomski odnosi Marsa i Zemlje bili najpogodniji za komunikaciju. Vozilo je tako svaki dan dobivalo skup naredbi kao što su fotografiranje, obavljanje eksperimenata, sakupljanje i transport uzoraka do planetarnog modula. Te naredbe je valjalo obaviti na siguran način, izbjegavajući stijene, odrone i prevelike strmine, upotrebom laserskih označivača i jedne video kamere. Ohrabrena velikim uspjehom misije iz 1997. godine, NASA planira slijedeće istraživanje Marsa 2001. godine koje podrazumijeva još veću samostalnost istraživačkog vozila.

Istraživači sa sveučilišta u Bonnu su uspješno pokazali mogućnost upotrebe autonomnog robota kao muzejskog vodiča [77]. Razvijeni robot “Rhino” ima 24 ultrazvučna, 56 infra-crvenih i 24 taktilna senzora, dva laserska određivača udaljenosti te stereo par CCD kamera. Prilikom kretanja, robot koristi zemljovid okoliša koji izrađuje samostalno, kada ga se kroz novo okruženje provede pod ljudskim upravljanjem.

1.4 Struktura rada

Opisuje se lokalizacijska procedura kojom se može odrediti apsolutna orijentacija samostalnog robota u sustavu hodnika. Pri rješavanju zadatka, primjenjuje se aktivni pristup analizi 3D scena zatvorenog prostora i projekcijska geometrija. Razmatraju se teoretski i praktični aspekti problema postavljenog u odjeljku 1.1 te procedura lokalizacije kao temeljni korak u njegovom rješavanju.

U poglavlju 2 su prikazani dosada postignuti rezultati na području samostalnog snalaženja u prostoru s posebnim naglaskom na lokalizaciju metodama računarskog vida. Teoretska osnova za rješavanje zadatka se sastoji od modela kamere i scene, teorema projekcijske geometrije te aktivnog vida, a izložena je u poglavlju 3. Poglavlje 4 sadrži detaljne opise pronalaženja težinskih rubnih elemenata i povezivanja rubova u težinske pravocrtne segmente. Postupak kojim se na temelju težinskog pravocrtnog prikaza određuje dominantni nedogled vodoravnih bridova scene, te njegova primjena u ostvarivanju procedure lokalizacije opisani su u poglavlju 5. Konačno, u poglavlju 6 se nalazi kratka rasprava o dobivenim rezultatima i mogućim smjerovima za daljnje istraživanje te zaključak.

Poglavlje 2

Pregled pristupa samostalnoj lokalizaciji

Pristupi snalaženju u prostoru pokretnih robota su dugo bili ograničeni na slijedenje ukopanih metalnih vodova odnosno crta na podu okoliša [14]. Takvi roboti se u literaturi često nazivaju automatski vođenim vozilima (*engl.* AGV — automatic guided vehicle [18, 62]). Radi se o vrlo pouzdanim i jednostavno ostvarivim tehnikama koje, međutim, jako ograničavaju gibanje pokretnog sustava. U posljednjem desetljeću, brojni su istraživači pokušavali kvalitativno poboljšati sposobnosti kretanja samostalnih robota. Rezultat tih napora su roboti sposobni za samostalnu navigaciju — dinamičko određivanje i korekciju puta koji robot prelazi kako bi obavio zadatak.

Leonard i Durrant-Whyte problem navigacije svode na slijedeća tri pitanja [23]:

1. Gdje sam?
2. Kuda idem?
3. Kako tamo doći?

U ovom radu razmatra se odgovor na pitanje 1: određivanje vlastitog položaja (lokalizacija) u okolišu čitanjem senzorskih ulaza i korištenjem apriornog znanja. Pitanje 2. se odnosi na odabir strategije putovanja, odnosno na određivanje slijedećeg u nizu međuodredišta kroz koje robot treba proći kako bi obavio zadatak. Planiranje kretanja je tema pitanja 3. — odabir trajektorije do slijedećeg međuodredišta uz izbjegavanje eventualnih pokretnih i nepokretnih prepreka.

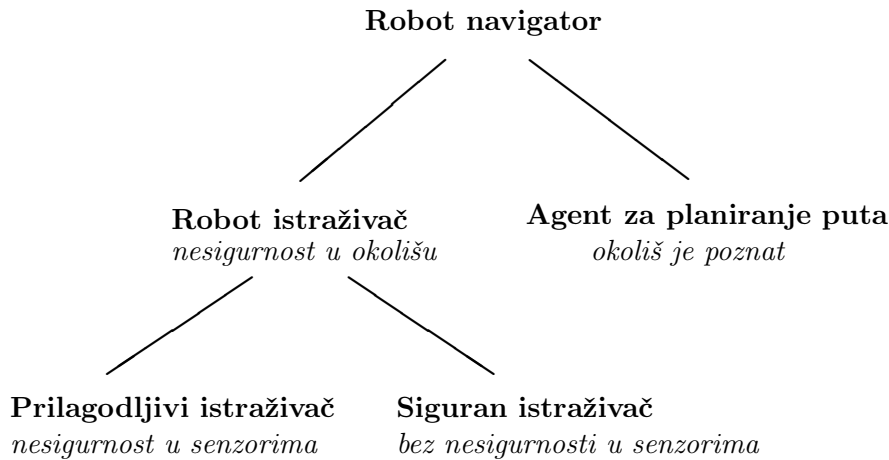
Ako se okoliš predstavi grafom [17] čiji čvorovi predstavljaju karakteristične lokacije u okolišu, a lukovi međusobnu prostornu povezanost lokacija, odabir strategije putovanja odgovara pronalasku puta u grafu koji omogućava uspješno obavljanje zadatka. U slučaju da je okoliš poznat, taj problem se svodi na traženje optimalnog puta u grafu [40]. Pregled metoda istraživanja grafa koji u početku nije poznat, se može naći u [25].

Tehnike planiranja kretanja [46] se općenito mogu podijeliti na egzaktnu i heurističku [27]. Heurističke metode su obično računski manje zahtjevne i zato šire primjenljive od egzaktnih. Međutim, egzaktnu metode imaju teorijsku važnost jer određuju mjere složenosti pojedinih algoritama odnosno zadataka. Rao daje pregled [36] egzaktnih metoda za određivanje putanje robota u neistraženom okolišu.

U literaturi se uglavnom obrađuju pojedinačna pitanja samostalne navigacije, a rijetko se nudi cjelokupno rješenje. Jedna od iznimki je rad Taylora i Kriegmana [87] koji opisuju robotski sustav sposoban za istraživanje zatvorenog prostora. Težište tog rada ipak je na algoritmima za određivanje strategije putovanja i planiranja kretanja (pitanja 2. i 3.) jer je odabrana jednostavna procedura lokalizacije koja se temelji na prepoznavanju umjetnih referentnih objekata.

2.1 Klasifikacija sustava za samostalnu navigaciju

Betke [41] klasificira zadatke koji se rješavaju samostalnom navigacijom na temelju nesigurnosti koja je uz njih vezana. Različiti robotski sustavi se razlikuju po tome koliko apriornog znanja o okolišu imaju na raspolaganju — sl. 2.1. Što više takvog znanja robot ima, to više svojih akcija može isplanirati unaprijed.



Slika 2.1: Klasifikacija robota prema nesigurnosti s kojom se susreću.

U ekstremnom slučaju, apriorno znanje robota potpuno opisuje odnosno modelira okoliš. Takvi roboti su nazvani “agentima za planiranje puta” (npr. [59]) jer nakon početne lokalizacije mogu u potpunosti odrediti svoj put i prije početka fizičkog kretanja kroz okoliš. Pretpostavka da robot ima kompletnu informaciju o okolišu međutim ne vrijedi za velik broj potencijalnih primjena u kojima robot ima vrlo ograničenu mogućnost planiranja puta unaprijed. Takav robot se naziva “istraživačem” jer znanje o okolišu stiže u dijelovima, tokom obavljanja zadatka. Nakon svakog novog saznanja, robot može donositi odluke na temelju cjelokupnog akumuliranog znanja. Razlikuju se “sigurni” i “prilagodljivi” istraživači. Sigurni istraživač (npr. [90]) se može potpuno pouzdati u svoje senzore pa može slijepo vjerovati informacijama koje je prikupio tokom navigacije. To nije slučaj s “prilagodljivim” istraživačem (npr. [79]) koji “zna” da prikupljene informacije o okolišu stvarni svijet opisuju neprecizno. Kako vrijeme prolazi, model okoliša koji takav robot gradi može divergirati od stvarnog svijeta. Zato prilagodljivi istraživač mora periodički provoditi “test stvarnosti” kako bi provjerio konzistenciju svog modela te popravljati model u skladu s rezultatima testa.

Stvarni robot nije nikad siguran istraživač jer se uvijek susreće sa greškama u sensorima zbog raznih vrsta šuma, proklizavanja kotača i sl. Međutim, model sigurnog istraživača može biti koristan ako se želi apstrahirati izvedbene detalje i koncentrirati se na navigacijske zadatke na hijerarhijski višoj razini. Zato sigurni istraživači često prikazuju okoliš kvalitativnim, topološkim modelom (grafom). Za razliku od njih, prilagodljivi istraživači svoje znanje o okolišu obično prikazuju geometrijskim modelom. Takav, kvantitativni model je detaljniji, ali i teži za izgraditi, upotrebljavati i usaglašavati sa novim saznanjima.

Procedura lokalizacije je neizostavni dio svakog od modela sa sl. 2.1. Međutim, složenost izvedbe, učestalost upotrebe i važnost lokalizacije u odnosu na druge komponente navigacijskog sustava su tim veće što je veća nesigurnost s kojom se robot susreće. Zato se problem lokalizacije u slijedećim poglavljima razmatra u kontekstu robota koji je prilagodljivi istraživač.

2.2 Tehnike lokalizacije

Određivanje položaja robota u okolišu (lokalizacija) se u literaturi označava raznim nazivima: *lociranje* (*engl.* positioning), *određivanje položaja* (*engl.* position measurement [62]), *ocjena položaja* (*engl.* position estimation [41, 59]), *lokalizacija* (*engl.* localisation [23, 32, 61, 79, 86], self-localisation [65]). Raznolikost u nazivlju odražava raznolikost pristupa lokalizaciji. U svom opširnom preglednom članku, Borenstein [62] konstatira da, iako postoji mnogo djelomičnih rješenja, istinski elegantno rješenje problema još nije pronađeno. Dobra osobina postojećih tehnika lokalizacije je što se međusobno ne isključuju pa se mogu obavljati istovremeno, a njihovi rezultati kombinirati. Pokazuje se općenito da se takvom kombinacijom više mjernih tehnika mogu dobiti rezultati bolji od rezultata bilo koje od pojedinačnih tehnika [84, 91].

Pristupi lokalizaciji se mogu općenito podijeliti na relativne i apsolutne [62]. Kod relativne lokalizacije, položaj robota se određuje u odnosu na neki referentni položaj u kojem se robot prethodno nalazio. Za razliku od relativne, apsolutna lokalizacija podrazumijeva određivanje položaja u referentnom koordinatnom sustavu svijeta. Metode relativne lokalizacije uključuju mjerenje prijeđenog puta i inercijsku navigaciju. Određivanje apsolutne orijentacije, prepoznavanje aktivnih ili pasivnih referentnih objekata u okolišu te izrada i korištenje modela strukture okoliša su glavni pristupi apsolutnoj lokalizaciji. Zbog nedostatka ustoličene “najbolje” metode lokalizacije, u postojećim sustavima za samostalnu navigaciju i automatski vođenim vozilima, uglavnom se kombiniraju jedna relativna i jedna apsolutna tehnika lokalizacije.

Formalno, rezultat apsolutne lokalizacije je transformacija koja preslikava 3D koordinatni sustav robota u 3D koordinatni sustav svijeta. Sasvim općenito, ta transformacija ima šest stupnjeva slobode: tri translacijska i tri rotacijska. Talluri i Aggarwal [59] međutim tvrde kako se većina postojećih robota kreću po vodoravnoj podlozi. Takvi roboti se razmatraju i u ovom radu. Pod tim ograničenjem, transformacija se svodi na ravninsko preslikavanje pa se može opisati sa dva translacijska i jednim rotacijskim parametrom. Položaj (*engl.* position) robota u užem smislu riječi se stoga definira kao uređeni par (x, y) koji određuje translaciju robota u ravnini kretanja, dok se orijentacija (*engl.* pose) robota odnosi na njegovu rotaciju (φ) oko uspravne osi okoliša (sl. 1.1).

2.2.1 Mjerenje prijeđenog puta

Mjerenje prijeđenog puta je tehnika kojom se promjena položaja robota određuje praćenjem okretanja kotača. To je najšire korištena lokalizacijska tehnika — pruža relativno dobru kratkoročnu preciznost, relativno je jeftina i dozvoljava visoku učestalost uzorkovanja. Međutim, kao i svaki drugi relativni pristup lokalizaciji, svodi se na integraciju inkrementalnih pomaka u vremenu pri čemu se porast mjerne nesigurnosti položaja robota ne može kontrolirati. Posebno su problematične greške u orijentaciji jer vode velikim bočnim greškama u položaju koje rastu proporcionalno s prijeđenim putem. Unatoč ograničenjima, većina istraživača se slaže kako se radi o vrlo važnoj tehnici lokalizacije, pogotovo u kombinaciji s nekom od apsolutnih tehnika. Mjerenje prijeđenog puta se zasniva na jednostavnim jednadžbama koje važe ako okretanje kotača odgovara prijeđenoj pravocrtnoj udaljenosti. Uzroci grešaka u takvom proračunu mogu biti sistematski (npr. nejednaki dijametri kotača) i slučajni (izbočine i pukotine u podlozi te proklizavanje kotača). Tipično, u sustavu u kojem se kombinira mjerenje prijeđenog puta s prepoznavanjem referentnih objekata, potrebna učestalost prepoznavanja navigacijskih objekata za uspješnu navigaciju se određuje empirijski, na temelju procijenjene sistematske greške u najgorem slučaju.

2.2.2 Inercijska navigacija

Inercijska navigacija koristi akcelerometre i žiroskope kako bi se odredile promjene brzine odnosno rotacije sustava. Novi položaj robota se određuje dvostrukom integracijom promjena brzine, dok se integriranjem rotacijskih pomaka dobiva nova orijentacija. Kao i kod prethodne tehnike, bilo kakva mala konstantna pogreška u mjerenju neograničeno raste nakon integracije u vremenu. Zato ni ova tehnika nije prikladna za upotrebu tokom dužeg vremenskog intervala.

2.2.3 Određivanje apsolutne orijentacije

Orijentacija robota je najvažniji parametar njegovog položaja što se tiče njegovog utjecaja na akumuliranu grešku mjerenja prijeđenog puta. Zato su senzori koji pribavljaju takvu informaciju iznimno važni. Senzori za određivanje apsolutne orijentacije se mogu podijeliti na geomagnetske senzore (npr. magnetski kompas) i razne vrste žiroskopa. Nedostatak geomagnetskih senzora je u tome što je magnetsko polje zemlje često izobličeno u blizini energetskih vodova i čeličnih struktura. To bitno komplicira njihovu upotrebu u zatvorenim prostorima. Žiroskopi nemaju takve poteškoće, međutim sve do nedavno, njihova cijena je bila previsoka za primjenu u autonomnoj navigaciji. U posljednje vrijeme na tržištu se pojavilo nekoliko manje skupih modela komercijalnih žiroskopa čija cijena iznosi oko 1000 američkih dolara [50].

2.2.4 Lokalizacija uz pomoć aktivnih referentnih objekata

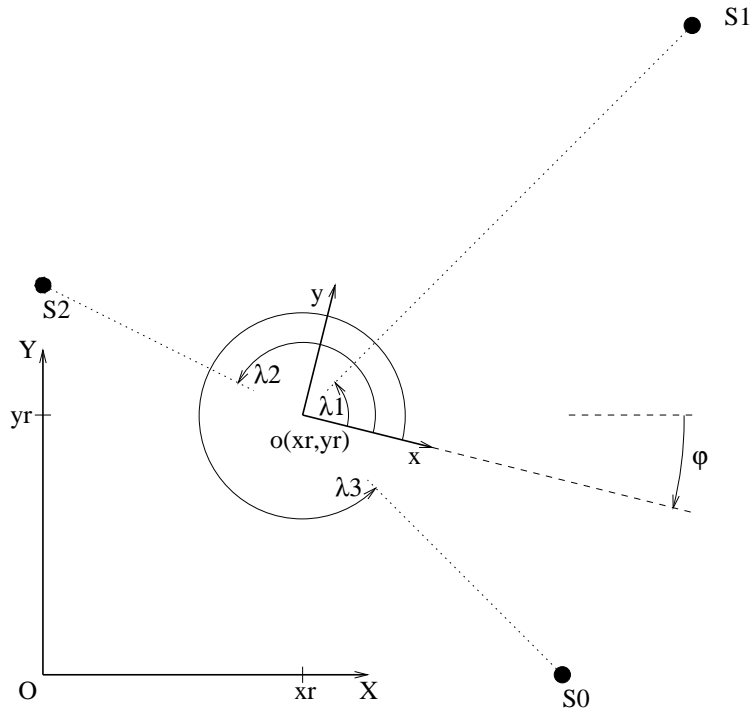
Aktivni referentni objekti¹ ili svjetionici (*engl.* active beacon) su najčešće izvori elektromagnetskog zračenja poznatih karakteristika. Aktivni sustavi za lokalizaciju objedinjuju više takvih objekata sa svrhom omogućavanja navigacije u nekom širem okolišu. Svjetionici su najčešća navigacijska pomagala kod brodova i zrakoplova, a koriste se i kod komercijalnih robota. Mogu se pouzdano detektirati i pružaju precizne lokalizacijske podatke uz visoku učestalost uzorkovanja i minimalno procesiranje. S druge strane, ovakav pristup je povezan s visokim troškovima gradnje i održavanja svjetionika, a ostvareni sustavi su neupotrebljivi u okolišima bez navigacijske infrastrukture. Razlikuju se dva temeljna pristupa lokalizaciji na temelju poznatih položaja referentnih objekata: trijangulacija i trilateracija.

Lokalizacija trijangulacijom

Kod ovog pristupa, tri ili više svjetionika su postavljeni na poznatim lokacijama u okolišu. Robot ima rotirajući senzor koji mu omogućuje prepoznavanje pojedinih svjetionika i određivanje njihovih azimuta (kutova pod kojima se svjetionici “vide”) — sl. 2.2. Iz tih mjerenja može se odrediti nepoznat položaj robota (x_R, y_R, φ) .

Trijangulacija je posebno važna tehnika jer se može primijeniti i kod lokalizacije u odnosu na percipirane pasivne referentne objekte. Tada je obično na raspolaganju veći broj referentnih objekata čiji identitet i azimut nisu sasvim točno određeni. Stoga se u literaturi mogu naći razne varijante osnovnog problema trijangulacije. Tako Betke i Gurvits [61] predlažu algoritam za rješavanje poopćenog problema trijangulacije koji glasi: odredi svoj položaj na temelju položaja referentnih objekata S_1, S_2, \dots, S_n ako su zadani odgovarajući azimuti $\lambda_1, \lambda_2, \dots, \lambda_n$. Algoritam ima linearnu složenost i dobro se ponaša u slučaju velikih grešaka u azimutima pojedinih objekata. Sugihara [15] je prethodno rješavao sličan problem u kojem se međutim referentni objekti međusobno ne razlikuju. Složenost predloženog algoritma je $O(n^3 \cdot \lg n)$.

¹Aktivni referentni objekti nisu povezani s aktivnom percepcijom. Upravo suprotno, u sustavima lokalizacije koji se temelje na aktivnim referentnim objektima, obično nema potrebe za aktivnom percepcijom.



Slika 2.2: Osnovni problem trijagulacije: robot određuje svoj položaj (x_R, y_R, φ) u koordinatnom sustavu svijeta $(\mathbf{O}, \mathbf{X}, \mathbf{Y})$ na temelju azimuta $\lambda_0, \lambda_1, \lambda_2$ svjetionika S_0, S_1, S_2 u koordinatnom sustavu robota $(\mathbf{o}, \mathbf{x}, \mathbf{y})$.

Lokalizacija trilateracijom

Postupak trilateracije je određivanje položaja robota na temelju izmjerenih udaljenosti do referentnih objekata. Ta tehnika se koristi u satelitskim sustavima (GPS — *engl.* Global Positioning System) za određivanje globalnog položaja na Zemlji. Takvi sustavi su vrlo obećavajuće rješenje za navigaciju na širem otvorenom prostoru jer nude preciznost lokalizacije reda veličine 5 m [50]. Trilateracija se može koristiti i u kombinaciji sa pasivnim navigacijskim objektima koji se u okolišu pronalaze stereo vidom ili ultrazvučnim odnosno laserskim određivačima udaljenosti.

2.2.5 Prepoznavanje pasivnih referentnih objekata

Pasivni referentni objekti (*engl.* landmark) su izdvojeni objekti ili značajke okoliša koje robot može prepoznati analizom senzorskih ulaza, a koriste se tijekom lokalizacije kao referentne točke. Prije lokalizacije, u memoriji robota moraju biti pohranjeni položaji i osobine (modeli) pojedinih referentnih objekata. Problem lokalizacije se tada svodi na pouzdano percipiranje objekata u okolišu i primjenu jedne od tehnika koje su opisane u pododjeljku o aktivnim sustavima lokalizacije. Kako bi se pojednostavnio problem prepoznavanja, često se pretpostavlja da je trenutni položaj robota približno poznat tako da robot treba tražiti navigacijske objekte u ograničenom prostoru. Zato je ovu tehniku pogodno kombinirati sa relativnom lokalizacijom mjerenjem prijašnjeg puta.

Opća lokalizacijska procedura na temelju prepoznavanja referentnih objekata u okolišu se može opisati sa slijedeća četiri koraka [41]:

1. prepoznavanje potencijalnih referentnih objekata u okolišu;
2. podudaranje prepoznatih objekata sa pohranjenim modelima;
3. određivanje azimuta prepoznatih objekata;

4. određivanje položaja poopćenom metodom trijangulacije.

Najveću poteškoću u postupku predstavlja naravno prepoznavanje objekata iz okoliša te njihovo podudaranje sa nekim od zadanih modela.

Razlikuju se dvije vrste pasivnih referentnih objekata u okolišu — prirodni (*engl.* landmark), i umjetni ili navigacijski znakovi (*engl.* artificial landmark, beacon). Prirodni referentni objekti se nalaze u okolišu potpuno neovisno o zadatku kojeg robot rješava. Za razliku od njih, navigacijski znakovi se postavljaju u okoliš isključivo s namjerom da olakšaju navigaciju; glavni kriterij prilikom njihovog oblikovanja je laka prepoznatljivost. Senzor izbora za lokalizaciju prepoznavanjem referentnih objekata jest računarski vid. Više riječi o takvim sustavima lokalizacije će stoga biti u slijedećem odjeljku.

Prednost prirodnih referentnih objekata je u tome što ne zahtijevaju modificiranje okoliša. S druge strane, navigacijske znakove je lakše prepoznati i međusobno razlikovati, a bitno su jeftiniji od svjetionika. Maksimalna udaljenost od pasivnog referentnog objekta do robota je međutim suštinski kraća nego kod aktivnih sustava. Prepoznavanje pasivnih referentnih objekata u okolišu dosta ovisi o uvjetima u kojima se navigacija odvija (osvjetljenje, prisutnost ljudi i drugih robota i sl.). Isto tako, povoljno je znati približan početni položaj kako bi robot otprilike znao gdje treba tražiti referentne objekte. Ako početni položaj nije poznat, potrebno je obavljati vremenski zahtjevno iscrpno pretraživanje okoliša. Nadalje, takav sustav za lokalizaciju treba održavati neku vrst zemljovida — bazu podataka objekata i njihovih položaja u okolišu. Iz tih razloga, dosada je razvijen vrlo mali broj komercijalnih sustava za samostalnu navigaciju prepoznavanjem prirodnih objekata u okolišu.

2.2.6 Modeliranje strukture okoliša

Lokalizacija modeliranjem strukture okoliša je tehnika u kojoj robot koristi svoje senzore kako bi kreirao zemljovid susjednih područja okoliša. Tako dobiveni lokalni model se zatim uspoređuje s dijelovima globalnog modela okoliša koji se nalazi u memoriji robota. Ako usporedba uspijeva, robot je odredio svoj položaj u okolišu. Pretraživanje globalnog zemljovida se može ograničiti ako je približni položaj robota poznat te ako se greške u postupku određivanja tog položaja i građenja lokalnog zemljovida mogu modelirati. Dva su suštinski različita pristupa lokalizaciji korištenjem zemljovida. Globalni zemljovid okoliša može biti zadan prije početka navigacije ili se može graditi u toku gibanja. Robot kojim zemljovid okoliša nije unaprijed u potpunosti zadan je robot istraživač (odjeljak 2.1). Takav robot, pored snalaženja u prostoru ucrtanom u zemljovid, mora se znati snaći i u nedokumentiranim dijelovima okoliša te svoja zapažanja ucrtati na odgovarajuća mjesta u zemljovidu.

Prednost ovakvog pristupa je u tome što se, kao i kod prepoznavanja prirodnih objekata u okolišu, ne zahtijeva mijenjanje okoliša. Nadalje, postojeće znanje se može nadopunjavati pa sustav može biti otporan na promjene u okolišu. Dobiveni zemljovid se može primijeniti i pri rješavanju drugih navigacijskih zadataka — odabiru strategije putovanja i planiranju kretanja. Konačno, izrada zemljovida omogućava učenje novog okoliša te poboljšanje rezultata navigacije istraživanjem. Lokalizaciju modeliranjem okoliša je posebno pogodno vršiti u zatvorenom prostoru, upotrebljavajući stereo vid ili višestruke ultrazvučne odnosno laserske određivače udaljenosti kao senzore. U tom slučaju je za zemljovid prikladno upotrijebiti tlocrt okoliša.

Ovakav pristup se međutim može upotrijebiti samo u visoko strukturiranim okolišima sa dovoljno nepokretnih i lako prepoznatljivih značajki za podudaranje. Nadalje, senzori moraju omogućiti dovoljno precizno prepoznavanje strukture okoliša što se može kositi sa odvijanjem navigacije u realnom vremenu.

Opisani pristup je sličan lokalizaciji prepoznavanjem referentnih objekata utoliko što se u oba pristupa vrši usporedba senzorskih ulaza sa pohranjenim modelom okoliša. Kod

navigacijskog zemljovida, međutim, vrši se analiza i podudaranje strukture okoliša, dok se u prethodnom pristupu prepoznaju diskretni, izdvojeni objekti u okolišu. Postoje i hibridni pristupi u kojima se izlučeni elementi strukture okoliša (npr. uspravni pravocrtni segmenti [15]) tretiraju kao referentni objekti.

2.3 Lokalizacija upotrebom računarskog vida

U knjizi koja je posvećena tehnikama lokalizacije samostalnog pokretnog robota, Borenstein et al. [50] tvrde kako je, od svih senzora koji se danas koriste, računarski vid potencijalno najmoćniji izvor informacija za samostalnu navigaciju. Međutim upravo zbog bogatstva dobivene informacije, izlučivanje vizualnih značajki za određivanje položaja nije lagan zadatak. Lokalizacija računarskim vidom se svodi na određivanje podudarnosti između izlučenih značajki iz 2D slika i strukture 3D okoliša. Temeljne komponente procesa lokalizacije računarskim vidom su stoga:

1. prikaz okoliša,
2. model percepcije i
3. algoritam za pronalaženje podudarnosti.

Prikaz okoliša može biti topološki ili geometrijski. Topološki prikaz okoliša jest graf u kojem čvorovi odgovaraju karakterističnim lokacijama okoliša, dok lukovi obično opisuju njihovu prostornu povezanost. Geometrijski prikaz okoliša opisuje kvantitativne geometrijske odnose među lokacijama okoliša i može biti kontinuiran ili diskretan. Diskretni prikaz okoliša se izvodi tablicom koja povezuje modele referentnih objekata s njihovim položajima u prostoru, a koristi se kada se u okolišu nalazi veći broj karakterističnih, lako prepoznatljivih objekata ili značajki (npr. stropna rasvjeta). Kada takvih objekata nema a struktura okoliša je poznata, prikladno je okoliš opisati kontinuiranim prikazom i prepoznati elemente njegove strukture (zidove, strop, podlogu, profil terena).

Modeli percepcije i algoritam podudaranja se odabiru na temelju odabranog prikaza okoliša odnosno skupa referentnih objekata koje treba prepoznati u cilju lokalizacije. Najčešće nije dovoljno samo analizirati informacije koje se dobivaju putem senzora nego je potrebno i aktivno sudjelovati u dobivanju informacija, postupcima aktivnog vida Razlog tome je što se u svakom stvarnom okolišu nalazi vrlo velika količina vizualnih informacija koje su irelevantne za lokalizaciju.

Glavni pristupi lokalizacije pomoću računarskog vida su prepoznavanje referentnih objekata i modeliranje strukture okoliša. Te metode su u širem kontekstu već opisane u prošlom odjeljku pa će u slijedeća dva pododjeljka biti detaljnije opisane neke izvedbe iz literature koje se temelje na računarskom vidu.

2.3.1 Lokalizacija prepoznavanjem referentnih objekata

Referentni objekti su u prethodnom odjeljku podijeljeni na prirodne referentne objekte i navigacijske znakove. Prirodni referentni objekti pri tome mogu biti stvarni trodimenzionalni objekti iz okoliša ili 2D značajke slike koje se na neki način ističu i izdvajaju od susjednih područja.

Navigacijski znakovi

Zbog jednostavnosti primjene, ovakav pristup je pogodan u razvojnim sustavima robotske navigacije u kojima lokalizacija nije glavni predmet istraživanja. Taylor i Kriegman [87] opisuju postupke odabira strategije putovanja i planiranja kretanja robota koji je siguran istraživač. Opisani postupci pretpostavljaju da robot može slijediti rub prepreka te odrediti

svoj položaj u okolišu metodom prepoznavanja referentnih objekata. Kako bi iskušali opisane postupke, autori su oblikovali jednostavnu metodu lokalizacije prepoznavanjem navigacijskih znakova. Navigacijski znakovi su pravokutnog oblika površine $\sim 0,25$ m, a sastoje se od crnog linijskog kôda na bijeloj pozadini. Na svakoj izdvojenoj prepriči okoliša (izdvojena prepreka se može običi praćenjem ruba) je postavljen jedan ili više navigacijskih znakova; granica prostorije se također tretira kao izdvojena prepreka. U fazi istraživanja, robot gradi kvalitativni topološki prikaz okoliša u kojem su izdvojene prepreke opisane čvorovima usmjerenog grafa. Čvor A je povezan sa čvorom B grafa ako se sa ruba prepreke A može vidjeti navigacijski znak na prepriči B. Upotrebom predloženih tehnika, robot je u vremenu od pola sata uspio istražiti okoliš od 300 m^2 u kojem je prethodno postavljeno 11 navigacijskih znakova.

Clarke [78] opisuje metodu lokalizacije uz pomoć navigacijskih znakova za navigaciju u zatvorenim prostorima. U predloženom pristupu, navigacijski znakovi se postavljaju na dobro vidljivim i često posjećivanim mjestima u okolišu pa predloženom metodom robot može periodički popravljati greške nagomilane nenadziranim mjerenjem prijeđenog puta. Tvrdi se kako navigacijski znakovi moraju zadovoljavati slijedeće donekle oprečne kriterije:

- trebaju biti lako prepoznatljivi iz raznih smjerova pogleda (tj. znakovi trebaju biti što više simetrični);
- u cilju lokalizacije, normala ravnine znaka se mora moći čim preciznije i pouzdanije odrediti (tj. znakovi trebaju imati što manje simetrija).

Većina prethodnih pristupa [50] je zanemarivala zahtjev (2) pa su oblikovani navigacijski znakovi bili invarijantni na refleksije i neke rotacije i robot nije mogao odrediti normalu ravnine navigacijskog znaka. Predlaže se rješenje u kojem se navigacijski znakovi sastoje od obojenih uzoraka simetričnih geometrijskih likova. Pri tome simetričan oblik likova omogućava jednostavno određivanje položaja znaka dok asimetrični obojeni uzorak služi za prepoznavanje pojedinačnog znaka i njegove orijentacije. Eksperimentalni rezultati pokazuju da se oblikovani znakovi mogu pouzdano detektirati jednom kamerom s udaljenosti do 5 m i pod kutem pogleda prema normali znaka do 80° .

Prepoznavanje objekata iz okoliša

U svojoj doktorskoj disertaciji, Betke [41] predlaže i postupak za prepoznavanje referentnih objekata u okolišu. Opisanim postupkom, položaji traženih objekata u ulaznoj slici se određuju na temelju biblioteke slika kojima su objekti modelirani. Za prepoznavanje se koristi metoda podudaranja sa uzorkom koji se iz modela dobiva transformacijom sa tri stupnja slobode: rotacijom te uvećanjima po x odnosno y osi. Postupkom pronalaženja uzorka koji se najbolje poklapa s nekim dijelom slike upravlja iterativni optimizacijski postupak poznat kao brzo simulirano kaljenje. Eksperimentalni rezultati su pokazali da je optimizacijski postupak brži od iscrpnog traženja svih mogućih uzoraka zadanog modela u svim dijelovima slike više tisuća puta (složenost iscrpnog traženja je $O(n^5)$). Postupak je testiran na prepoznavanju saobraćajnih znakova pod realističnim uvjetima. U svim eksperimentima kod kojih je model objekta koji se nalazi u slici prethodno zadan, dobiveni su ispravni rezultati. U općenitijim eksperimentima, kad model znaka koji se nalazi u slici nije zadan, postupak je dao pogrešan odgovor u 3 od ukupno 144 eksperimenata, jer mjere poklapanja za različite modele nisu sasvim usklađene. Vrijeme obrade jedne slike iznosi oko 15 s.

Dulimarta i Jain [65] opisuju lokalizaciju robota prepoznavanjem stropne rasvjete i pločica sa brojem na vratima prostorija u sustavu hodnika. Jednostavnim tehnikama za obradu slike, dobivene su brze tehnike lokalizacije koje se mogu koristiti i u stvarnom vremenu. Primjerice, određivanje položaja svjetala na stropu okoliša se izvodi analizom gradijenta u samo tri retka pribavljene slike, čime se postiže ukupno vrijeme obrade od

50 ms. U proceduri za pronalaženje pločice sa brojem na vratima prostorije, prvo se segmentira područje pločice u izvornoj slici, zatim se pronalazi kontura pločice i opisuje se lančanim kôdom, dok se na kraju određuje položaj pločice pronalaženjem pravih kutova u lančanom kôdu. Nakon pronalaženja položaja pločice, pristupa se prepoznavanju broja na pločici metodom podudaranja s uzorcima iz baze podataka. Ukupno vrijeme prepoznavanja broja na pločici iznosi 1,3 s što po autorima zadovoljava kriterije obrade u stvarnom vremenu. Lokalizacijska procedura je isprobana u realističnim uvjetima. Robot s kojim su obavljani eksperimenti je opremljen s dvije nepokretne kamere od kojih je jedna usmjerena prema gore (za detekciju stropne rasvjete), a druga u stranu (za prepoznavanje pločica na vratima prostorija). Odabrana konfiguracija kamera i algoritama za obradu slike s jedne strane omogućava primjenu jednostavnih upravljačkih algoritama, ali s druge strane jako ograničava područje primjene robotskog sustava.

Referentne značajke

Sim i Dudek [86, 95] rješavaju problem učenja novog okoliša kod agenta za planiranje puta, koncentriranjem pažnje na markantne dijelove okoliša. U predloženom pristupu lokalizaciji se tako izbjegavaju pretpostavke ovisne o okolišu u kojem se navigacija odvija. U fazi učenja, robot se upoznaje s novim okolišem tako da mu se predoči niz slika dobivenih iz raznih položaja u okolišu. Alternativno, može se reći da se robot pod pratnjom čovjeka vodi kroz okoliš s kojim se treba upoznati. Kandidati za referentne značajke se odabiru kao pravokutne regije izvornih slika u kojima vrijednost tzv. funkcije jedinstvenosti (funkcija jedinstvenosti se računa kao gustoća rubnih elemenata u regiji) poprima lokalni maksimum. Time se pažnja sustava usmjerava na područja slike koja se izdvajaju od ostalih. Odabrane regije se zatim grupiraju u skupove koji odgovaraju raznim pogledima na isti dio okoliša, a skupovi se na kraju prikazuju u parametarskom prostoru i spremaju za daljnje korištenje. U fazi navigacije, lokalizacija se obavlja na temelju jedne slike okoliša, slijedećim postupkom. Kandidati za referentne značajke se izdvajaju iz izvorne slike korištenjem istog modela pažnje kao i u fazi učenja. Svaki kandidat se uspoređuje sa parametarskim opisom naučenih značajki te se na temelju uspješnosti podudaranja generiraju procijenjeni položaj robota i njegova nesigurnost. Procjene položaja koje su dobivene za različite kandidate se Kalmanovim filterom kombiniraju u konačnu ocjenu položaja robota. Autori su opisali uspješne eksperimente na složenim slikama zatvorenog prostora, međutim nisu naveli performansu sustava.

Davison i Murray [79] također opisuju metodu lokalizacije prepoznavanjem karakterističnih pravokutnih regija izvorne slike aktivnim vidom. Njihov pristup međutim rješava općenitiji problem robota — prilagodljivog istraživača, praćenjem manjeg broja značajki kroz sljed slika stereo vidom. Autori tvrde kako se značajke koje odgovaraju dijelovima slika dimenzija 15×15 slikovnih elemenata zajedno sa jednostavnim korelacijskim postupkom podudaranja mogu koristiti za lokalizaciju u širokom području okoliša. Prikladne značajke u okolišu se odabiru metodom iz literature koja brzo pronalazi homogene regije s velikim kontrastom u odnosu na pozadinu. Značajan problem u takvom pristupu predstavljaju značajke uzrokovane prekrivanjem objekata i refleksijama jer se očekuje da će biti vidljive samo iz malog broja položaja u okolišu. Zato se svakoj značajki pridjeljuje tzv. područje vidljivosti na temelju položaja robota iz kojeg je značajka prvi put prepoznata. Ako se značajka ne prepozna iz većeg broja položaja u području vidljivosti, odbacuje se kao nepouzdana. Najveći problem autorima je stvaralo pronalaženje pouzdanih značajki. Zato su tokom eksperimenata učvrstili više bijelih listova papira formata A4 na vertikalne plohe u okolišu i tako olakšali lokalizaciju. Tokom navigacije robot podešava nagib i zakret svojih kamera kako bi pouzdanu značajku doveo na presjecište osi projekcija kamera. U takvoj konfiguraciji udaljenost značajke se vrlo jednostavno određuje pa robot može odrediti svoj položaj u odnosu na značajku. Budući da se radi o pouzdanoj značajci, njeni

prethodni položaji su poznati pa se određivanje položaja poboljšava Kalmanovim filterom. Tijekom navigacije, robot gradi i održava matricu stanja svijeta u kojoj se nalaze položaji i nesigurnosti položaja robota i pouzdanih značajki koje su otkrivene tokom navigacije. Tako se sposobnošću robota da odredi svoj točan položaj u odnosu na već videne značajke omogućuju dugi vremenski intervali precizne navigacije. Opisana procedura je primijenjena za navigaciju u laboratorijskom okruženju pri brzini od $20 \frac{cm}{s}$ i frekvenciji praćenja jedne značajke od 5 Hz.

2.3.2 Lokalizacija modeliranjem strukture okoliša

Lebeque i Aggarwal [34] razmatraju interpretaciju slika dobivenih jednom kamerom u cilju omogućavanja navigacije pokretnog robota. Temeljna pretpostavka je da okoliš ima pravilnu strukturu u kojoj su svi bridovi dijelovi pravaca i imaju jednu od nekoliko apriorno poznatih 3D smjerova. To ograničenje je iskorišteno i u proceduri za izlučivanje pravocrtnih segmenata iz slike pa su dobiveni vrlo kvalitetni pravocrtni prikazi izvornih slika. Opisani su algoritmi kojima se gradi geometrijski model okoliša na temelju interpretacije jedne ili više slika. Jedna od primjena tih algoritama jest i određivanje apsolutne orijentacije robota i ispravljanje grešaka u mjerenju prijednog puta. Pokazani eksperimentalni rezultati upućuju na dramatično poboljšanje preciznosti lokalizacije kada se orijentacija robota određuje računarskim vidom.

Schuster et al. [37] također analiziraju upotrebu nedogleda za određivanje apsolutne orijentacije robota. Oni međutim pretpostavljaju da se strop okoliša sastoji od niza pravocrtnih ploča što je česta pojava u modernim zgradama sa spuštenim stropovima. Predloženi algoritam pronalazi nedogled pravaca na stropu i preko njega određuje apsolutnu orijentaciju robota u okolišu. Robot je opremljen jednom kamerom s mogućnostima podešavanja kuta zakreta i nagiba. Kut nagiba kamere se postavlja na vrijednost od $\sim 45^\circ$ kako bi pravci na stropu okoliša bile u vidnom polju kamere te kako bi proračun nedogleda tih pravaca bio numerički stabilan. Opisani postupak je jednostavan i pouzdan, međutim može se primijeniti samo na ograničenu klasu okoliša.

Kosaka i Pan [44] opisuju dva pristupa oblikovanju agenta za planiranje puta koji koriste geometrijski odnosno topološki prikaz okoliša. I u jednom i u drugom pristupu, glavna metoda lokalizacije je mjerenje prijednog puta, a računarski vid se koristi za periodičko smanjivanje mjerne nesigurnosti. Razmatraju se isključivo zatvoreni prostori (eksperimenti su provedeni u sustavu hodnika), pa se slike okoliša u fazi pretprocesiranja opisuju pravocrtnim segmentima. U prvom pristupu, robotu se zadaje trodimenzionalni geometrijski model okoliša, približni početni položaj te konačna lokacija do koje bi trebao samostalno doći. Robot stalno prati svoj položaj mjerenjem prijednog puta međutim nesigurnost tako određenog položaja raste proporcionalno s prijednom udaljenosti. Kada nesigurnost pređe zadani prag, skup pravocrtnih segmenata dobiven računarskim vidom se podudara s modelom okoliša iz memorije i dobiva se položaj čija je nesigurnost zanemariva. Opisanom metodom postiže se lokalizacija u vremenu od 30 s, što dozvoljava brzinu navigacije od $\sim 10 \frac{cm}{s}$. U drugom pristupu, zadaje se kvalitativni topološki opis okoliša, približni položaj i cilj navigacije. Iz takvog zadatka, modul za određivanje strategije putovanja određuje niz navigacijskih naredbi (npr. *idi ravno do T križanja, skreni desno, idi ravno do trećih vrata s lijeva*) koje prosljeđuje modulu za planiranje kretanja. Modul za planiranje kretanja izvršava dobivenu naredbu upravljanjem pogona robota na temelju lokalizacijskih podataka. Eventualne prepreke na putu se pri tome izbjegavaju očitavanjem ultrazvučnih određivača udaljenosti. Lokalizacija se obavlja specijaliziranim neuronskim mrežama koje na temelju pravocrtnog prikaza slike izlučuju relevantne značajke kao što su pravac pružanja hodnika, udaljenost od križanja, okvir vrata. Autori su potvrdili zadovoljavajuće eksperimentalne rezultate, međutim nisu naveli performansu sustava.

Talluri i Aggarwal [59] te Simsarian, Olson i Nandhakumar [58] su nezavisno raz-

matrali problem podudarnosti između pohranjenog pravocrtnog modela okoliša (radi se dakle o agentu za planiranje puta) i niza pravocrtnih segmenata koji su izlučeni iz dobivene slike, pod pretpostavkom da približni položaj robota u okolišu nije poznat. Sasvim općenito, takav problem podudaranja se može riješiti pretraživanjem interpretacijskih stabala za svaku početnu pretpostavljenu podudarnost između odabranog segmenta slike i proizvoljnog segmenta modela. Složenost takvog pretraživanja je vrlo velika i eksponencijalna pa se predlaže postupak pretprocesiranja modela okoliša kojom se izvodi particija modela na tzv. regije s invarijantnim pogledom [58]. Te regije karakterizira da je isti skup segmenata modela vidljiv iz svake njihove točke. Problem podudaranja se formulira kao problem pretraživanja u dva koraka. U prvom koraku se na temelju broja i ukupne duljine segmenata slike [58] odnosno Houghovom transformacijom [59] određuje relativno mali broj regija u kojima se robot može nalaziti. U drugom koraku se podudaranje obavlja izradom interpretacijskog stabla, ali pod pretpostavkom da se robot nalazi u jednoj od prethodno određenih regija što bitno skraćuje složenost pretraživanja. Opisani pristup je posebno prikladan za lokalizaciju pokretnog robota u urbanom okolišu koji se sastoji od poliedarskih zgrada.

Poglavlje 3

Modeli, metode i teoretske pretpostavke

U ovom poglavlju se detaljno opisuju modeli, metode i teoretske pretpostavke, potrebne za ostvarivanje pristupa samostalnoj lokalizaciji koji je opisan u poglavlju 1. U odjeljku 3.1, opisuje se perspektivni model kamere kojim se modelira postupak pribavljanja slike kod scena sa nezanemarivom dubinom. Modeli scena koji se mogu primijeniti u razmatranom okolišu su opisani u odjeljku 3.2. Odjeljak 3.3 sadrži matematički formalizam koji opisuje prirodu nedogleda te svojstva i operacije nad točkama i pravcima kao elementima projekcijske ravnine. Konačno, potreba za aktivnim pristupom percepciji je naglašena u odjeljku 3.4.

3.1 Model kamere

Lokalizacija računarskim vidom je usko povezana sa određivanjem kvantitativnih podataka o okolišu na temelju slika dobivenih kamerom. Bitno je stoga definirati precizne kvantitativne modele tih uređaja.

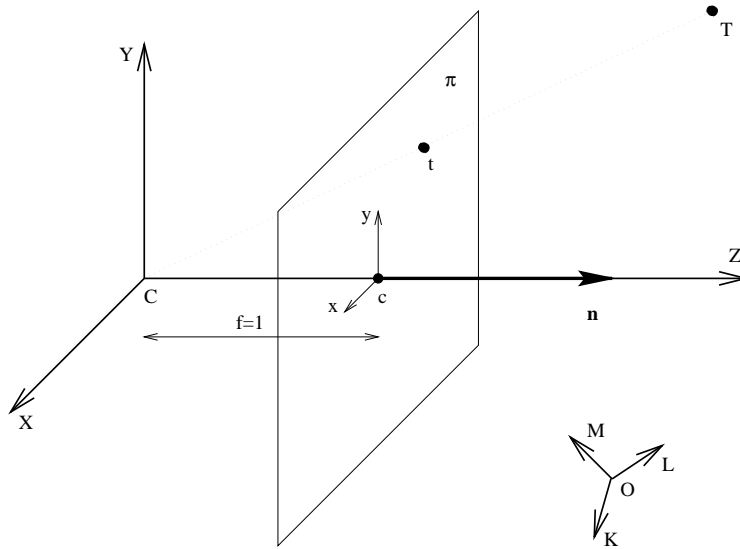
Kod novijih kamera, radijalna distorzija leće se može zanemariti u velikom broju primjena. U tim slučajevima, realna kamera se modelira idealnom kamerom (*engl.* pinhole camera) čiji je optički sustav izveden kao infinitezimalni otvor [54]. Postupak dobivanja slike idealnom kamerom se opisuje perspektivnom projekcijom trodimenzionalnog prostora na slikovnu ravninu sa slijedećim parametrima (sl. 3.1):

- C — točka koja definira središte projekcije, a odgovara žarištu optičkog sustava kamere.
- \mathbf{n} — jedinični vektor normale ravnine projekcije π koji definira os projekcije, a odgovara orijentaciji kamere, odnosno smjeru gledanja.
- f — udaljenost ravnine projekcije π od središta projekcije koja je jednaka žarišnoj daljini kamere; prikladno je za jediničnu vrijednost u koordinatnim sustavima slike i kamere odabrati $f = 1$ pa su proračuni jednostavniji.

Točka c na sl. 3.1 je određena ortogonalnom projekcijom točke C na ravninu π , pa vrijedi $\overrightarrow{Cc} = \mathbf{n}$. Ta točka se naziva *glavnom točkom* projekcije jer definira ishodište koordinatnog sustava ravnine projekcije π .

Perspektivna projekcija prostora na ravninu

Perspektivna projekcija iz 3D prostora u 2D ravninu je bila poznata davno prije prvog fotografskog aparata. Korištena je još u doba renesanse kao tehnika za postizanje realističnih efekata u slikama. Njena svojstva se mogu sažeti na slijedeći način [89]:



Slika 3.1: Jednostavni model kamere s koordinatnim sustavima svijeta (O, K, L, M) , kamere (C, X, Y, Z) i slike (c, x, y) . Točka T trodimenzionalnog prostora se preslikava u točku t ravnine projekcije.

1. Pravci iz 3D prostora (3D pravci) se preslikavaju u pravce iz ravnine projekcije (2D pravce).
2. Skup paralelnih 3D pravaca čiji smjer nije paralelan sa ravninom projekcije se preslikava u skup 2D polupravaca čija krajnja točka se naziva nedogledom, a određena je smjerom 3D pravaca.
3. Skup paralelnih 3D pravaca čiji je smjer paralelan sa ravninom projekcije se preslikava u skup paralelnih 2D pravaca s nepromijenjenim smjerom.

“Jedina” razlika između perspektivne i ortogonalne projekcije je sadržana u drugom svojstvu; *perspektivna izobličenja* se zbog toga definiraju kao deformacije perspektivne slike u odnosu na ortogonalnu sliku 2D objekta. Više perspektivnih izobličenja se može očekivati u sceni u kojoj ima više pravaca koji su paralelni s optičkom osi kamere jer će u tom slučaju položaj nedogleda pravaca biti bliže središtu slike. Suprotno, ako je glavna os (ili osi) strukture scene okomita na optičku os kamere, postupak oblikovanja slike se može aproksimirati jednostavnijom ortogonalnom projekcijom. Širina vidnog polja kamere također utječe na perspektivna izobličenja slike: šire vidno polje kamere znači da je kut između ulaznih zraka svjetlosti veći, odnosno da se dobivena slika više razlikuje od ortogonalne projekcije.

Neka je zadana točka prostora $T(X, Y, Z)$ u koordinatnom sustavu kamere, a njena slika u ravnini projekcije neka je $t(x, y)$. Izravnom primjenom Talesovog poučka, pokazuje se da uz $f = 1$ vrijedi (sl. 3.1):

$$\begin{aligned} x &= \frac{X}{Z} \\ y &= \frac{Y}{Z} \end{aligned} \tag{3.1}$$

Jednadžbe (3.1) su nelinearne pa je daljnji rad s njima nespretan. Zato je zgodno točke ravnine projekcije (x, y) opisati tzv. *homogenim koordinatama* (x_H, y_H, t_H) tako da vrijedi:

$$x = \frac{x_H}{t_H}, y = \frac{y_H}{t_H}, t_H \neq 0. \tag{3.2}$$

Svaka točka ravnine projekcije ima beskonačno mnogo prikaza u homogenim koordinatama što je na prvi pogled zbunjujuće. Međutim, takav opis omogućava zapis perspektivne projekcije linearnim jednadžbama. Pokazuje se da je odnos između homogenih koordinata točaka ravnine projekcije i koordinata njihovih originala u prostoru trivijalan [22, 33, 54]:

$$\begin{bmatrix} x_H \\ y_H \\ t_H \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (3.3)$$

Promjena koordinatnog sustava u ravnini projekcije

Ishodište koordinatnog sustava kod realnih slika često nije u glavnoj točki projekcije nego u gornjem lijevom kutu slike. Isto tako, na osima x i y često se ne koristi mjerna jedinica f (žarišna daljina kamere) nego širina odnosno visina slikovnog elementa. Zato u općem slučaju, koordinate dobivene jednadžbom (3.3) treba podvrgnuti daljnjoj transformaciji. Pokazuje se da je ta transformacija nad točkama slike u homogenim koordinatama linearna i da se može opisati gornjom trokutastom matricom \mathbf{K} slijedećeg oblika [54]:

$$\mathbf{K} = \begin{bmatrix} s_x & s_\theta & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

U gornjoj jednadžbi, s_x i s_y predstavljaju uvećanja po x odnosno y osi slikovne ravnine, s_θ opisuje međusobnu nakošenost (odstupanje od ortogonalnosti) osi (obično, $s_\theta \approx 0$), dok su (t_x, t_y) koordinate glavne točke projekcije u slici. Matrica \mathbf{K} ne ovisi ni o položaju ni o orijentaciji kamere (smjeru gledanja) pa kažemo da sadrži unutarnje (intrinzične) parametre kamere.

Promjena koordinatnog sustava svijeta

Koordinatni sustav svijeta (sl. 3.1) se obično ne poklapa sa koordinatnim sustavom kamere, pa je, prije uvrštenja u jednadžbu (3.3), koordinate točke prostora potrebno transformirati u koordinatni sustav kamere. Tražena transformacija ima tri rotacijska i tri translacijska stupnja slobode koji se skupnim imenom nazivaju vanjskim (ekstrinzičnim) parametrizacijama kamere. Kao i prije, prikladno je točke prostora (k, l, m) opisati homogenim koordinatama (k_H, l_H, m_H, s_H) , gdje je

$$k = \frac{k_H}{s_H}, l = \frac{l_H}{s_H}, m = \frac{m_H}{s_H}, s_H \neq 0. \quad (3.5)$$

Tada transformacija iz koordinatnog sustava svijeta u koordinatni sustav kamere postaje linearna i može se opisati matricom \mathbf{M} dimenzija 4×4 [33]:

$$\mathbf{M} = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{t} \\ \mathbf{0}_3^T & 1 \end{bmatrix}. \quad (3.6)$$

Matrica \mathbf{R} u gornjoj jednadžbi je dimenzija 3×3 , ima tri stupnja slobode i definira rotaciju koordinatnog sustava kamere u odnosu na koordinatni sustav svijeta. Translacija ishodišta koordinatnog sustava kamere u koordinatnom sustavu svijeta, opisana je trodimenzionalnim vektorom \mathbf{t} .

Opći model idealne kamere

Konačno, kombinacijom jednadžbi (3.4) i (3.6), dobiva se opći model idealne kamere koji preslikava homogene koordinate točaka koordinatnog sustava svijeta u homogene koordi-

nate točaka projekcijske ravnine:

$$\begin{bmatrix} x_H \\ y_H \\ t_H \end{bmatrix} = \mathbf{P}_{3 \times 4} \cdot \begin{bmatrix} k \\ l \\ m \\ 1 \end{bmatrix}, \quad (3.7)$$

gdje je

$$\mathbf{P}_{3 \times 4} = \mathbf{K} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \mathbf{M}. \quad (3.8)$$

Sasvim općenito, transformacija (3.7) ima 11 stupnjeva slobode. Općeniti postupak određivanja unutarnjih i vanjskih parametara kamere (kalibracija kamere [9]) dakle treba imati podatke o pravim položajima u koordinatnom sustavu svijeta za barem 6 točaka slike (6 točaka sa dvije koordinate daje dvanaest jednadžbi). Kod primjena u robotici unutarnji parametri kamere su obično konstantni i poznati pa je za određivanje transformacije dovoljno obaviti kalibraciju vanjskih parametara kamere [31]. Problem postaje još jednostavniji na području samostalne navigacije gdje je odnos između koordinatnog sustava robota i svijeta ograničen na tri stupnja slobode: dvije translacije i rotaciju u ravnini gibanja robota. Za lokalizaciju su tada relevantni položaji objekata na osi x koordinatnog sustava slike jer je os y okomita na ravninu gibanja robota. Tako je za određivanje položaja robota potrebno imati podatke o x koordinatama triju objekata čiji je apsolutni položaj poznat. Budući da se iz slikovne x koordinate objekta može odrediti njegov azimut, dobiveni postupak je ekvivalentan trijangulaciji koja je opisana u prethodnom poglavlju.

3.2 Model scene

Osnovno svojstvo sustava za tumačenje scena računarskim vidom je način modeliranja percipiranih objekata. Čovjekova percepcija također modelira objekte iz stvarnog svijeta, a izgrađene modele pohranjuje za kasniju upotrebu. Iskustva kao što su snovi, hipnoza, halucinacije i optičke iluzije, govore nam kako ti modeli “žive” u svijesti čovjeka i bez izravne veze sa vizualnim podražajem. Svojedobno su veliki naponi ulagani u razvijanje općenitih sustava za raspoznavanje 3D objekata (npr. ACRONYM, IMAGINE [6]). U tim sustavima, formalizam modeliranja je zamišljen tako da se njim može opisati proizvoljni kruti objekt do zadane razine detalja. Proizvedeni apstraktni oblikovni opis je kasnije korišten za raspoznavanje objekata u slici podudaranjem modela sa izlučenim slikovnim značajkama. Nedostaci spomenutih sustava su velika složenost i slaba performansa kada izgled objekta (npr. hodnika) jako ovisi o točki gledanja [29]. Takav pristup se u novije vrijeme koristi samo u slučajevima kada su razmatrane scene sasvim općenite.

Scene zatvorenog prostora se često sastoje od objekata čije granice su plohe omeđene ravnim bridovima. Klasa takvih scena se u literaturi naziva *origami* scenama [55]. U praksi, ograničenja koja se uvode u origami scenama se često mogu dodatno pojačati. Tako se mogu analizirati *triedarske* scene [2] koje su karakterizirane daljnjim ograničenjem, da se u svakom vrhu u sceni spajaju točno tri plohe. Konačno, ako svi vektori normala ploha scene imaju jedan od tri međusobno okomita smjera, govori se o *ortogonalnim triedarskim* scenama ili o scenama iz *lego* svijeta [30]. Opisani modeli scena zatvorenog prostora su sažeti u tablici 3.1.

3.3 Projekcijska geometrija

Geometrija koja je čovjeku intuitivno najjasnija je poznata pod nazivom Euklidska geometrija. U toj, intuitivnoj geometriji, stranice likova imaju duljine, dva pravca koji se

Tablica 3.1: Neki modeli scena zatvorenog prostora

model	ograničenja granica objekata u sceni
origami scene	plohe su omeđene ravnim bridovima
triedarske scene	u svakom vrhu, sastaju se točno tri plohe*
ortogonalne triedarske scene	normale ploha imaju jedan od tri međusobno okomita smjera *

*ograničenja su kumulativna: modeli nasljeđuju ograničenja prethodnika.

sijeku određuju kut, dok se paralelni pravci ravnine nigdje ne sastaju. Spomenuta svojstva su bitna i vrijedna pažnje, jer su invarijantna na Euklidske transformacije — translaciju i rotaciju. To je također intuitivno jasno: ako neki predmet premjestimo s mjesta A na mjesto B (tj. podvrgnemo ga Euklidskoj transformaciji), očekivati ćemo da mu se geometrijska svojstva neće promijeniti, jer se radi o istom predmetu na dva različita mjesta.

Međutim, stvari se kompliciraju kad želimo takvo rezoniranje primijeniti na fotografske slike. Ako usporedimo dvije fotografije istog objekta koje su snimljene iz različitih položaja prostora, Euklidska geometrija nam može reći da to nisu isti objekti. Primjerice, usporedimo dvije slike željezničke pruge koje su dobivene za dva različita položaja kamere, kako slijedi:

1. kamera je postavljena 10 metara iznad pruge (npr. na nadvožnjaku) tako da joj je optička os usmjerena okomito prema dolje;
2. kamera je postavljena 1 metar iznad pruge tako da joj je optička os usmjerena paralelno sa smjerom pružanja pruge.

U slici 1, čelične tračnice će biti paralelne, a duljina svih drvenih pragova će biti konstantna. Ta zapažanja su u skladu s našim očekivanjima, jer takvi odnosi vrijede i na stvarnoj pruzi. Međutim, u slici 2, tračnice nisu paralelne, nego se spajaju u naizgled beskonačno udaljenoj točki. U skladu s tim, duljina drvenih pragova nije konstantna kao u slici 1, nego se tim više smanjuje što su pragovi dalji od kamere. Čovjek koji je napravio fotografije zna da pruge na slikama 1. i 2. odgovaraju istom fizičkom objektu, međutim primjena Euklidske geometrije na dobivene slike “govori” da to nije tako.

Rješenje problema je u uvođenju nove geometrije, koja je invarijantna na perspektivnu projekciju. Ta geometrija se naziva *projekcijskom geometrijom* [33, 54], a proučava svojstva i, posebno, invarijantna svojstva geometrijskih likova koji su podvrgnuti *projekcijskim transformacijama*. Temeljni pojam projekcijske geometrije je *projekcijski prostor*. Pojednostavnjeno rečeno, projekcijski prostor dimenzije N se dobiva kao perspektivna slika $N+1$ -dimenzionalnog Euklidskog prostora. Tako je projekcijska ravnina dvodimenzionalni projekcijski prostor. Projekcijska transformacija ili kraće *projekcija*, je svako preslikavanje između dva projekcijska prostora, koje pravce domene preslikava u pravce kodomene. Jedno takvo preslikavanje je i perspektivna projekcija. Obrat međutim ne vrijedi: svaka projekcijska transformacija nije perspektivna projekcija. Pokazuje se da su projekcije linearna preslikavanja, te da skup svih projekcija nad prostorima istih dimenzija čini grupu s obzirom na operaciju kompozicije. Svojstvo zatvorenosti projekcije se može iskoristiti u raznim prijenama računarske grafike. Primjenom odgovarajućih projekcija, moguće je samo na temelju jedne slike nekog geometrijskog lika, generirati slike koje bi se dobile snimanjem tog istog lika iz drugih, proizvoljno zadanih točaka prostora. Pri tome, naravno, valja izuzeti degenerativne slučajeve kod kojih je originalna slika dobivena iz točke gledanja koja pripada ravnini lika.

3.3.1 Projekcijski prostor

Točka N -dimenzionalnog projekcijskog prostora \mathcal{P}^n , može se predstaviti $N+1$ -dimenzionalnim vektorom $\mathbf{x} = [x_1, x_2, \dots, x_{n+1}]^\top$, u kojem je barem jedan x_i različit od nule [33]. Brojevi x_i , $i = 1, 2, \dots, n+1$ se nazivaju homogenim koordinatama točke prostora \mathcal{P}^n . Svakoj točki prostora \mathcal{P}^n , time je pridružen beskonačan broj koordinatnih vektora. Vektori $[x_1, x_2, \dots, x_{n+1}]^\top$ i $[y_1, y_2, \dots, y_{n+1}]^\top$ predstavljaju istu točku ako i samo ako postoji skalar $\lambda \neq 0$ takav da vrijedi $x_i = \lambda \cdot y_i$, za $i = 1, 2, \dots, n+1$.

Točke Euklidskog prostora \mathcal{R}^n se mogu izomorfno preslikati u \mathcal{P}^n standardnom injekcijom [54]

$$[x_1, x_2, \dots, x_n]^\top \rightarrow [x_1, x_2, \dots, x_n, 1]^\top. \quad (3.9)$$

Obratno, točke projekcijskog prostora \mathcal{P}^n za koje je $y_{n+1} \neq 0$, mogu se preslikati u \mathcal{R}^n po jednadžbi

$$[y_1, y_2, \dots, y_{n+1}]^\top \rightarrow \left[\frac{y_1}{y_{n+1}}, \frac{y_2}{y_{n+1}}, \dots, \frac{y_n}{y_{n+1}} \right]^\top. \quad (3.10)$$

Točke $[y_1, y_2, \dots, y_n, 0]^\top \in \mathcal{P}^n$ se dobivaju kao “presjecišta” paralelnih pravaca koji u \mathcal{R}^n imaju vektor smjera $[y_1, y_2, \dots, y_n]^\top$. One nemaju prikaz u \mathcal{R}^n i nazivaju se *točkama u beskonačnosti (idealnim točkama)*. Skup svih idealnih točaka u \mathcal{P}^n ima jednadžbu hiper-ravnine $y_{n+1} = 0$ i naziva se *hiper-ravninom u beskonačnosti (idealnom hiper-ravninom)*.

3.3.2 Točke i pravci u projekcijskoj ravнини

Za računarski vid, najzanimljiviji su projekcijski prostori \mathcal{P}^2 i \mathcal{P}^3 . Dvodimenzionalni projekcijski prostor \mathcal{P}^2 ili projekcijska ravnina se može dobiti perspektivnom projekcijom prostora \mathcal{R}^3 (odjeljak 3.1). Tom se transformacijom u svaku točku projekcijske ravnine preslikava pravac prostora \mathcal{R}^3 koji se naziva interpretacijskim pravcem te točke.

Svakoj točki $[x_H, y_H, t_H]^\top \in \mathcal{P}^2$ za koju vrijedi $t_H \neq 0$ odgovara točka Euklidske ravnine¹ $\left[\frac{x_H}{t_H}, \frac{y_H}{t_H} \right]^\top$. Točke $[x_H, y_H, 0]^\top \in \mathcal{P}^2$ nemaju prikaz u Euklidskoj ravнини i nazivaju se idealnim točkama. Te točke se dobivaju kao presjecišta svih paralelnih pravaca projekcijske ravnine, koji u Euklidskoj ravнини imaju zajednički koeficijent smjera $k = \frac{y_H}{x_H}$. Skup svih idealnih točaka čini *pravac u beskonačnosti (idealni pravac, horizont ravnine)*. Može se stoga reći da se projekcijska ravnina dobiva kao unija točaka Euklidske ravnine i pravca u beskonačnosti.

Kao i točke, i pravci projekcijske ravnine se mogu predstaviti trodimenzionalnim vektorima. Točke $[x_H, y_H, t_H]^\top \in \mathcal{P}^2$ koje leže na pravcu $[a, b, c]^\top$ su određene jednadžbom

$$a \cdot x_H + b \cdot y_H + c \cdot t_H = 0. \quad (3.11)$$

Lako se provjerava da, kod perspektivne projekcije $\mathcal{R}^3 \rightarrow \mathcal{P}^2$, točke prostora \mathcal{R}^3 čije slike zadovoljavaju (3.11) čine ravninu kroz ishodište. Ta ravnina se naziva interpretacijskom ravninom pravca. Alternativno, može se reći da se interpretacijska ravnina pravca projekcijske ravnine dobiva kao skup svih pravaca prostora \mathcal{R}^3 koji se preslikavaju u zadani pravac. Paralelni pravci projekcijske ravnine se sijeku u idealnoj točki koja je tim pravcima pridružena. Ta točka je punopravan element projekcijske ravnine pa, analogno pravilu da svake dvije točke definiraju pravac, u projekcijskoj ravнини vrijedi i da svaka dva pravca definiraju točku. To svojstvo upućuje na fundamentalni princip dualnosti pravaca i točaka u projekcijskoj ravнини. Taj princip kaže da se svi teoremi projekcijske geometrije javljaju u dualnim parovima za koje vrijedi da se, počevši od bilo kojeg od njih, drugi teorem odmah može izvesti zamjenom riječi “točka” i “pravac” [89].

Lako se pokazuje da točka \mathbf{t} leži na pravcu \mathbf{p} ako i samo ako vrijedi²

$$(\mathbf{t}, \mathbf{p}) = 0, \quad (3.12)$$

¹U računarskom vidu, ta ravnina što odgovara slikovnoj ravнини.

² (\mathbf{a}, \mathbf{b}) označava skalarni produkt vektora \mathbf{a} i \mathbf{b} .

tj. ako interpretacijski pravac točke \mathbf{t} leži u interpretacijskoj ravnini pravca \mathbf{p} .

Pravac \mathbf{p} prolazi kroz točke \mathbf{t}_A i \mathbf{t}_B , ako vrijedi³

$$\mathbf{p} = \mathbf{t}_A \times \mathbf{t}_B, \quad (3.13)$$

tj. ako je normala interpretacijske ravnine pravca okomita na interpretacijske pravce točaka. Izravnom primjenom principa dualnosti, iz jednadžbe (3.13) slijedi da se točka \mathbf{t} nalazi na presjecištu pravaca \mathbf{p}_1 i \mathbf{p}_2 ako vrijedi

$$\mathbf{t} = \mathbf{p}_1 \times \mathbf{p}_2. \quad (3.14)$$

3.3.3 Nedogled kao slika idealne točke prostora \mathcal{P}^3

U perspektivnim slikama, pravci koji odgovaraju paralelnim bridovima scene se često sijeku u konačnoj točki slike koja se naziva nedogledom (*engl.* vanishing point). Svakom skupu paralelnih bridova scene je pri tome pridružen različiti nedogled. Tu pojavu elegantno opisuje projekcijska geometrija pa je to još jedan od razloga (pored onih koji su navedeni u odjeljku 3.1) zašto je i svijet prikladno opisati projekcijskim prostorom.

Prema (3.9), točke $[x, y, z]^\top$ uobičajenog Euklidskog prostora \mathcal{R}^3 odgovaraju točkama $[x, y, z, 1]^\top$ projekcijskog prostora \mathcal{P}^3 . Slično kao i u projekcijskoj ravnini, idealne točke prostora \mathcal{P}^3 imaju oblik $[x_H, y_H, z_H, 0]^\top$, a dobivaju se kao presjecišta paralelnih pravaca pa nemaju prikaz u \mathcal{R}^3 . Skup svih idealnih točaka prostora \mathcal{P}^3 čini *ravninu u beskonačnosti* (*idealnu ravninu*).

Posebno je zanimljivo pogledati u koje točke projekcijske ravnine se preslikavaju idealne točke prostora \mathcal{P}^3 prilikom formiranja slike u idealnoj kameri. Sasvim općenito, jednadžba projekcijskog preslikavanja $\mathcal{P}^3 \rightarrow \mathcal{P}^2$ glasi (3.7):

$$\begin{bmatrix} x_H \\ y_H \\ t_H \end{bmatrix} = P_{3 \times 4} \cdot \begin{bmatrix} k_H \\ l_H \\ m_H \\ s_H \end{bmatrix}. \quad (3.15)$$

Analizom izraza (3.4), (3.6) i (3.8) iz odjeljka 3.1, može se pokazati da je rang matrice $P_{3 \times 4}$ jednak tri. Idealne točke su punopravne točke projekcijskog prostora pa se one preslikavaju po jednadžbi

$$\begin{bmatrix} x_H \\ y_H \\ t_H \end{bmatrix} = [P'_{3 \times 3} \quad P''_{3 \times 1}] \cdot \begin{bmatrix} k_{Hid} \\ l_{Hid} \\ m_{Hid} \\ 0 \end{bmatrix} = P'_{3 \times 3} \cdot \begin{bmatrix} k_{Hid} \\ l_{Hid} \\ m_{Hid} \end{bmatrix}. \quad (3.16)$$

Matrica $P'_{3 \times 3}$ također ima rang 3 pa definira bijektivno preslikavanje. To znači da se pri perspektivnoj projekciji $\mathcal{P}^3 \rightarrow \mathcal{P}^2$, u svaku točku projekcijske ravnine \mathcal{P}^2 preslikava točno jedna idealna točka prostora \mathcal{P}^3 . Perspektivna projekcija tako preslikava neke idealne točke iz \mathcal{P}^3 u točke projekcijske ravnine koje nisu idealne pa imaju konačne koordinate u slikovnoj ravnini. Može se pokazati da se u idealne točke projekcijske ravnine preslikavaju samo one idealne točke projekcijskog prostora, koje odgovaraju pravcima koji su okomiti na normalu slikovne ravnine.

Time je nastanak nedogleda objašnjen. Paralelni pravci prostora se sijeku u idealnoj točki, čija slika određuje presjecište slika pravaca u \mathcal{P}^2 . Ako ta slika nije i sama idealna točka, onda ima konačne koordinate u slikovnoj ravnini i naziva se nedogledom. Pokazuje se da, pri perspektivnoj projekciji, svaka točka slikovne ravnine odgovara jednoj i samo jednoj idealnoj točki projekcijskog prostora \mathcal{P}^3 . To znači da se na temelju položaja nedogleda pravaca slikovne ravnine može odrediti smjer njihovih originala u projekcijskom prostoru.

³ $\mathbf{a} \times \mathbf{b}$ označava vektorski produkt vektora \mathbf{a} i \mathbf{b} .

3.4 Aktivni vid

Do sredine 80-ih godina, predlagani su uglavnom pasivni pristupi tumačenju scene računarskim vidom. Postupci podešavanja parametara kamere se kod takvih pristupa obavljaju odvojeno od postupka analize pribavljenih slika, ručnim i najčešće empirijskim metodama. U posljednjem desetljeću, međutim, sve se više prihvaća ideja da je percepcija⁴ i na razini prikupljanja podataka aktivan proces. U tom procesu, u ovisnosti o raspoloživom apriornom ili stečenom znanju o svijetu kao domeni osjetilne spoznaje, svijet se istražuje, pretražuje, ispituju se kozistentnosti skupova pretpostavki i sl. U okviru takvog poimanja percepcije, može se očekivati da će klasični pristup sa pasivnom kamerom, zbog ograničenih procesnih i prostornih računarskih resursa, moći obavljati samo uski podskup svih perceptivnih aktivnosti koje se temelje na vidnom podražaju.

3.4.1 Motivacija

Velik broj problema na području umjetne inteligencije se svodi na problem pretraživanja grafa [40]. Čvorovi grafa mogu predstavljati skupove činjenica, položaje objekata u sobi ili nepotpuno raščlanjene rečenice prirodnog govora. Lukovi grafa tada predstavljaju korake u zaključivanju, pokrete robota odnosno primjene pravila raščlanjivanja [40]. Međutim, kod netrivialnih problema, veličina grafa je redovno ograničavajući faktor — često je broj čvorova praktično beskonačan iako je broj lukova traženog puta relativno malen. Tako na primjer, rijetke partije šaha se ne završe do dvjestotog poteza dok je broj različitih šahovskih pozicija nepojmljivo velik. *Slijepa* metode pretraživanja [46] (npr. pretraživanje u širinu ili pretraživanje u dubinu) koje jamče pronalaženje optimalnog rješenja (npr. najkraći put do završnog čvora grafa), u praksi su zato često neupotrebljive jer im se resursi (npr. procesorsko vrijeme, memorijski prostor, mrežna propusnost) za dobivanje rješenja ne mogu ograničiti. Zato se razvijaju heurističke metode pretraživanja koje daju suboptimalne rezultate ili nepotpuna rješenja, ali prilikom izvršavanja koriste bitno manju ili unaprijed određenu količinu resursa.

Analogni pristup se može primijeniti za opis složenih zadataka samostalnog snalaženja i kretanja u prostoru. Primjeri takvih zadataka su traženje određenog objekta u nepoznatom zatvorenom prostoru [87] ili pronalaženje puta u hodniku čiji je tlocrt prethodno zadan [44]. Neka čvorovi grafa predstavljaju skup parametara za dohvat slike (parametri kamere, parametri algoritama za obradu slike te položaj robota), a lukovi grafa neka predstavljaju odabir parametara za dohvat slijedeće slike. U završno stanje tako postavljenog problema se dolazi kad se pribavi slika analizom koje se može zaključiti da je robot riješio problem (tj. da je odredio vlastiti položaj, stigao na odredište itd.). U svakoj netrivialnoj primjeni postoji ogroman broj čvorova takvog grafa i nije ih ni praktično ni moguće sve iskušati. Zato je nužno razviti strategije traženja završnog čvora koje ne zahtijevaju posjećivanje većine ostalih čvorova. Takve strategije izučava aktivna percepcija odnosno aktivni vid, a analogne su heurističkim algoritmima pretraživanja grafova.

3.4.2 Definicija aktivnog vida

Sasvim općenito, aktivna percepcija se definira kao modeliranje i upravljanje strategijama za ostvarivanje percepcije [11]. Više specifično, aktivni vid se definira kao prilagodljivo upravljanje parametara kamere (žarišna daljina, širina vidnog polja, smjer gledanja, pozicija, otvor objektiva) u cilju pojednostavljenja obrade i poboljšanja rezultata sustava računarskog vida. Za razliku od pasivnog pristupa, u sustavu aktivnog vida ravnopravno se specificiraju postupak pribavljanja slike i algoritmi za njihovu obradu. Može se reći i da

⁴percepcija — proces stvaranja predodbi u svijesti na temelju osjeta, dojmova; V. Anić Rječnik hrvatskog jezika, 1991.

je aktivni vid primjena teorije inteligentnog upravljanja na postupak pribavljanja slike, u skladu s trenutnim zadatkom ili konačnim ciljem obrade složenog sustava [11].

U ovom kontekstu, aktivni vid ne označava korištenje aktivnih senzora (npr. radar, sonar, laserski određivač udaljenosti), nego percepciju pasivnim sensorima na aktivan način, svrhovito prilagođavajući parametre senzora u cilju rješavanja zadatka, u skladu sa odabranom strategijom percepcije. Aktivni računarski vid je uvijek komponenta složenijeg računalnog sustava: oblikuje se u skladu sa njegovim zahtjevima i vrednuje u kontekstu njegovih rezultata. Aktivni vid obuhvaća slijedeće glavne ideje [71]:

1. *što vidjeti?* — pribavljanje slika se odvija u skladu sa ciljem obrade u složenom sustavu;
2. *kako gledati?* — pribavljanjem slika za analizu se upravlja prilagodljivim (“inteligentnim”) postupcima.

Na važnost aktivnog vida ukazali su i Aloimonos i Bandyopadhyay [10] tako što su pokazali da neki tipični problemi u računarskom vidu (analizirani su problemi optičkog toka, strukture iz kretanja te oblika iz sjenčenja, konture i teksture) koji su za pasivnog promatrača nelinearni, nestabilni i nemaju jedinstveno rješenje, za aktivnog promatrača postaju linearni i stabilni problemi, odnosno problemi s jedinstvenim rješenjem.

3.4.3 Svojstva sustava aktivnog vida

Aktivni vid se zasniva na pretpostavci da se dohvaćanje podataka odvija u skladu s potrebama koje određuju hijerarhijski više razine obrade u složenom računalnom sustavu kojem je potrebna mogućnost percepcije. Slijedeći koncepti se zato mogu naći kod većine sustava aktivnog vida [11, 83]:

- modeli postupaka za dobivanje slike i algoritama obrade (tzv. lokalni modeli) koji se opisuju odgovarajućim skupovima parametara te njihovim domenama;
- globalni model sustava koji specificira međuovisnost lokalnih modela te stečenog i apriornog znanja;
- dobro definirano početno i ciljno stanje sustava.

Globalni model definira povratne veze među komponentama sustava aktivnog vida (dakle i sensorima) koje su opisane lokalnim modelima.

U sustavima aktivnog vida, podaci se dohvaćaju i obrađuju u skladu sa zahtjevima složenog sustava. Budući da su performanse računarskih sustava još uvijek ograničavajući faktor, to znači da sustav aktivnog vida u pravilu radi neprekidno. Aktivni vid djeluje kao filter za informaciju — iz svih podataka koji su dostupni senzoru, potrebno je izdvojiti podatke bitne za rad složenog sustava (ne može se gledati svuda u isto vrijeme). Pogodno je stoga ograničiti obradu na malo područje slike, koje se onda naziva područjem pažnje. Također je često pogodno da sustav aktivnog vida uvijek daje odgovor u predodređenom vremenskom intervalu koji ovisi o prirodi složenog sustava.

Aktivna percepcija se često vezuje sa pojmom selektivne percepcije koji označava selektivnu obradu podražaja u prostoru, rezoluciji i vremenu. Tako se slijedeći koncepti ocjenjuju kao iznimno važni u razvoju sustava aktivnog vida [43]:

- Pažnja (*engl.* attention [64, 79, 86])
Selektivna obrada pojedinih područja ulazne slike na temelju pozicije, udaljenosti ili kretanja.
- Senzori s nejednolikom gustoćom (*engl.* foveal sensing)
kamere modeliraju gustoću osjetilnih stanica u ljudskom oku koja radijalno opada od središta prema rubovima slike.

- Upravljanje smjerom gledanja (*engl.* gaze control [69])
Upravljanje orijentacijom kamere tako da dobivene slike budu prikladne za rješavanje cilja obrade. Može se podijeliti na slijedeće komponente:
 - *stabilizacija smjera gledanja* [71, 79]: upravljanje smjerom gledanja tako da objekt pažnje ostane unutar vidnog polja kamere;
 - *promjena smjera gledanja* [79]: upravljanje smjerom gledanja tako da se vidno polje kamere premjesti na novi objekt pažnje.
- Koordinacija ruke i oka (*engl.* hand-eye coordination) [88]: Integracija sustava robotske ruke sa sustavom računarskog vida.

Poglavlje 4

Određivanje težinskog pravocrtnog prikaza slike

U ovom poglavlju, opisana su dva koraka postupka kojim se izvorna slika intenziteta svjetlosti u sceni opisuje težinskom listom pravocrtnih segmenata. U kraćem obliku, taj postupak je već opisan u [85]. Postupci izdvajanja geometrijskih značajki se u literaturi obično nazivaju pretprocesiranjem jer se odvijaju prije samog tumačenja scene. Pravocrtni međuprikaz je odabran zbog pretpostavke da se scene hodnika mogu dovoljno precizno opisati jednim od modela iz tablice 3.1. Prvi korak postupka je zadužen za pronalaženje rubnih elemenata u izvornoj slici, a opisan je u odjeljku 4.1. Predloženi algoritam se zasniva na dobro poznatom Cannyjevom algoritmu koji se preporuča kao jedan od najprikladnijih za analizu scena s neteksturiranim objektima iz urbanog okruženja. Rezultat algoritma su težinska slika rubnih elemenata te slika smjerova gradijenta na kojima se temelji traženje pravocrtnih segmenata koje je, kao drugi korak postupka, opisano u odjeljku 4.2. U predloženom iterativnom postupku, težinski pravocrtni segmenti se traže na pravcima određenim analizom modificirane Houghove transformacije težinske slike rubnih elemenata, u skladu sa odgovarajućom slikom smjerova gradijenta.

4.1 Pronalaženje rubova

U ovom odjeljku, opisuje se algoritam pronalaženja rubova koji je primijenjen u fazi pretprocesiranja svih postupaka koji su opisani u ovom radu. Predloženi algoritam na ulazu zahtijeva sliku opisanu poljem intenziteta, dok na izlazu određuje težinsku sliku rubnih elemenata i sliku smjerova gradijenta.

Granice objekata u sceni se često manifestiraju kao orijentirane i lokalizirane promjene intenziteta u slici koje se nazivaju rubovima. Pronalaženje rubova se zato kod velikog broja sustava računarskog vida koristi kao prvi korak analize slike na kojem se temelji cjelokupna naknadna obrada. Izrada algoritma za pronalaženje rubova koji bi u nekom smislu bio bolji od postojećih bi stoga poboljšala svojstva većine primjena računarskog vida pa je ta tema konstantno zastupljena u znanstvenim i stručnim časopisima. Općenito se vjeruje da je postignut priličan napredak u odnosu na rane radove Sobela i Robertsa, ali postoje i ocjene da se performansa novih algoritama približava asimptotskoj razini. Većina postojećih algoritama na izlazu vraćaju binarnu sliku rubova dok neki određuju i smjer gradijenta u elementima izvorne slike.

Odjeljak je organiziran na slijedeći način. Motivacija za odabir pristupa pronalaženju rubova je navedena u pododjeljku 4.1.1. Pododjeljak 4.1.2 sadrži opis Cannyjevog algoritma za pronalaženje rubnih elemenata. Na kraju, u pododjeljcima 4.1.3 i 4.1.4, opisane su modifikacije koje su u tom algoritmu uvedene kako bi se on mogao koristiti u neinteraktivnom sustavu, odnosno detalji izvedbe konačnog postupka.

4.1.1 Usporedba postojećih algoritama

Iako je razvijen velik broj različitih postupaka za pronalaženje rubova u slikama stvarnih scena, relativno je malen broj predloženih metodologija za kvantitativnu procjenu njihovih performansi. Pri uspoređivanju performanse različitih algoritama, stoga se uvriježila praksa usporednog prikazivanja vizualnih rezultata pri čemu se zadatak subjektivne procjene prepušta čitatelju. Jedan od malobrojnih novijih prijedloga za kvantitativnu usporedbu dao je Heath u radovima [53, 80]. Predložena metodologija se temelji na statističkoj obradi subjektivnih ocjena većeg broja ljudi. Metodologija je primijenjena pri usporedbi rezultata algoritama za pronalaženje rubova slijedećih autora: Bergholm, Canny, Iverson, Nalwa-Binford, Rothwell, Sarkar-Boyer i Sobel.

Uvriježena je praksa da algoritmi za pronalaženje rubnih elemenata, pored slike koju treba obraditi, na ulazu zahtijevaju i parametre kojima se algoritam prilagođava odnosno konfigurira za zadanu sliku. Obično je broj parametara algoritama malen (oko tri), a određuju ili mjeru gladenja ulazne slike, ili vrijednosti praga koje upravljaju nižim razinama obrade u algoritmu. Precizan postupak odabira parametara ispitivanih algoritama, međutim u pravilu nije propisan i obično se provodi empirijskim metodama.

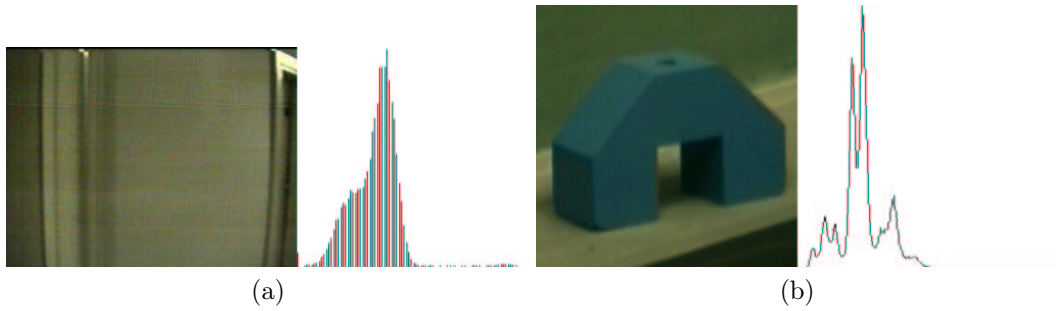
Za svaku pokusnu sliku, ispitivanja [53, 80] su provedena za niz različitih konfiguracija pojedinačnih algoritama. Kao konačni rezultat takvog ispitivanja, za svaki algoritam dobivene su dvije srednje ocjene. Prva ocjena se dobiva kao srednja vrijednost najvećih pojedinačnih ocjena za svaku pokusnu sliku, za proizvoljnu konfiguraciju algoritma. Ta ocjena vrijedi u primjenama u kojima se može osigurati prilagođavanje konfiguracije algoritma analiziranoj slici. Druga ocjena se dobiva kao srednja vrijednost pojedinačnih ocjena uz stalnu konfiguraciju algoritma koja tu ocjenu maksimizira. Ta ocjena je donja granica performanse algoritma, koja se postiže kada ne postoje nikakve indicije o tome koja bi konfiguracija mogla postići dobre rezultate za analiziranu sliku. Ispitivanje je pokazalo da algoritmi, nažalost, najbolje rezultate postižu tek kad se optimalno konfiguriraju za svaku ispitnu sliku. Za sve algoritme su dobivene značajne razlike između prve i druge ocjene, što znači da konfiguracija algoritama bitno utječe na njihovu performansu.

Slike nad kojima su provedena ispitivanja su podijeljene u četiri skupine, na temelju objekata koje prikazuju:

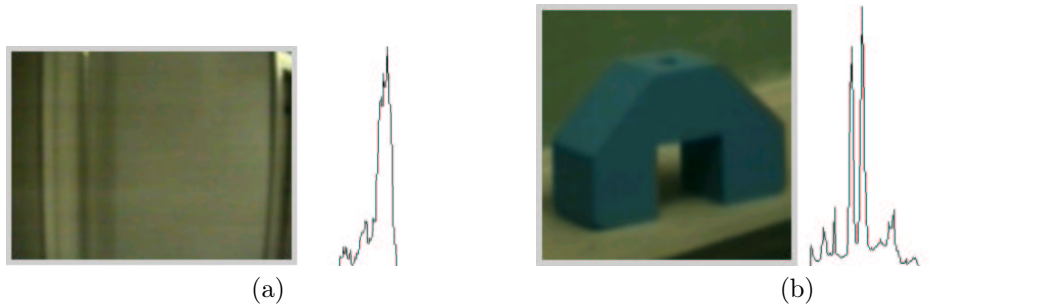
1. prirodne scene sa naglašenom teksturom;
2. prirodne scene bez tekture;
3. scene iz ljudskog okruženja sa naglašenom teksturom;
4. scene iz ljudskog okruženja bez tekture.

Za ovaj rad, najrelevantnija je performansa koju pojedini algoritmi postižu na neteksturiranim scenama iz ljudskog okruženja. Pokazalo se da za takve scene, uz prilagođenu konfiguraciju, najbolje rezultate postiže Cannyjev algoritam — 5,54 na ljestvici od 1 do 7. Prosječna ocjena ostalih algoritama u takvim uvjetima iznosi 5,07, dok je ocjena najgoreg algoritma 4,85. Međutim, rezultati Cannyjevog algoritma uz stalnu konfiguraciju su bitno lošiji — 4,79, dok su prosječna i najmanja ocjena ostalih algoritama 4,91 i 4,76 [53].

Sustav aktivnog vida pri obavljanju nekog zadatka analizira veći broj slika okoliša, dobivenih za različite parametre kamere. Brzina obrade pojedine slike u takvom sustavu je vrlo bitna jer se stvarni zadaci obično moraju rješavati u stvarnom vremenu. Iskustva pokazuju da se u gotovo svim sustavima računarskog vida više od pola vremena obrade potroši u niskim razinama obrade jer je tada dimenzija problema (koliko god on jednostavan bio) jednaka broju slikovnih elemenata slike. Drugo relevantno svojstvo algoritama za pronalaženje rubova u kontekstu ovog rada, stoga je brzina izvođenja. Analiza [53] pokazuje da je Cannyjev algoritam značajno najbrži u skupu ispitivanih algoritama (tri puta brži od najbližeg suparnika).



Slika 4.1: Slike i odgovarajući histogrami sivih razina dobiveni za dva različita para kamera i digitalizatora slike.



Slika 4.2: Slike i odgovarajući histogrami dobiveni glaćenjem slika na sl. 4.1 uz $\sigma = 1,7$.

Na temelju izloženih svojstava različitih algoritama iz literature, donesen je zaključak da bi najprimjereniji za razmatranu primjenu mogao biti Cannyjev algoritam. Uvjet za to je pronalaženje pravila za određivanje parametara algoritma na temelju svojstava ulazne slike. Slijedeći pododjeljak stoga sadrži definiciju izvornog algoritma, dok je procedura za dinamičko određivanje njegovih parametara opisana u pododjeljku 4.1.3.

4.1.2 Cannyjev algoritam za pronalaženje rubnih elemenata

Cannyjev algoritam [5, 73] je vrlo često korišten i smatra se standardnom metodom za pronalaženje rubnih elemenata [53]. Kao i kod većine ostalih algoritama obrade slike niske razine, u prvom koraku tog algoritma obavlja se tzv. “glađenje” slike kako bi se odstranili utjecaji šuma i slabog uzorkovanja. Glađenje se definira kao korigiranje početnog skupa podataka kako bi se dobio približno neprekidan skup u kojem su podaci sa velikim odstupanjem približeni srednjoj vrijednosti ili izbačeni [89]. Ako se slika promatra kao 2D signal, postupak glađenja odgovara nisko propusnom filteru jer se njime potiskuju visoke frekvencije u slici. Pokazuje se da je za tu namjenu posebno pogodna Gaussova funkcija s parametrom σ koji se postavlja u ovisnosti o kvaliteti ulazne slike te dimenzijama značajki koje se želi izlučiti

$$f(x, y) = \frac{1}{2\pi\sigma^2} e^{-[(x-\mu_x)^2 + (y-\mu_y)^2]/2\sigma^2}. \quad (4.1)$$

Sl. 4.1 i sl. 4.2 prikazuju utjecaj glađenja na digitaliziranu sliku i njen histogram, za dva različita para kamera i digitalizatora slike. Slika na sl. 4.1(a) je dobivena pomoću relativno stare VHS kamere pa je primjetan utjecaj radialne distorzije. Kod novijih kamera, taj problem je bitno manje zastupljen kao što se vidi na sl. 4.1(b). Histogram slike na sl. 4.1(a) nije neprekidan jer korišteni digitalizator ima ograničene mogućnosti uzorkovanja.

Nakon glađenja, slika se diferencira u vodoravnom i uspravnom smjeru za svaki slikovni element pa se na temelju dobivenih uspravnih i vodoravnih diferencija određuju apsolutni iznos i smjer gradijenta. Rezultat provedenog postupka su slike iznosa i smjerova gradijenta. Sl. 4.3 prikazuje sliku iznosa gradijenta dobivenu obradom slike na sl. 4.1(b).



Slika 4.3: Iznos gradijenta u slici na sl. 4.1(b).

U slijedećem koraku algoritma, slika iznosa gradijenta se podvrgava postupku stanjivanja rubova, u skladu sa slikom smjerova gradijenta. U tom postupku, na nulu se postavljaju vrijednosti onih elemenata slike iznosa gradijenta, koji nisu veći od svojih susjeda u smjeru koji je definiran odgovarajućim elementom slike smjerova gradijenta. Sl. 4.4 prikazuje masku za stanjivanje rubova — binarnu sliku čiji tamni slikovni elementi odgovaraju slikovnim elementima na sl. 4.3 koje je u postupku stanjivanja potrebno zadržati.



Slika 4.4: Binarna maska za stanjivanje rubova u slici na sl. 4.3. Svijetli slikovni elementi odgovaraju elementima slike na sl. 4.3 koje je u postupku stanjivanja potrebno odbaciti.

Konačno, istanjena slika iznosa gradijenta se podvrgava uspoređivanju s dvostrukim pragom (t_V, t_N). U rubne elemente se pri tome promoviraju elementi istanjene slike iznosa gradijenta čija vrijednost je:

1. veća od t_V , ili,
2. veća od t_N , a slikovni element je povezan sa nekim već promoviranim rubnim elementom.

Sl. 4.5 prikazuje konačnu sliku rubnih elemenata, koja se dobiva stanjivanjem rubova i uspoređivanjem s dvostrukim pragom elemenata slike na sl. 4.3.

Opisani algoritam traži od korisnika unos triju parametara. Prvi parametar je σ , standardna devijacija Gaussove funkcije, izražena u slikovnim elementima. Drugi parametar, p_V , određuje visoki prag u postupku uspoređivanja, a izražava se kao odnos t_V i ukupnog broja elemenata koji su prošli postupak stanjivanja rubova. Treći parametar algoritma, p_N , određuje niži prag postupka uspoređivanja, a zadaje se kao relativni odnos niskog i visokog praga ($p_N = \frac{t_N}{t_V}$).



Slika 4.5: Konačan rezultat pronalaženja rubova u slici na sl. 4.1(b).

4.1.3 Određivanje parametara za Cannyjev algoritam

Priroda sustava kojeg se želi izvesti podrazumijeva analizu različitih dijelova scene pa rješenje u kojem bi sustav pronalaženje rubova u svim slikama obavljao uz stalni skup parametara (konfiguraciju) nije prihvatljivo. Samostalna navigacija nije interaktivna primjena pa empirijsko određivanje parametara t_V i t_N u toku rada sustava također ne dolazi u obzir. Empirijskim metodama je ocijenjeno da algoritam nije značajno osjetljiv na odabir parametara t_N i σ , jer se za većinu slika vrlo dobri rezultati postižu za $\sigma = 1,7$ i $t_N = \frac{t_V}{2}$. S druge strane, relativno odstupanje od optimalne vrijednosti parametra t_V u iznosu od 50% često rezultira slikom rubova koja je potpuno neupotrebljiva za daljnju obradu [53]. Stoga je razvijena metoda kojom se t_V odabire automatski, na temelju statističkih svojstava izvorne slike.

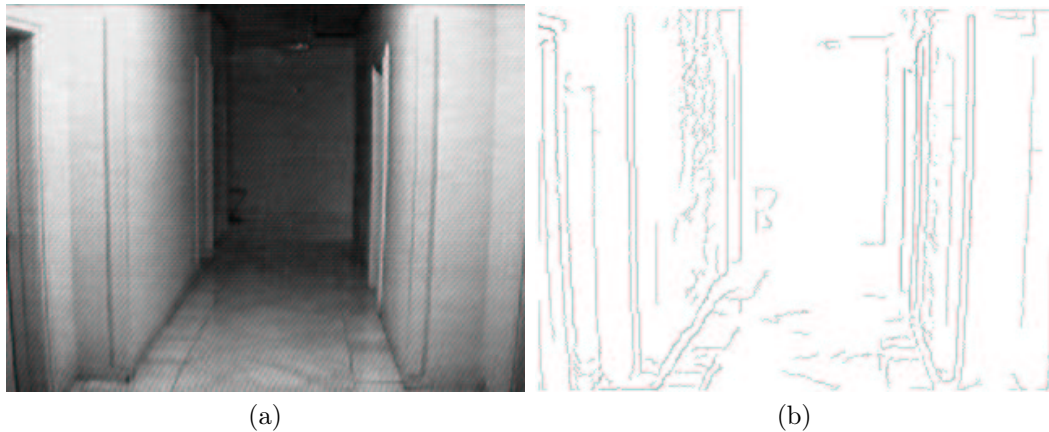
Nakon stanjivanja rubova, većina slikovnih elemenata slike iznosa gradijenta koji su različiti od nule se može svrstati u jednu od slijedeće dvije grupe:

- *rubna grupa*, koja se sastoji od malobrojnih slikovnih elemenata (tipično $< 10\%$) sa velikim vrijednostima iznosa gradijenta koji odgovaraju rubovima u početnoj slici;
- *površinska grupa*, u kojoj se nalaze brojni slikovni elementi sa malim vrijednostima iznosima gradijenta koji su rezultat šuma ili fine teksture ravnih površina.

Vrijednost glavnog parametra, t_V , treba postaviti tako da se elementi površinske grupe isfiltriraju, dok bi elementi rubne grupe trebali ostati nepromijenjeni.

Elementi površinske grupe su bitno brojniji od ostalih pa njihova svojstva približno odgovaraju svojstvima svih elemenata slike iznosa gradijenta. Parametar t_V se tako može računati na temelju srednje vrijednosti μ i standardne devijacije σ svih slikovnih elemenata po formuli $t_V = \mu + k * \sigma$ [21]. Parametar t_V je time zamijenjen novim parametrom, k , na kojeg je algoritam znatno manje osjetljiv: pokazalo se da, za dobro osvijetljene scene,

gotovo svi elementi "rubne" grupe prođu filtriranje uz $k=0,5$. Kod nejednoliko osvijetljenih scena, međutim, javlja se problem da iznos gradijenta u točkama koje odgovaraju bridovima u slabo osvijetljenim dijelovima scene, može biti manji od iznosa gradijenta elemenata površinske grupe (sl. 4.6). Taj problem sugerira da jednostavno uspoređivanje s pragom slike iznosa gradijenta nije sasvim zadovoljavajući postupak za određivanje rubnih elemenata.



Slika 4.6: Slika neravnomjerno osvijetljene scene hodnika (a) te rezultat algoritma za pronalazanje rubova (b).

Ovisno o tipu scene, u konačnoj slici rubnih elemenata se obično dobije i manji dio elemenata "površinske" grupe. Zato je donesena odluka da, protivno ustaljenoj praksi, konačna slika rubnih elemenata ne bude binarna nego siva. To je postignuto na način da je svaki rubni element slike prikazan odgovarajućim apsolutnim iznosom lokalnog gradijenta. Tako dobivena slika je nazvana *težinska* slika rubnih elemenata u kojoj svaki rubni element ima *težinu* ili "važnost" koja može biti korištena u višim razinama obrade. Sl. 4.7 sadrži alternativni prikaz rezultata obrade slika na sl. 4.5 i sl. 4.6. U tom prikazu, svaki rubni element se prikazuje dužinom čija jedna krajnja točka odgovara položaju rubnog elementa. Smjer i duljina dužine pri tome opisuju iznos odnosno smjer gradijenta u tom rubnom elementu.

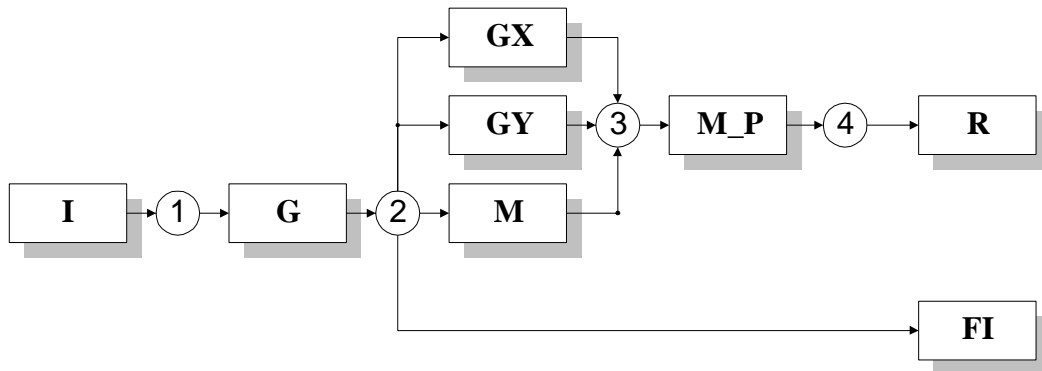


Slika 4.7: Alternativni prikaz rezultata obrade sa sl. 4.6 i sl. 4.5.

4.1.4 Izvedbeni detalji algoritma

Predloženi algoritam za pronalaženje rubova se sastoji od četiri koraka, kao što je prikazano na sl. 4.8.

1. Konvolucija izvorne slike **I** (sl. 4.1) sa dvodimenzionalnom Gaussovom funkcijom (4.1). Rezultat te operacije je izgladena slika **G** (sl. 4.2).
2. Diferenciranje slike **G** u vodoravnom i uspravnom smjeru te računanje iznosa i smjera gradijenta na temelju dobivenih diferencija. Rezultat te operacije su slike uspravnih i vodoravnih diferencija **GX** i **GY** te slike apsolutnog iznosa i smjera gradijenta **M** (sl. 4.3, 4.7) odnosno **FI**.
3. Stanjivanje rubova u slici iznosa gradijenta **M** primjenom maske **P** (sl. 4.4) koja se određuje na temelju slika **GX**, **GY** i **M**. Rezultat te operacije je slika istanjenih rubova **M_P**.
4. Potiskivanje rubova sa malom vrijednošću iznosa gradijenta u slici istanjenih rubova **M_P** upotrebom dvostrukog praga sa histerezom. Rezultat te operacije je težinska slika rubnih elemenata **R** (sl. 4.5).



Slika 4.8: Prikaz toka obrade algoritma za pronalaženje rubova. Pravokutnici označavaju ulazne i izlazne slike te međurezultate, dok krugovi označavaju korake obrade (opširnije u tekstu).

Slijedi opis izvedbenih detalja za svaki od opisanih koraka.

Glavljenje

Glavljenje je računski najintenzivniji dio predloženog algoritma za pronalaženje težinske silke rubnih elemenata. Kada Gaussova konvolucijska funkcija (4.1) ne bi bila linearno separabilna, konvolucijski postupak bi podrazumijevao obavljanje

$$N_M^{2D} = W \cdot H \cdot N^2 \quad (4.2)$$

binarnih operacija množenja, gdje su W i H dimenzije slike **I**, a N dimenzija kvadratne konvolucijske jezgre. Pri tome se N odabire po starom inženjerskom pravilu po kojem iznos Gaussove funkcije postaje zanemariv na udaljenosti 3σ od ishodišta:

$$N = 2 \cdot 3 \cdot \sigma + 1 \quad (4.3)$$

Za odabrani $\sigma = 1,7$, N iznosi 11. U eksperimentima su uglavnom korištene slike dimenzija 320×240 pa se može ocijeniti

$$N_M^{2D} \approx 5 \cdot 10^6 \quad (4.4)$$

Srećom, dvodimenzionalna Gaussova funkcija jest linearno separabilna, pa se konvolucija s njom može obaviti dvjema konvolucijama, po vodoravnoj i uspravnoj osi sa jednodimenzionalnom Gaussovom funkcijom

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}, \quad (4.5)$$

Takva izvedba konvolucije zahtijeva “samo”

$$N_M^{1D} = 2 \cdot W \cdot H \cdot N \approx 1 \cdot 10^6 \quad (4.6)$$

množenja što odgovara ubrzanju od pet puta. Eksperimentalni rezultati potvrđuju tu ocjenu: na računalu s performansom od 4 SPECint95, glaćenje slike dimenzija $320 \times 240 \times 24$ bita uz pomoć dvodimenzionalne konvolucije traje oko 3,0 s. Ekvivalentni postupak izveden dvjema jednodimenzionalnim konvolucijama zahtijeva samo oko 0,6 s.

Programska izvedba konvolucije se temelji na generičkom algoritmu [74], koji je parametriziran po tipu slikovnog elementa i dimenziji konvolucijske jezgre. Pri tome slikovni element može biti opisan sa 8 i 16 bitova za sivu sliku, odnosno 24 i 32 bita za sliku u boji. Svi ostali algoritmi pretprocesiranja su također parametrizirani sa tipom slikovnog elementa čime su istovremeno osigurane i efikasnost i prilagodljivost programskog sustava.

Diferenciranje

Za svaki slikovni element slike \mathbf{G} , računaju se odgovarajući slikovni elementi slika \mathbf{GX} , \mathbf{GY} , \mathbf{M} , \mathbf{FI} u skladu s jednadžbama:

$$\begin{aligned} g_x &= \mathbf{G}[i, j + 1] - \mathbf{G}[i, j - 1] \\ g_y &= \mathbf{G}[i + 1, j] - \mathbf{G}[i - 1, j] \\ \mathbf{GX}[i, j] &= g_x \\ \mathbf{GY}[i, j] &= g_y \\ \mathbf{M}[i, j] &= \sqrt{g_x^2 + g_y^2} \\ \mathbf{FI}[i, j] &= \arctan\left(\frac{g_y}{g_x}\right) \end{aligned} \quad (4.7)$$

Ako su slikovni elementi RGB vektori, onda je operacija oduzimanja definirana kao

$$\begin{bmatrix} r_1 \\ g_1 \\ b_1 \end{bmatrix} - \begin{bmatrix} r_2 \\ g_2 \\ b_2 \end{bmatrix} = s \cdot \sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2}, \quad (4.8)$$

gdje je

$$s = \text{sgn}(r_1 + g_1 + b_1 - r_2 - g_2 - b_2). \quad (4.9)$$

Stanjivanje rubova

Ovaj korak obrade se sastoji od dvije operacije:

(a) Određivanje maske za stanjivanje rubova:

Za svaki slikovni element slike \mathbf{M} , računa se odgovarajući slikovni element binarne slike \mathbf{P} po algoritmu:

```

gx=GX[i, j]
gy=GY[i, j]

el00 = M[i - 1, j - 1]
el01 = M[i - 1, j]
el02 = M[i - 1, j + 1]
el10 = M[i, j - 1]
el11 = M[i, j]
el12 = M[i, j + 1]
el20 = M[i + 1, j - 1]
el21 = M[i + 1, j]
el22 = M[i + 1, j + 1]

ako (gx ≥ 0 ∧ gy ≥ 0) ∨ (gx ≤ 0 ∧ gy ≤ 0)
  ako (|gx| ≥ |gy|)
    dx1 := el11 - el10
    dy1 := el10 - el00
    dx2 := el11 - el12
    dy2 := el12 - el22
  inače
    dx1 := el01 - el00
    dy1 := el11 - el01
    dx2 := el21 - el22
    dy2 := el11 - el21
inače
  ako (|gx| ≥ |gy|)
    dx1 := el11 - el10
    dy1 := el10 - el20
    dx2 := el11 - el12
    dy2 := el12 - el02
  inače
    dx1 := el21 - el20
    dy1 := el11 - el21
    dx2 := el01 - el02
    dy2 := el11 - el01

ako (dx1 · |gx| + dy1 · |gy| < 0) ∨ (dx2 · |gx| + dy2 · |gy| ≤ 0)
  P[i, j] := ⊤
inače
  P[i, j] := ⊥

```

(b) Primjena maske za stanjivanje rubova:

Za svaki slikovni element slike **M**, u skladu sa slikom **P**, računa se odgovarajući slikovni element slike **M_P** po algoritmu:

```

ako (P[i, j] == ⊤)
  M_P[i, j] := M[i, j]
inače
  M_P[i, j] = 0

```

Primjena dvostrukog praga s histerezom

Na temelju srednje vrijednosti μ i standardne vrijednosti σ elemenata slike $\mathbf{M_P}$, određuju se parametri t_V i t_N , po jednadžbama:

$$\begin{aligned}t_V &= \mu + 0,5 * \sigma \\t_N &= \frac{t_V}{2}\end{aligned}\tag{4.10}$$

Svi elementi slike \mathbf{R} se postavljaju na nulu. Elementi slike \mathbf{R} se na temelju odgovarajućih elemenata slike $\mathbf{M_P}$ određuju po slijedećem algoritmu:

```
ako R[i, j]==0
  ako M_P[i, j]>t_V
    R[i, j]:=M_P[i, j]
    za svih 8 susjednih elemenata
      obradi_susjeda(i_s, j_s)

procedura obradi_susjeda(i, j)
  ako R[i, j]==0
    ako M_P[i, j]>t_N
      R[i, j]:=M_P[i, j]
      za svih 8 susjednih elemenata
        obradi_susjeda(i_s, j_s)
```

4.2 Povezivanje rubnih elemenata u pravocrtne segmente

Pronalaženje granica u slici (rubna segmentacija) se tradicionalno izvodi u dva koraka:

- pronalaženje rubnih elemenata (*engl.* edge detection);
- povezivanje rubnih elemenata (*engl.* edge linking).

Tipična izvedba algoritma za pronalaženje rubnih elemenata (odjeljak 4.1) ne sadrži niti ne zahtijeva bilo kakvo apriorno znanje o slici koja se analizira. Postupci povezivanja rubnih elemenata, naprotiv, obično su usko prilagođeni specifičnostima analizirane slike i algoritmima na hijerarhijski višim razinama sustava računarskog vida. Naravno, bilo bi vrlo korisno oblikovati jedan sustav za obavljanje rubne segmentacije u svim vrstama slika, ali pokazuje se da se takav zahtjev u velikom broju slučajeva svodi na loše postavljen problem. Rješenje takvog problema može ne ovisiti neprekidno o ulaznim podacima, ne biti jedinstveno ili čak uopće ni ne postojati [12]. Zato je razvijen velik broj algoritama za prepoznavanje različitih uzoraka u slici rubnih elemenata. Neki od njih obrađuju samo binarnu sliku rubnih elemenata dok ostali zahtijevaju i sliku smjerova gradijenta.

Najjednostavniji pristup povezivanju rubnih elemenata je njihovo grupiranje u pravocrtne segmente. Taj pristup ima dobro svojstvo da je prilično općenit: može se koristiti za mnoge primjene računarskog vida, kod scena u kojima prevladavaju ravni bridovi i plohe. Primjeri takvih primjena su raspoznavanje objekata [28], određivanje strukture scene stereo vidom [16] te procjena kretanja u sceni pronalaženjem odgovarajućih značajki u susjednim slikovnim okvirima [47].

Jedan od češćih pristupa izdvajanju pravocrtnih segmenata se temelji na analizi parametarskog prostora dobivenog Houghovom transformacijom¹ za pravce koja je opisana u pododjeljku 4.2.1. Iako se radi o vrlo zanimljivoj ideji, izvorna HT se teško može uspješno

¹U daljnjem tekstu, za Houghovu transformaciju koristit će se pokratak *HT*.

primijeniti kod slika sa većim brojem pravocrtnih segmenata. U literaturi su stoga predložene brojne modifikacije izvornog postupka [13, 19, 35, 38, 75]. Pregled nekoliko važnijih članaka s tom tematikom je dan u pododjeljku 4.2.2, dok pododjeljak 4.2.3 opisuje modifikaciju koja je korištena u ovom radu. Analiza parametarskog prostora koji se dobiva tako modificiranom HT je opisana u odjeljku 4.2.4, dok odjeljak 4.2.5 opisuje sam postupak izlučivanja pravocrtnih segmenata koji se zasniva na rezultatima te analize. Reprezentativni podskup eksperimentalnih rezultata je prikazan u odjeljku 4.2.6, dok su izvedbeni detalji konačnog algoritma pokazani u odjeljku 4.2.7.

4.2.1 Houghova transformacija za pravce

HT je odavno poznata kao metoda za pronalaženje pravaca i drugih parametriziranih krivulja u slikama [4]. Nešto kasnije, ta je metoda generalizirana za pronalaženje proizvoljnih oblika [13]. Glavne prednosti pristupa pronalaženju pravocrtnih segmenata zasnovanog na HT u odnosu na druge pristupe (npr. ulančavanje rubnih elemenata [7] — *engl.* edge chaining) su otpornost na nepotpunu i netočnu sliku rubnih elemenata i ne zahtijevanje apriornog znanja o položaju pravocrtnih segmenata u analiziranoj slici.

Houghova transformacija

Houghova transformacija je prvi put uvedena 1962. godine kao metoda za prepoznavanje složenih uzoraka elemenata binarnih slika. U najopćenitijem obliku, Houghova transformacija [13] označava preslikavanje slikovnih elemenata u višedimenzionalni parametarski prostor geometrijskih značajki. Tom se transformacijom udaljeni dijelovi traženog uzorka preslikavaju u kompaktne nakupine značajki u parametarskom prostoru. Svaki element parametarskog prostora pri tome odgovara jednoj značajki (npr. pravcu, kružnici ili nekom složenijem uzorku). Parametarski prostor je podijeljen na povezana područja, a svakom području se pridružuje “akumulator” — cijeli broj koji opisuje mogućnost da značajke koje odgovaraju elementima tog područja postoje u izvornoj slici. U procesu transformacije, svaki slikovni element početne slike aditivno doprinosi (“glasa”) akumulatorima koji su pridruženi značajkama čija prisutnost je u skladu sa vrijednošću tog elementa. Može se stoga reći da slikovni elementi početne slike svjedoče o prisutnosti podskupa značajki za kojeg su glasali. Tako se primjenom HT težak problem globalnog pretraživanja prostora slikovnih elemenata može izraziti lakše rješivim problemom pronalaženja lokalnih maksimuma u skupu akumulatora koji odgovaraju elementima parametarskog prostora značajki.

Temeljna ideja metode može se ilustrirati na primjeru pronalaženja skupova kolinearnih elemenata zadane binarne slike [13]. Skup slikovnih elemenata koje leže na pravcu definiranom parametrima (k, l) može se opisati relacijom:

$$f((\hat{k}, \hat{l}), (x, y)) \equiv \hat{k} \cdot x + \hat{l} - y = 0. \quad (4.11)$$

Jednadžba (4.11) svakoj kombinaciji parametara (\hat{k}, \hat{l}) pridružuje skup slikovnih elemenata pa je definirano preslikavanje višeznačno, odnosno “(1:n)”. Simbol “ $\hat{}$ ” se pri tome koristi za označavanje veličina iz domene preslikavanja. HT se temelji na ideji da jednadžba (4.11) može shvatiti i kao preslikavanje u drugom smjeru koje svakom slikovnom elementu (x, y) pridružuje skup svih parametara pravaca koji kroz njega prolaze. To preslikavanje opisuje relacija

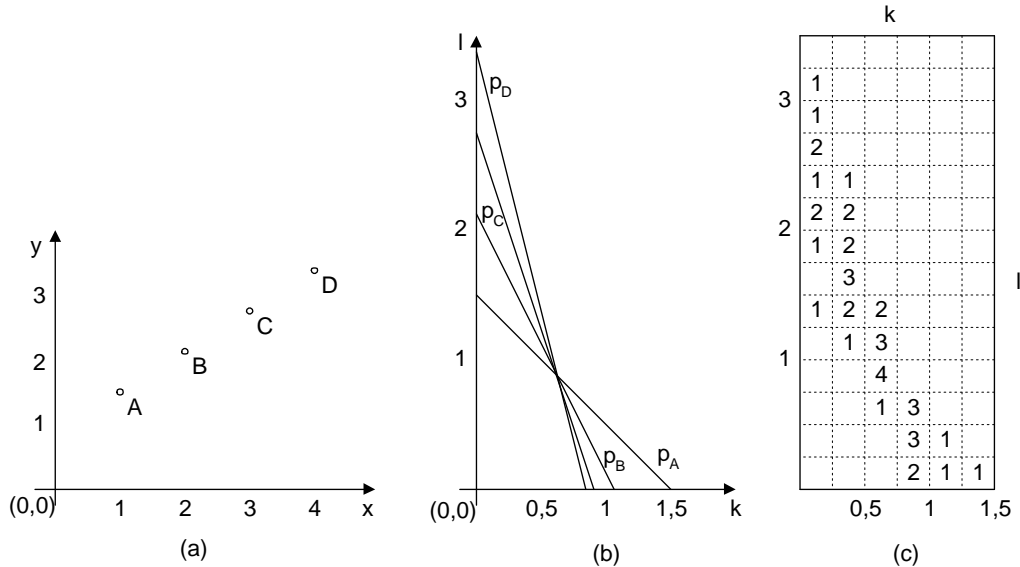
$$g((\hat{x}, \hat{y}), (k, l)) \equiv k \cdot \hat{x} + l - \hat{y} = 0. \quad (4.12)$$

Sl. 4.9(a) prikazuje binarnu sliku sa skupom kolinearnih elemenata $S = \{A, B, C, D\}$. Svakoj točki $s \in S$, jednadžba (4.12) pridružuje skup točaka p_s parametarskog prostora (sl. 4.9(b)), koji je u ovom slučaju pravac. Pravci p_s koji odgovaraju kolinearnim točkama slike se sijeku u zajedničkoj točki parametarskog prostora, a koordinate te točke definiraju zajednički pravac kolinearnih točaka u skladu sa jednadžbom (4.11). Može se reći da

HT ističe presjecišta pravaca u parametarskom prostoru jer će akumulatori koji su tim presjecištima pridruženi, nakon obavljene transformacije imati maksimalne vrijednosti — sl. 4.9(c). Određivanje takvih točaka parametarskog prostora je lokalna operacija (npr. uspoređivanje s pragom) i trebala bi biti bitno manje složena od pronalaženja raspršenih elemenata traženog uzorka u cijeloj slici. Analiza akumulatora u parametarskom prostoru na sl. 4.9(c) bi pokazala da se najveći skup približno kolinearnih točaka binarne slike na sl. 4.9(a) sastoji od četiri točke, te da se one nalaze na pravcima iz skupa

$$y = \hat{k} \cdot x + \hat{l}, \hat{k} \in [0,5, 0,75], \hat{l} \in [0,75, 1).$$

Nesigurnost metode pri tome ovisi o rezoluciji parametarskog prostora.

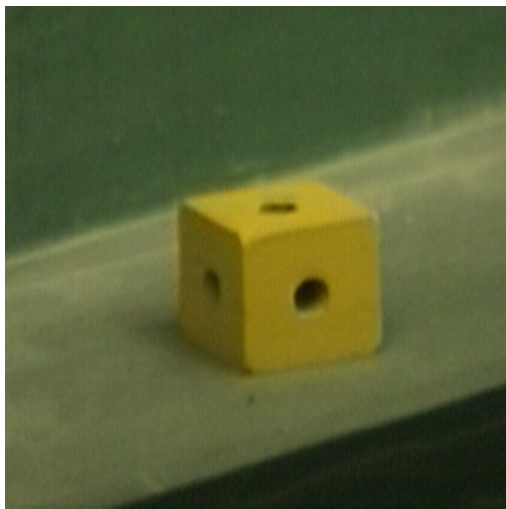


Slika 4.9: Primjer upotrebe Houghove transformacije za pronalaženje pravaca: binarna slika sa slikovnim elementima A, B, C i D (a), parametarski prostor sa odgovarajućim pravcima p_A , p_B , p_C i p_D , (b) te odgovarajuće polje akumulatora nakon transformacije (c).

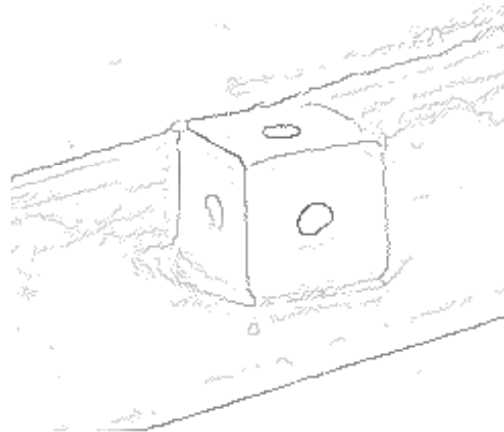
Primjena HT za pronalaženje pravaca

U ovom specijalnom slučaju HT, tražene značajke su pravci čiji se segmenti pojavljuju u početnoj slici. Pravocrtni segmenti izvorne slike intenziteta svjetlosti u sceni se mogu pronaći primjenom HT na sliku rubnih elemenata izvorne slike. Članak koji su Duda i Hart objavili još 1972. godine [13], opisuje jednu od prvih praktičnih izvedbi HT za pravce. Oni su sugerirali da se pravci najkorisnije mogu prikazati (θ, ρ) parametrizacijom jer se tako dobiva zatvoren parametarski prostor, a transformacija nema singularnih točaka — sl. 4.10(c). U tako definiranoj transformaciji, svaki rubni element binarne slike rubnih elemenata pridonosi akumulatorima parametarskog prostora (θ, ρ) , koji odgovaraju pravcima koji kroz taj rubni element prolaze. Pravci koji sadrže pravocrtne segmente u slici se na kraju mogu odrediti kao lokalni maksimumi u polju akumulatora parametarskog prostora — sl. 4.10(d).

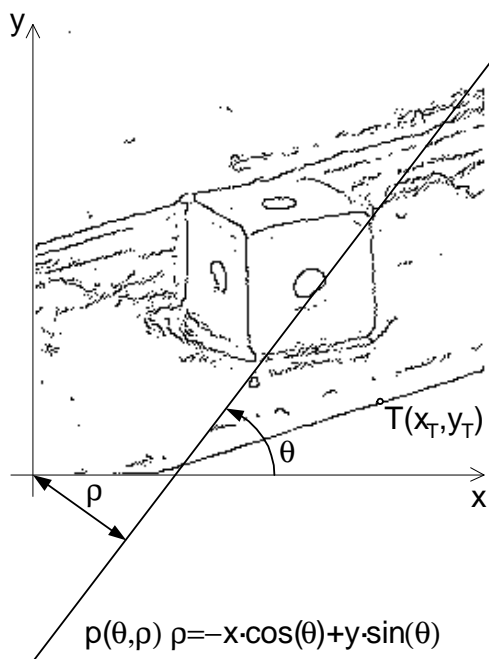
Sl. 4.10 prikazuje postupak pronalaženja pravaca u binarnoj slici rubnih elemenata dok je uvećani detalj transformirane slike prikazan na sl. 4.11. U svim slikama akumulatora parametarskog prostora, sive razine slikovnih elemenata su zbog jasnijeg prikaza obrnuto proporcionalne vrijednostima odgovarajućih akumulatora. Tako lokalni maksimumi u polju akumulatora odgovaraju tamnijim područjima transformirane slike. Slikovni elementi čija



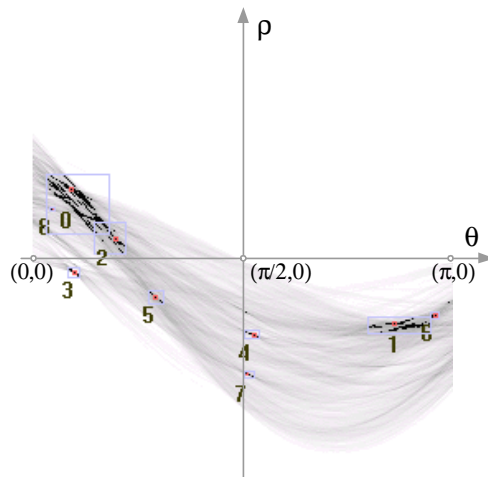
(a)



(b)



(c)



(d)

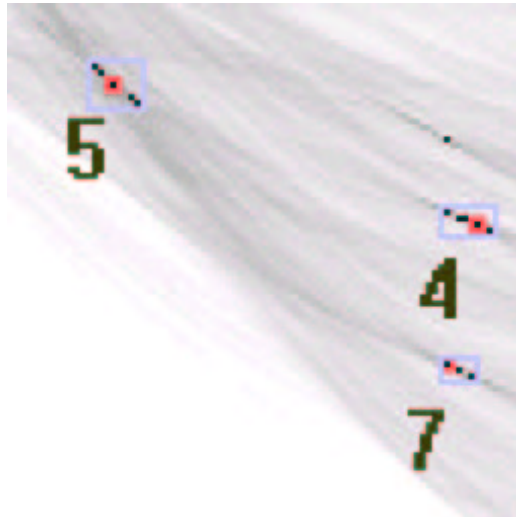
Slika 4.10: Izvorna slika (a), težinska slika rubnih elemenata (b), binarna slika rubnih elemenata (c) sa ucrtanim pravcem $p(\theta_p, \rho_p)$ koji odgovara lokalnom maksimumu #5 u polju akumulatora parametarskog prostora dobivenog Houghovom transformacijom (d).

vrijednost je veća od prvog praga² su označeni crnom bojom, dok su bliska ili povezana područja u kojima je suma vrijednosti takvih slikovnih elemenata veća od drugog praga proglašena lokalnim maksimumima.

Kad bismo zamolili nekog ljudskog promatrača da izdvoji pravocrtne segmente u izvornoj slici na sl. 4.10(a), on bi vrlo vjerojatno izdvojio devet segmenata koji čine kocku i dva ili tri segmenta pozadine. Nazovimo takve segmente markantnim segmentima izvorne slike. Analizom standardne HT binarne slike rubnih elemenata sl. 4.10(d), izdvojeno je devet pravaca od kojih samo sedam odgovaraju markantnim pravocrtnim segmentima izvorne slike. Dobiveni rezultati ukazuju na brojne nedostatke izvorne HT.

Glavni nedostatak HT je je neriješen problem pronalaženja “pravih” lokalnih maxi-

²Postupak pronalaženja lokalnih maksimuma u transformiranoj slici je opisan u odjeljku 4.2.4.



Slika 4.11: Uvećani detalj slike na sl. 4.10(d).

muma u polju akumulatora. U slučaju HT za pravce, pravi lokalni maksimumi su oni maksimumi polja akumulatora koji odgovaraju pravcima na kojima leži jedan ili više markantnih pravocrtnih segmenata izvorne slike. Iako markantni pravocrtni segmenti izvorne slike u velikom broju slučajeva odgovaraju maksimumima u polju akumulatora, posebno kod izvorne HT se često javljaju i anomalije od tog pravila.

1. “Lažni” lokalni maksimumi

Pravci definirani lokalnim maksimumima #2 i #5 transformirane slike na sl. 4.10 ne sadrže niti jedan markantni pravocrtni segment. Unatoč tome, vrijednost odgovarajućih akumulatora je relativno velika pa su oni izabrani među osam najvjerojatnijih kandidata za pravocrtne elemente u izvornoj slici.

2. Višestruki “pravi” maksimumi

Sl. 4.11 prikazuje uvećani detalj sa sl. 4.10(d). Pravi lokalni maksimumi #4 i #7 odgovaraju središnjem i desnom uspravnom bridu kocke dok je maksimum #5 lažan; tamno područje iznad maksimuma #4 odgovara lijevom uspravnom bridu kocke. Može se vidjeti da se područja koja odgovaraju svakom “lokalnom maksimumu” zapravo sastoje od većeg broja vrhova u transformiranoj slici te da postupak njihovog stapanja nije trivijalan.

3. Zaklanjajući učinak

Problemi lažnih i višestrukih pravih lokalnih maksimuma su često uzrokovani tzv. zaklanjajućim učinkom koji je inherentan za HT, ali se različitim modifikacijama može ublažiti. Zaklanjajući učinak se uvijek javlja u blizini jakih pravih lokalnih maksimuma (#1, sl. 4.10), a očituje se zaklanjanjem lokalnih maksimuma kraćih markantnih segmenata slike. Konačni rezultat je stapanje slabijih pravih maksimuma s jakim lokalnim maksimumom i pojava lažnih lokalnih maksimuma u njegovoj neposrednoj blizini (#2, sl. 4.10).

4. Osjetljivost na šum u slici rubova

U izvornoj HT za pravce, svaki slikovni element binarne slike rubova pridonosi svim akumulatorima koji odgovaraju pravcima koji kroz taj slikovni element prolaze. Tako je moguće da slučajan raspored rubnih elemenata potpomognut kratkim odsječkom kružnice uvjetuje jači maksimum u polju akumulatora od kontinuiranog lijevog uspravnog brida kocke (#5, sl. 4.10).

4.2.2 Pregled postojećih modifikacija HT za pravce

Lie i Chen [19] su probleme izvorne HT za pravce u binarnoj slici rubova rješavali interpolacijom pravca kroz svaki slikovni element, na temelju analize njegovog susjedstva dimenzija 7×7 . Autori su modifikaciju transformacije nazvali polariziranom jer rubni elementi pridonose samo onim pravcima čiji je nagib približno jednak nagibu interpoliranog pravca. Doprinos rubnih elemenata je naknadno potisnut, proporcionalno razlici u nagibu pravaca interpoliranih na temelju 3×3 i 7×7 susjedstava. Tako su doprinosi dugih ravnih nizova rubnih elemenata dodatno naglašeni. Lokalni maksimumi u parametarskom prostoru su pronađeni jednostavnim uspoređivanjem s empirijski određenim globalnim pragom. Eksperimentalni rezultati na umjetnim slikama su vrlo dobri, dok eksperimenti sa stvarnim slikama nisu provedeni. Opisani postupak ne koristi informaciju sadržanu u slici smjerova gradijenta, a dosta je složen jer je kroz svaki element slike rubova potrebno interpolirati pravac na temelju analize susjedstva.

Foresti et al. [38] su pokušali minimizirati gubitak prostorne informacije u parametarskom prostoru koji je inherentan za pristupe temeljene na HT. Na početku predloženog postupka izlučivanja pravocrtnih segmenata, obavlja se preliminarno grupiranje elemenata slike rubova, na temelju svojstva malene lokalne zakrivljenosti. Svakoj dobivenoj nakupini se pridružuje oznaka, a na početni skup rubnih elemenata se primjenjuje modifikacija HT koja uzima u obzir i oznake nakupina pojedinih rubnih elemenata. Smjer lokalnog gradijenta je pri tome također iskorišten pa je svaki rubni element pridonosio samo onim pravcima koji su približno okomiti na lokalni smjer gradijenta. Konačno, lokalni maksimumi kojima pridonose prostorno udaljeni rubni elementi se odbacuju kao lažni na temelju provjere oznaka nakupina kojima rubni elementi pripadaju. Autori su pokušali napraviti potpuno prilagodljivi algoritam na temelju analize statističkih svojstava izvorne slike, iako konačni postupak ipak nije bio potpuno neovisan o parametrima. Pokazani su dobri rezultati eksperimentiranja na slici složene stvarne scene.

Kako bi smanjili greške koje se javljaju prilikom diskretizacije, Palmer et al. [35] su predložili modifikaciju HT u kojoj elementi slike rubova doprinose s različitim faktorima pravcima koji prolaze kroz njihovo susjedstvo. Ti faktori su organizirani u matrici koja se naziva glasačkom jezgrom, a obavljeni su eksperimenti sa različitim oblicima i veličinama glasačkih jezgri. U novijem članku, isti autori su opisali optimizacijsku metodu pronalazjenja lokalnih maksimuma u polju akumulatora pridruženog parametarskom prostoru [72]. Prikazani su i vrlo dobri eksperimentalni rezultati na slikama složenih stvarnih scena, međutim postignuto vrijeme izvođenja nije navedeno.

4.2.3 Odabrana modifikacija izvornog postupka

Na temelju iskustava opisanih u prethodnom odjeljku, razvijena je transformacija kojom se uz čim manji utrošak vremena pokušava ostvariti čim bolja razlučivost pravih lokalnih maksimuma u parametarskom prostoru.

Polarizacija Houghove transformacije

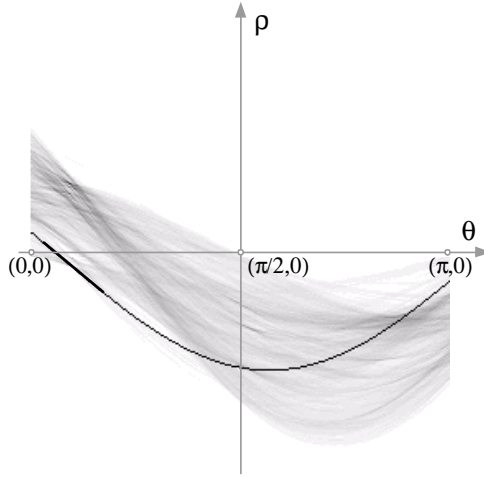
Velik broj problema vezanih uz izvornu HT je uzrokovan time što svaki rubni element doprinosi velikom broju pravaca koji ne odgovaraju granici u njegovoj blizini — sl. 4.12. Taj problem se može rješavati polarizacijom HT, tako da svaki rubni element $T = (x_T, y_T)$ glasa samo za podskup P_T svih pravaca koji kroz njega prolaze. U tom slučaju, potrebno je odrediti smjer tangente na granicu objekta u točki izvorne slike koja odgovara rubnom elementu i definirati P_T kao skup pravaca koji su približno jednaki određenoj tangenti [19]. Dobro bi bilo da se tangenta određuje lokalno, u nevelikom susjedstvu rubnog elementa T , kako bi se izbjegli složeni proračuni na razini slikovnih elemenata i preduga vremena izvođenja. Najjednostavniji način za određivanje podskupa P_T je na temelju lokalnog

smjera gradijenta φ_T u rubnom elementu T [38]:

$$P_T = \{p \equiv \rho = -x_T \cdot \cos(\theta) - y_T \cdot \sin(\theta), |\theta - \varphi_T - \frac{\pi}{2}| < \Delta\varphi\} \quad (4.13)$$

U tom slučaju, potrebno je prije pokretanja algoritma osigurati pored slike rubnih elemenata i sliku smjerova gradijenta.

U opisanoj modifikaciji, rubni elementi glasaju samo za pravce koji su približno okomiti na lokalni smjer gradijenta pa se kaže da su njihovi doprinosi polarizirani s obzirom na gradijent. Pri tome se kao “približno okomit” definira kut koji od 90° odstupa manje od parametra $\Delta\varphi$ koji je nazvan *širinom transformacije*. Eksperimentiranjem je utvrđeno da se dobri rezultati postižu za $\Delta\varphi \in [10^\circ, 20^\circ]$. Ako se odabere $\Delta\varphi = 10^\circ$, potrebno je izračunati doprinos rubnog elementa za samo jednu devetinu svih pravaca koji kroz njega prolaze što ubrzava postupak transformacije devet puta. Ova modifikacija potiskuje zaklanjajući učinak koji je obično uzrokovan kolinearnim i međusobno udaljenim rubnim elementima sa stohastičkim smjerovima gradijenta. Većina takvih rubnih elemenata neće doprinosti akumulatoru koji odgovara pravcu koji te elemente spaja jer je malena vjerojatnost da će gradijent u njima biti točno okomit na smjer pravca.



Slika 4.12: Tanka i debela crna krivulja označavaju doprinosi točke T sa sl. 4.10 transformiranoj slici u skladu sa izvornom odnosno polariziranom HT, uz $\Delta\varphi = 10^\circ$.

Sl. 4.12 prikazuje doprinose točke T sa sl. 4.10 transformiranoj slici kod izvorne i polarizirane HT. Markantni pravocrtni segment koji sadrži točku T odgovara lokalnom maksimumu #3 na sl. 4.10. Međutim, po izvornoj HT, točka T , doprinosi i maksimumima #1 i #7, koji odgovaraju pravocrtnim segmentima koji ne prolaze kroz tu točku.

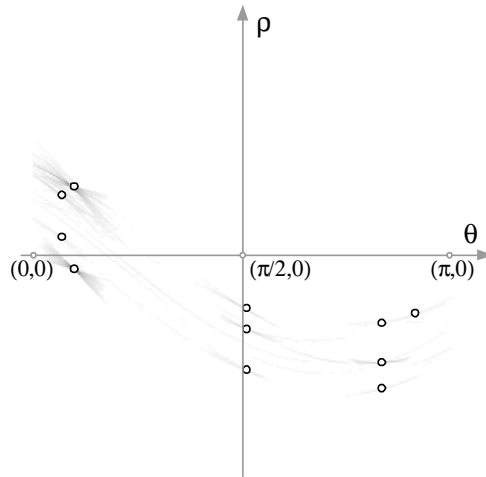
Primjena iznosa gradijenta u Houghovoj transformaciji

Drugi problem vezan uz izvornu HT je njena osjetljivost na rubne elemente koji su rezultat šuma ili tekstone ravnih površina scene. Takvi rubni elementi u izvornoj HT stohastički doprinose mnogim akumulatorima parametarskog prostora i tako povećavaju zaklanjajući učinak i podižu razinu šuma u transformiranoj slici. Nažalost, pokazuje se da se takvi rubni elementi ne mogu potpuno isfiltrirati iz gradijentne slike ni u ravnomjerno osvijetljenoj sceni, s ručnim podešavanjem parametara algoritma za pronalaženje rubnih elemenata. Još manje je to slučaj kod analize slika prostorno udaljenih scena, sa različitim brojem bridova u sceni i nekontroliranim uvjetima osvijetljenja.

Promotrimo gornji desni kut sl. 4.10(c). Rubni elementi u tom dijelu slike su prilično neorganizirani jer odgovarajuća ploha (školska ploča) ima naglašenu i nepravilnu teksturu. Iako ljudski promatrač može reći da su rubni elementi između ploče i svjetlije podloge

važniji od ostalih u tom području u kontekstu analize slike, za izvornu HT svi rubni elementi tog područja jednako vrijedno doprinose transformiranoj slici. Predlaže se stoga težinska modifikacija HT koja na ulazu zahtijeva težinsku (tj. ne binarnu) sliku rubnih elemenata. Težina rubnih elemenata se prilikom glasanja, upotrebljava kao doprinos odgovarajućim akumulatorima parametarskog prostora.

Sl. 4.13 prikazuje rezultat primjene težinske polarizirane HT na sl. 4.10(b). Vidi se da se akumulatori koji odgovaraju markantnim pravocrtnim segmentima slike znatno bolje ističu nego kod izvorne HT (sl. 4.10(d)). Kod ovakvog pristupa, mora biti zadovoljen uvjet da je iznos gradijenta tekstone manji od iznosa gradijenta na granicama objekata. Eksperimenti su međutim pokazali da sve izglađene slike razmatranog okoliša taj uvjet zadovoljavaju. Analiza tekstone je sama za sebe vrlo težak problem koji, srećom, u ovom kontekstu ne treba rješavati.



Slika 4.13: Polje akumulatora parametarskog prostora dobiveno na temelju sl. 4.10(b), težinskom polariziranom HT. Elementi parametarskog prostora koji odgovaraju markantnim pravocrtnim segmentima izvorne slike su označeni kružićima.

4.2.4 Analiza parametarskog prostora

Problem traženja ekstrema 1D i 2D funkcija je vrlo čest u računarskom vidu i obično se rješava na temelju znanja o njihovom broju, položaju i visini. Kod problema pronalženja maksimuma u parametarskom prostoru HT za pravce, znanje o broju i položaju maksimuma ne postoji: u slici može biti proizvoljan broj proizvoljno lociranih pravocrtnih segmenata. Međutim, iznos lokalnih maksimuma koji odgovaraju markantnim pravocrtnim segmentima slike može se procijeniti ako se postulira da se markantni pravocrtni segmenti sastoje od rubnih elemenata sa relativno velikim težinama. Tako se minimalni iznos elementa parametarskog prostora koji odgovara markantnom pravocrtnom segmentu može ocijeniti na temelju statističkih svojstava težinske slike rubnih elemenata po jednadžbi

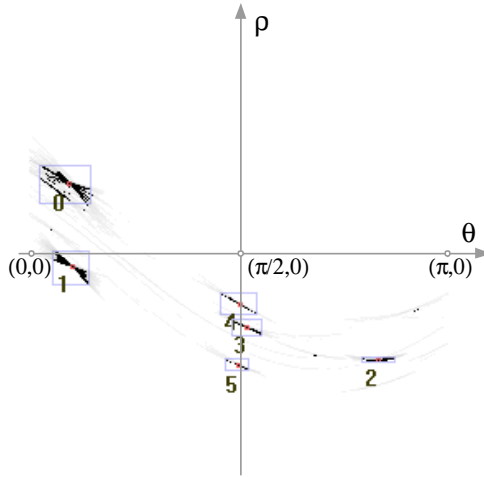
$$h_{prag} = l_{min} \cdot r_{prag}. \quad (4.14)$$

U jednadžbi (4.14), l_{min} označava minimalnu duljinu najmanje izraženog pravocrtnog segmenta izvorne slike kojeg se želi pronaći, dok je r_{prag} prosječna težina rubnih elemenata koji takav segment sačinjavaju. U provedenim eksperimentima, zadovoljavajući rezultati su postignuti za l_{min} postavljen na dvadesetinu manje dimenzije izvorne slike. Za r_{prag} se može odabrati vrijednost iz raspona

$$r_{prag} \in [r_{min}, r_{\mu}], \quad (4.15)$$

gdje su r_{min} i r_{μ} najmanja i srednja težina svih elemenata slike rubova.

Elementi parametarskog prostora koji su veći od h_{prag} se naknadno koriste kao klice za rast područja [21]. Problem višestrukih “pravih” maksimuma koji je vidljiv na sl. 4.11 se može rješavati ili stapanjem bliskih nakupina ili odbacivanjem nakupina s relativno malom površinom. Konačno, lokalni maksimumi se određuju kao točke s najvećom vrijednošću u pojedinoj nakupini — sl. 4.14.



Slika 4.14: Lokalni maksimumi koji su pronađeni analizom slike na sl. 4.13(b), sortirani po vrijednostima odgovarajućih akumulatora. Treba primijetiti da područje #0 ipak obuhvaća dva akumulatora koji odgovaraju markantnim pravocrtnim segmentima na sl. 4.10(a).

4.2.5 Izlučivanje pravocrtnih segmenata

U ovom odjeljku se opisuje postupak pronalaženja povezanih pravocrtnih skupova elemenata slike rubova. Postupak se temelji na ideji da se pretraživanje slike rubova ograniči samo na pravce koji odgovaraju lokalnim maksimumima u polju akumulatora parametarskog prostora (sl. 4.14).

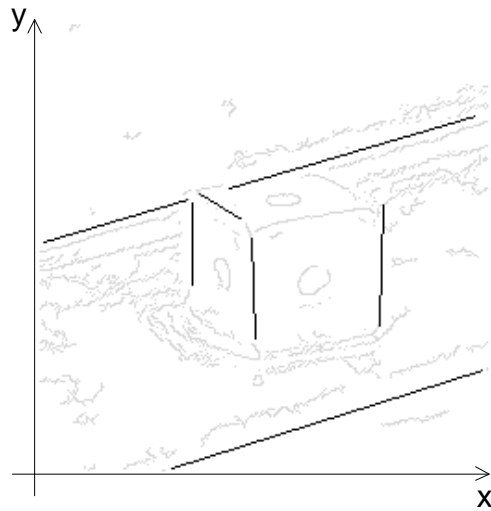
Temeljni postupak

Prilikom traženja pravocrtnih segmenata na nekom od određenih pravaca, u obzir se uzimaju samo rubni elementi za koje smjer gradijenta ne odstupa od normale pravca više od $\Delta\varphi$ koji je korišten u jednadžbi 4.13. Svaki izlučeni pravocrtni segment se opisuje težinom koja je proporcionalna sumi težina rubnih elemenata koji ga podržavaju podijeljenoj s duljinom segmenta. Rezultati opisanog postupka na temelju izvorne slike sa sl. 4.10(b) i analize parametarskog prostora na sl. 4.14 su prikazani na sl. 4.15.

Poboljšanje rezultata iteracijom

Vrlo je teško, ako ne i nemoguće pronaći sve lokalne maksimume koji odgovaraju bridovima scene čak i ako se upotrijebi predložena modifikacija HT. Razlog tome je zaklanjajući učinak koji se, iako bitno smanjen predloženim modifikacijama, ne može potpuno eliminirati jer je on inherentno svojstvo HT. Ipak, predložena modifikacija HT vrlo pouzdano nalazi najizraženije pravocrtne segmente izvorne slike pa se kao rješenje predlaže iterativno ponavljanje slijedećih koraka obrade:

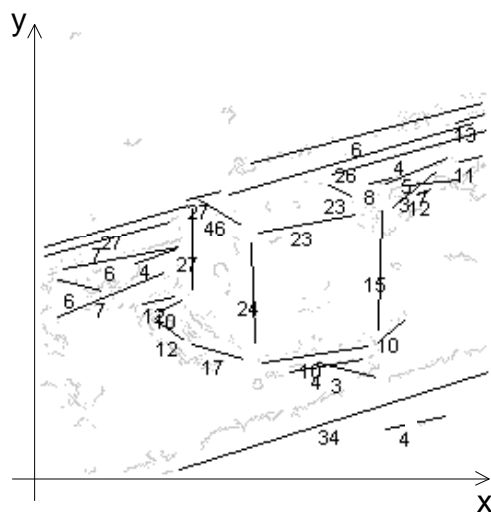
1. slika rubova se transformira modificiranom HT;



Slika 4.15: Pravocrtni segmenti u slici na sl. 4.10(b) koji se nalaze na pravicima koji odgovaraju lokalnim maksimumima na sl. 4.14.

2. traže se obećavajući lokalni maksimumi u parametarskom prostoru, u skladu s parametrom r_{prag} ;
3. pronalaze se pravocrtni segmenti slike rubova koji odgovaraju pronađenim maksimumima;
4. elementi slike rubova koji odgovaraju pronađenim segmentima se brišu, a obrada se vraća na prvi korak.

Opisani postupak se ponavlja sve dok se u četvrtom koraku ne pronade niti jedan novi pravocrtni segment. Parametar r_{prag} se na početku postavlja konzervativno ($r_{prag} = 2 \cdot r_{\mu}$), a u višim iteracijama se postepeno snižava prema r_{min} . Pokazuje se da se ovakvim "strpljivim" pristupom određivanju parametra r_{prag} smanjuje zaklanjajući učinak, a tako i broj lažnih lokalnih maksimuma u parametarskom prostoru te broj potrebnih iteracija za obavljanje postupka.

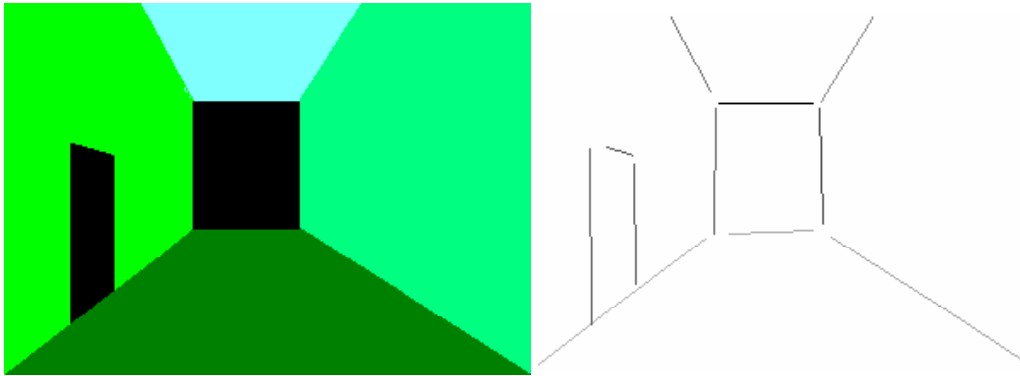


Slika 4.16: Pravocrtni segmenti u sl. 4.10(b) koji su pronađeni nakon 11. iteracije postupka. Izlučeni segmenti su označeni pridruženim težinama.

4.2.6 Eksperimentalni rezultati

Performansa predloženog postupka je ispitana na tri grupe slika kako bi se provjerila širina njegove primjenljivosti. Sve ispitne slike su obrađene bez zasebnog podešavanja parametara procedure, dok su slike stvarnih scena pribavljene bez korištenja specijalnih izvora osvjetljenja.

Prva ispitna grupa se sastoji od slika koje su umjetno sintetizirane metodama računarske grafike. Primjer takve slike i odgovarajućih rezultata obrade je dan na sl. 4.17. Svrha ispitivanja je u ovom slučaju bila pokazati performansu procedure u idealnim i kontroliranim uvjetima.



Slika 4.17: Sintetizirana slika koja prikazuje jednostavnu scenu zatvorenog prostora i izlučeni pravocrtni segmenti nakon 3. iteracije postupka. Tamnijim pravocrtnim segmentima pridružena je veća težina.

Druga ispitna grupa sadrži slike jednostavnih poliedara. Izlučivanje pravocrtnih segmenata u takvim slikama je vrlo pogodno obavljati pristupom uz pomoć HT jer sadrže relativno malen broj relativno dugačkih markantnih pravocrtnih segmenata. Takvi pravocrtni segmenti se u polju akumulatora parametarskog prostora manifestiraju izdvojenim visokim lokalnim maksimumima koje je relativno lako pronaći. Postignuti rezultati su prikazani na sl. 4.16, sl. 4.18 i sl. 4.19 potvrđuju očekivanja se predloženim pristupom mogu pronaći uglavnom svi markantni pravocrtni segmenti slike. Rezultati obrade sl. 4.18(b) nisu idealni zbog kromatske aberacije kamere koja se manifestira slabom kvalitetom slike na granici između zelene ploče i crvene prizme.

Konačno, treća grupa eksperimenata je obavljena na složenim scenama zatvorenog prostora u realističnim uvjetima. Analizirane su noćne i dnevne slike, koje su pribavljene pod uobičajenom stropnom rasvjetom, odnosno pod dnevnim svjetlom. Dobiveni rezultati nisu idealni, ali se čini da je to najbolje što se može dobiti kad apriorno znanje o sceni nije dostupno. Rezultati slika dnevnih scena su prikazani na sl. 4.20 dok se rezultati za noćne scene mogu vidjeti na sl. 4.21.

Zahvaljujući pažljivim oblikovanju postupka pronalaženja rubnih elemenata i iterativnog izlučivanja pravocrtnih segmenata uz pomoć HT, postignuta vremena izvođenja obećavaju mogućnost obrade u stvarnom vremenu u skoroj budućnosti. Vrijeme trajanja obrade svih ispitnih slika na računalu sa performansom od 4 SPECint95, bilo je nešto kraće od 2 s.



(a)



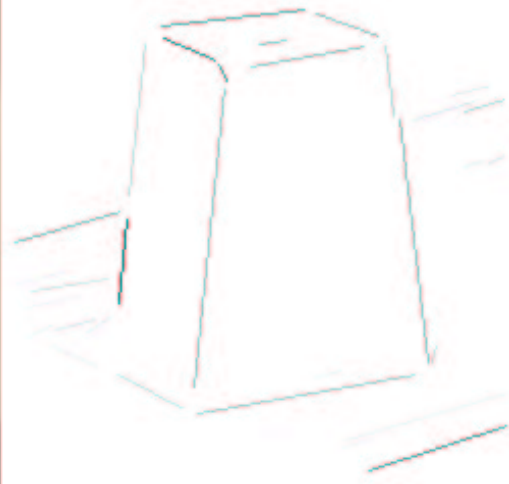
(b)



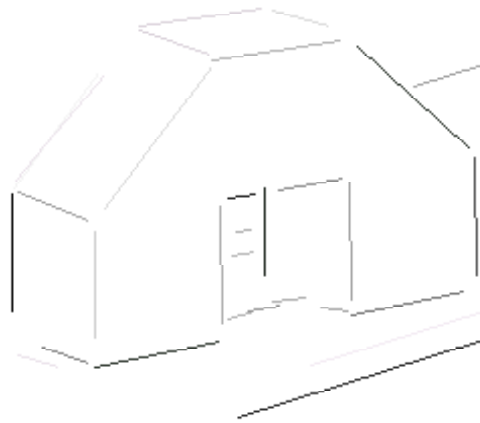
Slika 4.18: Rezultati obrade slika jednostavnih poliedara. Obrada je trajala 11 (a) odnosno 15 iteracija (b). Tamnijim pravocrtnim segmentima pridružena je veća težina.



(a)



(b)



Slika 4.19: Rezultati obrade slika jednostavnih poliedara. Obrada je trajala 10 (a) odnosno 12 iteracija (b). Tamnijim pravocrtnim segmentima pridružena je veća težina.



(a)



(b)



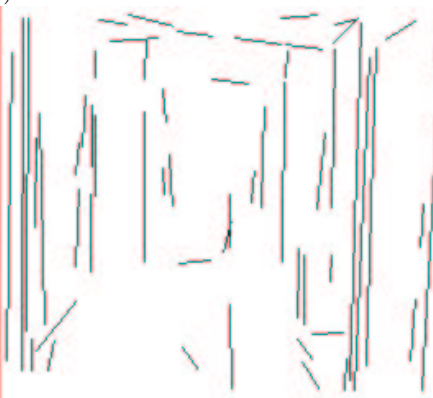
Slika 4.20: Rezultati obrade dnevnih slika hodnika. Obrada je trajala 16 (a), odnosno 14 iteracija (b). Tamnijim pravocrtnim segmentima pridružena je veća težina.



(a)



(b)

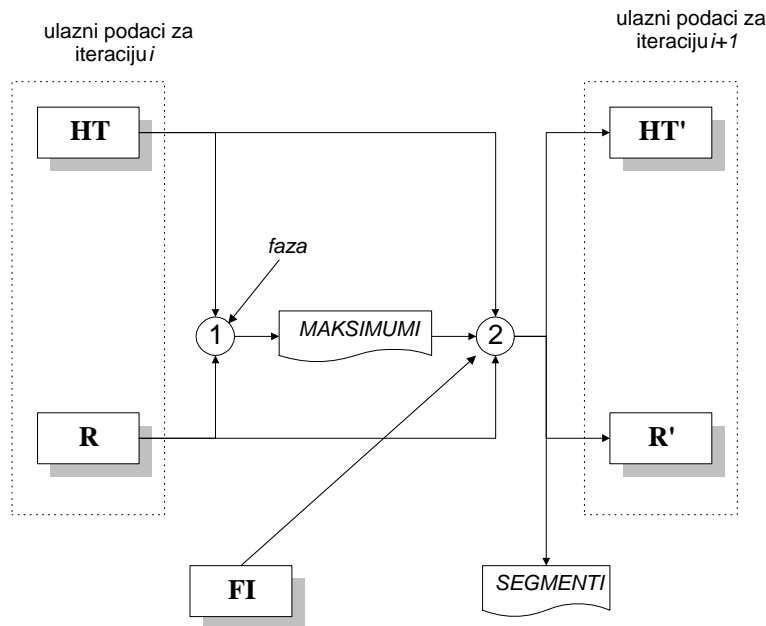


Slika 4.21: Rezultati obrade noćnih slika hodnika. Obrada je trajala 12 (a), odnosno 14 iteracija (b). Tamnijim pravocrtnim segmentima pridružena je veća težina.

4.2.7 Izvedbeni detalji algoritma

Predloženi algoritam za izlučivanje pravocrtnih segmenata iz zadanih slika težinskih rubova i smjerova gradijenta se sastoji od slijedeća dva koraka:

- (a) Inicijalizacijski postupak koji obuhvaća primjenu težinske polarizirane HT na težinsku sliku rubova \mathbf{R} (sl. 4.10(b)) i sliku smjerova gradijenta \mathbf{FI} . Rezultat te operacije je slika akumulatora parametarskog prostora \mathbf{HT} (sl. 4.13). Pored toga, određuje se i histogram slike \mathbf{R} .
- (b) Upravljanje iterativnim ponavljanjem slijedećih dvaju koraka, na temelju vrijednosti parametra *faza*. Shematski prikaz obrade u pojedinim iteracijama je prikazan na sl. 4.22
 1. Pronalaženje lokalnih maksimuma u slici \mathbf{HT} u skladu s vrijednošću parametra *faza* i histogramom slike \mathbf{R} . Rezultat te operacije (sl. 4.14) je lista točaka parametarskog prostora *MAKSIMUMI*.
 2. Traženje pravocrtnih segmenata na pravcima koji odgovaraju točkama parametarskog prostora elementima liste *MAKSIMUMI* i priprema slike \mathbf{R} i njenog histograma te slike \mathbf{HT} za slijedeću iteraciju postupka. Tom operacijom, pronađeni pravocrtni segmenti (sl. 4.15) se smještaju u izlaznu listu pravocrtnih segmenata *SEGMENTI*.



Slika 4.22: Prikaz toka obrade u jednoj iteraciji algoritma. Pravokutnici označavaju slike i strukture podataka, dok krugovi označavaju korake obrade (opširnije u tekstu).

Slijedi opis inicijalizacijskog algoritma, algoritma za upravljanje iteracijom te izvedbenih detalja za svaki od koraka pojedinih iteracija.

Inicijalizacijski algoritam

Prije svega, brišu se svi elementi liste pravocrtnih segmenata *SEGMENTI*. Nakon toga se određuje početni histogram slike \mathbf{R} koji će se kasnije koristiti za odabir praga binarizacije slike akumulatora parametarskog prostora \mathbf{HT} .

Svi elementi slike **HT** se postavljaju na nulu. Konačno, za svaki element težinske slike rubova $r = \mathbf{R}[i, j] \neq 0$, određuje se doprinos slici **HT** u skladu sa slikom **FI**, po slijedećem algoritmu:

```

procedura doprinos(i, j, težina)
    korak $_{\rho}$  := 2 · R.diagonala/HT.visina_slike
    ishodište $_{\rho}$  := HT.visina_slike/2
    korak $_{\theta}$  :=  $\pi$ /HT.širina_slike

    //dohvat i normalizacija smjera gradijenta
     $\varphi$  := FI[i, j] -  $\frac{\pi}{2}$ 
    dok  $\varphi < 0$ 
         $\varphi$  =  $\varphi + \pi$ 
     $\theta$  :=  $\varphi - \Delta\varphi$ 
     $\theta_{KRAJ}$  :=  $\varphi + \Delta\varphi$ 
    //pretvaranje  $\theta$  u odgovarajući indeks slike HT
    l :=  $\theta$  / korak $_{\theta}$ 
    dok  $\theta < \theta_{KRAJ}$ 
         $\rho$  :=  $-j * \cos(\theta) + i * \sin(\theta)$ 
        //pretvaranje  $\rho$  u odgovarajući indeks slike HT
        k :=  $\rho$  / korak $_{\rho}$  + ishodište $_{\rho}$ 
        HT[k, l] := HT[k, l] + težina
        l := l + 1
         $\theta$  :=  $\theta + korak_{\theta}$ 

    kraj

```

Parametar $\Delta\varphi$ predstavlja širinu transformacije, a opisan je u odjeljku 4.2.3. Dimenzije slike **HT** su 256×256 slikovnih elemenata, a odabrane su empirijskim putem.

Upravljački algoritam

Tok iteracije nadzire poseban algoritam koji određuje vrijednost parametra drugog koraka obrade *faza* na temelju rezultata trećeg koraka obrade u prethodnoj iteraciji. Obrada se izvodi u četiri ciklusa iteracija, za vrijednosti upravljačkog parametra *faza* = 0, 1, 2, 3:

```

faza=0
dok faza<4
    stari_broj_segmenata=SEGMENTI.brojnost()
    trazi_pravocrtne_segmente(faza)
    novi_broj_segmenata=SEGMENTI.brojnost()
    ako stari_broj_segmenata==novi_broj_segmenata
        faza:=faza+1

```

Pronalaženje lokalnih maksimuma u slici **HT**

Lokalni maksimumi u slici **HT** se pronalaze pomoću posebnog algoritma koji pronalazi povezana područja sa slijedećim svojstvima:

1. Vrijednost sive razine svakog slikovnog elementa područja je veća od $PRAG_E$;
2. Zbroj vrijednosti sivih razina svih elemenata područja (težina područja) je veća od $PRAG_P$;

Za svako pronađeno područje, algoritam sprema koordinate slikovnog elementa s najvećom težinom u listu *MAKSIMUMI*.

Parametri $PRAG_E$ i $PRAG_P$ se odabiru na temelju slijedećih vrijednosti:

1. r_μ — srednja vrijednost svih težina elemenata slike rubova;
2. r_{min} — donja granica težina svih elemenata slike rubova;
3. upravljački parametar *faza*.

Očekuje da će težina područja biti znatno veća od težina pojedinačnih akumulatora jer elementi slike \mathbf{R} doprinose većem broju akumulatora. Utjecaj takvog glasanja na težinu područja je modeliran parametrom *doprinos*. Heuristički algoritam odabira parametara je slijedeći:

```

korak $\theta$  :=  $\pi / \mathbf{HT}.širina\_slike$ 
doprinos :=  $\Delta\varphi / korak_\theta$ 

ako faza == 0
     $PRAG_E := 2 \cdot r_\mu$ 
     $PRAG_P := PRAG_E \cdot doprinos$ 
inače ako faza == 1
     $PRAG_E := r_\mu$ 
     $PRAG_P := PRAG_E \cdot doprinos$ 
inače ako faza == 2
     $PRAG_E := (r_\mu + r_{min}) / 2$ 
     $PRAG_P := PRAG_E \cdot doprinos / 2$ 
inače ako faza == 3
     $PRAG_E := r_{min}$ 
     $PRAG_P := PRAG_E \cdot doprinos / 2$ 

```

Traženje pravocrtnih segmenata

Za svaku točku $P \in \mathbf{HT}$ koja se nalazi u listi *MAKSIMUMI*, obavlja se slijedeća procedura:

1. Određuje se pravac $p \equiv \rho_P = -x \cos(\theta_P) + y \sin(\theta_P)$ slike \mathbf{R} koji odgovara točki P slike \mathbf{HT} ;
2. Traže se svi elementi slike \mathbf{R} koji leže na pravcu p , a smjer gradijenta im ne odstupa od $\theta_P - \frac{\pi}{2}$;
3. Povezani skupovi od više od 5 takvih slikovnih elemenata se proglašavaju pravocrtnim segmentima i opisuju krajnim točkama te težinom koja odgovara zbroju težina pojedinih elemenata.
4. Svaki tako pronađeni pravocrtni segment se smješta u listu *SEGMENTI*.

Kako bi se izbjegle diskretizacijske pogreške u parametarskom Prostoru i posljedice neidealne slike rubnih elemenata, tolerira se odstupanje pravocrtnog segmenta od pravca u iznosu od jednog slikovnog elementa. Iz istog razloga, dozvoljavaju se “rupe” u pravocrtnim segmentima duljine do dva slikovna elementa.

Kako bi se smanjio zaklanjajući efekt u slijedećoj iteraciji, potrebno je obrisati elemente slike \mathbf{R} čiji su pravocrtni segmenti pronađeni te uskladiti histogram te sliku \mathbf{HT} sa unesenim promjenama. Za svaki takav element $r = \mathbf{R}[i, j]$, stoga se izvodi slijedeća metoda od tri koraka:

1. poziva se procedura **doprin** koja je opisana u koraku inicijalizacije postupka, sa parametrima $i, j, -r$;
2. element histograma za vrijednost r se umanjuje za jedan;
3. Slikovni element $\mathbf{R}[i, j]$ se postavlja na nulu.

4.3 Ocjena rezultata pretprocesiranja

Pored efikasnosti, glavni kriteriji oblikovanja algoritama za pretprocesiranje bila je prilagodljivost na različite uvjete osvjetljenja u sceni. Sve vrijednosti praga u algoritmima se stoga odabiru dinamički, kao funkcije statističkih svojstava ulazne slike. Unatoč tome, subjektivna kvaliteta dobivenih rezultata dosta ovisi o analiziranoj slici. Najbolji rezultati se postižu kada se slika pribavlja pri strogo kontroliranim uvjetima osvjetljenja, za specijalno odabrane položaje kamere kod kojih su utjecaji refleksija i sjena zanemarivi. Takve intervencije međutim nisu prihvatljive u kontekstu primjene algoritama za lokalizaciju robota koji je prilagodljivi istraživač.

Postignuti rezultati pretprocesiranja slika stvarnih scena nisu ocijenjeni dovoljno dobrim za pristup tumačenju scene na klasičan način [2]. Lebegue i Aggarwal [34] međutim tvrde kako se rezultati algoritama za pravocrtni opis slike mogu bitno poboljšati ako su poznati smjerovi pružanja bridova u sceni. Shakunaga [29] je kao prvi korak analize slika hodnika predložio određivanje smjera pružanja hodnika jer je podudaranje značajki slike s 3D modelom bitno otežano kada izlučene značajke znatno ovise o smjeru gledanja kamere. Sličan pristup je odabran i u [30, 55, 56], a u skladu je s pristupom lokalizaciji prepoznavanjem referentnih objekata u kojem je približni položaj kamere u okolišu unaprijed poznat.

Gotovo svi stvarni hodnici se sastoje od bridova s dva vodoravna (*uzdužnim* i *poprečnim*) i jednim *uspravnim* smjerom pa se mogu modelirati kao ortogonalne triedarske scene. Ako se uvaži da se uspravne osi koordinatnih sustava robota i hodnika poklapaju (sl. 1.1), ostaju nepoznati samo uzdužni i poprečni smjer strukture hodnika u koordinatnom sustavu robota. Kad bi slike scene bile dobivene ortogonalnom projekcijom, slike svih uzdužnih odnosno poprečnih bridova scene bi bile paralelne pa bi se ti smjerovi mogli odrediti jednostavnim grupiranjem. Međutim, ortogonalna projekcija nije prikladna aproksimacija pribavljanja slike kod scena s izraženom trodimenzionalnom strukturom (tj. za scene sa značajnom dubinom). Kao što je već naglašeno u odjeljku 3.1, pribavljanje slike kod takvih scena mora se opisati perspektivnom projekcijom koja točno modelira postupak formiranja slike u idealnoj kameri. Odnosi pravaca scene i odgovarajućih pravocrtnih segmenata slike kod perspektivne projekcije nisu jednostavni kao kod ortogonalne projekcije ali omogućuju donošenje *dalekosežnijih* zaključaka o strukturi scene, što će se pokazati u slijedećem poglavlju.

Poglavlje 5

Lokalizacija u sustavu hodnika

U ovom poglavlju, opisana je lokalizacijska procedura za pokretnog samostalnog robota koji se nalazi u sustavu hodnika. Pretpostavljeno je da hodnici imaju ortogonalnu triedarsku strukturu te da u njima nisu postavljeni umjetni navigacijski znakovi. Smjer uspravne strukturne osi scene se ističe od ostalih smjerova u sceni jer odgovara lokalnom smjeru polja gravitacije. Robotu je taj smjer poznat jer može biti ili određen jednostavnim mjernim uređajima ili, u slučaju vodoravne podloge, apriorno poznat. Jedini senzor koji robot ima na raspolaganju je jedna kamera s mogućnošću podešavanja zakreta i nagiba.

Predložena procedura je zamišljena kao temeljni navigacijski postupak, koji bi se primjenjivao u situacijama u kojima nikakvo znanje o položaju robota u okolišu nije na raspolaganju. Takve iznimne situacije uključuju “buđenje” robota [16] nakon periodičkog gašenja zbog održavanja, ulazak u novi okoliš npr. nakon izlaska iz dizala te pojavu kontradiktornih zaključaka u hijerarhijski višim razinama odlučivanja. U tom kontekstu, robustnost procedure je važnija od brzine izvođenja.

Svojstvo promatranih scena je da sadrže velik broj ravnih paralelnih bridova koji imaju relativno mali broj smjerova pružanja u 3D prostoru. Posebno prikladna metoda za tumačenje takvih scena je određivanje nedogleda jer se time istovremeno može postići i grupiranje pravocrtnih segmenata slike koji odgovaraju paralelnim bridovima scene i određivanje smjera bridova originala u trodimenzionalnom prostoru. Preduvjet za korištenje te metode je postupak pretprocesiranja kojim se određuje težinski pravocrtni prikaz scene, a opisan je u prethodnom poglavlju.

5.1 Pronalaženje nedogleda

U velikom broju primjena umjetne inteligencije, prvi korak prema trodimenzionalnom tumačenju scene na temelju slike dobivene perspektivnom projekcijom jest određivanje nedogleda [39]. Budući da robot “zna” kakvom je kamerom opremljen, poznata je matrica K sa unutarnjim parametrima kamere (3.4), a onda i preslikavanje iz koordinatnog sustava kamere u koordinatni sustav slike te njegov inverz. Tako se na temelju položaja nedogleda pravaca slike, može odrediti trodimenzionalni smjer pružanja odgovarajućih bridova scene u koordinatnom sustavu kamere [22]. U literaturi su opisane brojne metode pronalaženja točaka nedogleda u slici na temelju njenog pravocrtnog prikaza. [8, 22, 20, 29, 37, 39, 45, 94]. Ti algoritmi su interesantni i zato što se mogu primijeniti na pronalaženje tzv. fokusa ekspanzije u polju optičkog toka.

Određivanje 3D strukture scene na temelju nedogleda u slici je posebno prikladno u okolišima sa velikim brojem paralelnih bridova koji imaju relativno mali broj smjerova. Primjer takvog okoliša su scene zatvorenog prostora [20], ali i urbani eksterijeri [39] te scene vezane uz automatsko upravljanje motornim vozilima po gradskim [56] i otvorenim [8] prometnicama. Dobivena informacija o smjerovima pružanja bridova u sceni može bitno pojednostavniti tumačenje scene [29, 30, 34, 55, 56].

5.1.1 Pregled postojećih algoritama

Jedan od najutjecajnijih članaka na području analize perspektivnih slika odnosno pronalaznja nedogleda napisao je Barnard 1983. godine. Taj postupak je opisan u [94], kako slijedi. Na samom početku postupka se za svaki pravocrtni segment slike pronalazi interpretacijska ravnina — skup svih pravaca 3D prostora koji se preslikavaju u pravac slike na kojem se nalazi zadani segment (odjeljak 3.3). Skup svih jediničnih vektora smjerova pravaca 3D prostora koji se nalaze na interpretacijskoj ravnini čini kružnicu koja odgovara presjecištu interpretacijske ravnine sa jediničnom sferom čije središte se podudara sa središtem projekcije (tzv. Gaussovom sferom). Temeljna ideja postupka je u ovom trenutku primijeniti HT tako da se Gaussova sfera iskoristi kao parametarski prostor. Gaussova sfera se dijeli na područja, svakom području se dodjeljuje akumulator i provodi se glasački proces u kojem svaki pravocrtni segment slike doprinosi akumulatorima kroz koje prolazi njegova interpretacijska ravnina. Lokalni maksimumi u parametarskom prostoru, tada predstavljaju smjerove scene koji mogu biti originali najvećem broju pravocrtnih segmenata slike. U tom postupku HT je klasično preslikavanje (1:m) u kojem svaka značajka (pravocrtni segment) doprinosi većem broju akumulatora. Ključna pretpostavka koja se u kontekstu određivanja nedogleda mora napraviti, jest da će traženi smjerovi paralelnih bridova 3D prostora odgovarati najvećim grupama konvergentnih pravocrtnih segmenata jer će takve grupe uvjetovati lokalne maksimume u polju akumulatora. Takva pretpostavka vrijedi u scenama sa velikim brojem paralelnih ravnih bridova pa se, nakon obavljene transformacije nad slikama takvih scena, mogu postulirati nedogledi na temelju lokalnih maksimuma u polju akumulatora.

Ubrzo nakon toga, Magee i Aggarwal predlažu drukčiji pristup, u kojem parovi pravocrtnih segmenata za koje se može pretpostaviti da odgovaraju paralelnim bridovima scene, doprinose akumulatoru koji sadrži projekciju njihovog hipotetskog nedogleda na Gaussovu sferu. U tom pristupu, HT je preslikavanje oblika (2:1) u kojem parovi značajki doprinose točno jednom akumulatoru. Taj postupak je opisan u [37], kako slijedi. Na početku, za svaki pravocrtni segment definiran krajnim točkama $A(x_A, y_A)$ i $B(x_B, y_B)$, određuje se interpretacijska ravnina predstavljena vektorom normale po izravnoj formuli

$$\mathbf{n}_{\mathbf{AB}} = \begin{pmatrix} x_A \\ y_A \\ f \end{pmatrix} \times \begin{pmatrix} x_B \\ y_B \\ f \end{pmatrix}. \quad (5.1)$$

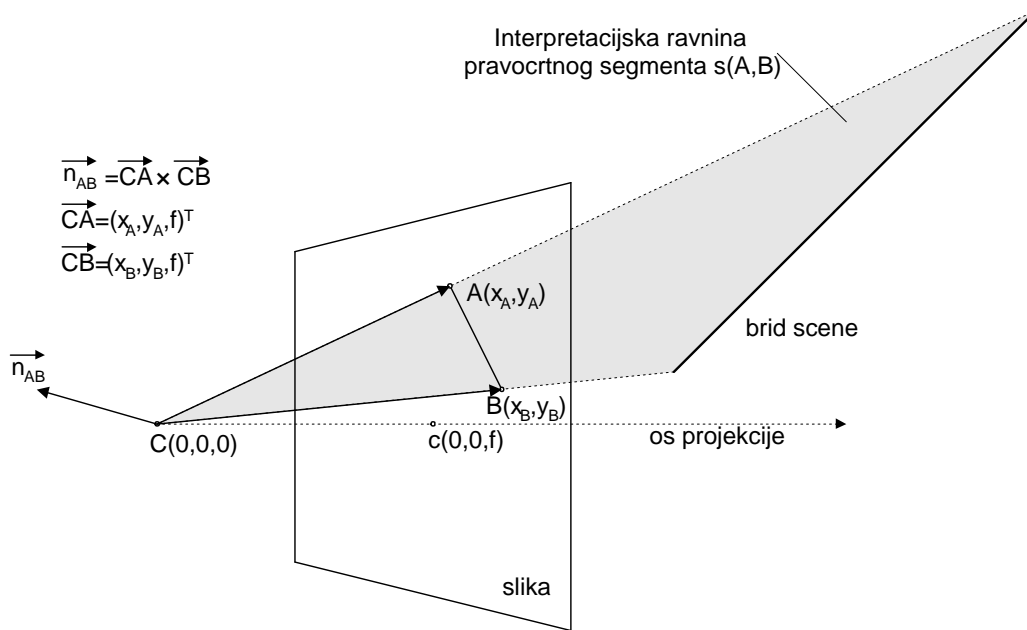
Jednadžba (5.1) je ilustrirana na sl. 5.1.

Hipotetski nedogled para pravocrtnih segmenata (s_1, s_2) odgovara vektoru \mathbf{N}_{12} koji se dobiva kao presjecište njihovih interpretacijskih ravnina. Vektor \mathbf{N}_{12} se može izračunati kao vektorski produkt normala dvaju ravnina \mathbf{n}_1 i \mathbf{n}_2 :

$$\mathbf{N}_{12} = \mathbf{n}_1 \times \mathbf{n}_2. \quad (5.2)$$

Jednadžba (5.2) je ilustrirana na sl. 5.2.

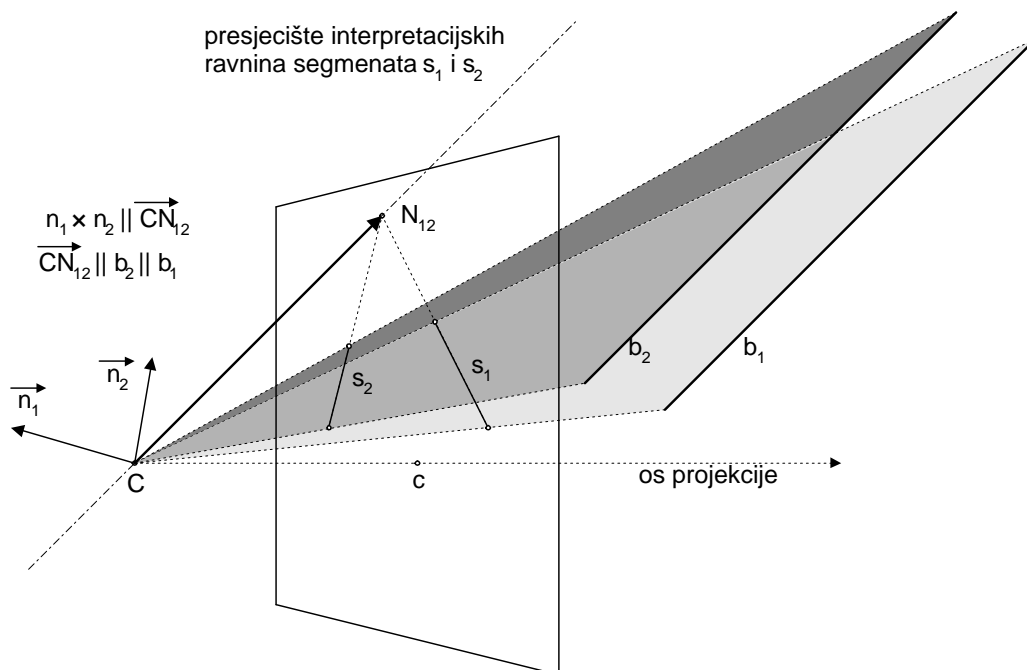
Postupak se može pojednostavniti ako se prihvati logično ograničenje da stvarni nedogledi mogu biti samo oni vektori \mathbf{N}_{12} koji nisu usmjereni prema unutrašnjosti nekog od segmenata koji ga definiraju. Prilikom transformacije, svaki hipotetski nedogled glasa za akumulator koji “pokriva” njegovu projekciju na Gaussovoj sferi, a nedogledi u slici se postuliraju na temelju lokalnih maksimuma u parametarskom prostoru pod istim pretpostavkama kao i u Barnardovoj metodi. Budući da je vektorski produkt jedina operacija kojom se određuje hipotetski nedogled dvaju pravocrtnih segmenata slike, postupak je nazvan *metodom vektorskog produkta* (engl. cross product method). Ovim formalizmom, nedogledi se predstavljaju smjerom u 3D prostoru što je analogno opisu točaka projekcijskom geometrijom. Nije stoga čudno da su jednadžbe (5.1) i (5.2) ekvivalentne jednadžbama (3.13) i (3.14).



Slika 5.1: Određivanje normale interpretacijske ravnine pravocrtnog segmenta \overline{AB} na temelju koordinata njegovih krajnjih točaka, po jednadžbi (5.1).

Liou i Jain [8] predlažu postupak za upravljanje motornim vozilom koji se temelji na algoritmu za pronalaženje nedogleda uzdužnih granica ceste. Pod pretpostavkom da je lokalni radijus zakrivljenosti ceste oko uspravne osi koordinatnog sustava svijeta velik, dobiveni nedogled odgovara lokalnom smjeru pružanja ceste što je najvažnija informacija za uspješno upravljanje vozilom. Autori su nadalje pretpostavili da je lokalna zakrivljenost ceste oko vodoravnih osi koordinatnog sustava svijeta zanemariva. Posljedica te pretpostavke je da je položaj idealnog pravca ravnina paralelnih sa ravninom ceste u slici poznat. Taj pravac se sastoji od svih nedogleda pravaca koji su paralelni s ravninom ceste pa tako sadrži i traženi nedogled uzdužnih granica ceste. Ako se nadalje uvaži da se nedogled uzdužnih granica ceste pri normalnim uvjetima vožnje mora nalaziti unutar slike, prostor pretraživanja se smanjuje sa cijele Gaussove sfere na samo jedan pravocrtni segment slike. Autori su taj pravocrtni segment razdijelili na kratke intervale kojima su dodijelili nezavisne procesne elemente koji obradu mogu izvoditi paralelno. Za svaki interval, odgovarajući procesni element razmatra sve pravocrtne segmente koji se iz središta intervala pružaju prema donjem rubu slike i to u dvije kategorije, za određivanje lijeve i desne uzdužne granice ceste. Pri tome se računa mjera podudaranja svakog razmatranog segmenta sa granicom ceste na temelju broja, povezanosti i apsolutnog iznosa gradijenta rubnih elemenata koji se na tom pravocrtnom segmentu nalaze. Konačno, svakom intervalu se pridružuje težina jednaka manjoj maksimalnoj mjeri podudaranja svih pravocrtnih segmenata u obje kategorije, a interval s najvećom težinom se proglašava nedogledom koji definira smjer pružanja ceste.

Velik broj autora predlagao je različite modifikacije osnovnih ideja koje su izložene u radovima Barnarda te Mageeja i Aggarwala. Tako Quan i Mohr [39] predlažu hijerarhijsku Houghovu transformaciju kojom se smanjuje računaska složenost postupka. Brillault i O'Mahony [20] predlažu alternativni izotropni akumulatorski prostor u kojem je nesigurnost određenih nedogleda konstantna za sve akumulatore. Shakunaga [29] predlaže specijalne funkcije kojima se za svaki hipotetski nedogled računaju doprinosi susjedstvu akumulatora te određivanje glavnih smjerova ortogonalne scene optimizacijskim postupkom. Lutton, Maître i Lopez-Krahe [39] također koriste izotropni parametarski prostor i uzimaju u obzir pogreške pri izlučivanju pravocrtnih segmenata prilikom određivanja triju



Slika 5.2: Pravocrtni segmenti s_1 i s_2 odgovaraju paralelnim bridovima scene b_1 i b_2 . Slika ilustrira postupak pronalaženja nedogleda pravocrtnih segmenata na temelju normala odgovarajućih interpretacijskih ravnina, u skladu sa jednadžbom (5.2).

smjerova strukture promatranih ortogonalnih scena.

Schuster et al. [37] opisuju primjenu modificirane metode vektorskog produkta za određivanje smjerova pružanja okoliša, na temelju nedogleda granica građevnih elemenata stropa. Njihova modifikacija se sastoji od slijedećih elemenata:

1. uzimaju se u obzir samo pravocrtni segmenti koji su dulji od apriorno zadane duljine l_{min} ;
2. polju akumulatora doprinose samo parovi segmenata koji nisu približno paralelni, nisu relativno blizu jedan drugoga i čiji hipotetski nedogledi ne leže u granicama pribavljene slike;
3. postupak traženja lokalnih maksimuma u parametarskom prostoru se obavlja grupiranjem elemenata parametarskog prostora čiji akumulatori su veći od nule.

McLean i Kotturi [45] navode slijedeće nedostatke klasičnih postupaka za određivanje nedogleda Houghovom transformacijom na Gaussovoj sferi:

1. metode određivanja nedogleda se temelje na položajima krajnjih točaka pravocrtnih segmenata dok u konačni proračun preko interpretacijske ravnine ipak ulazi jednadžba pravca;
2. poteškoće u pronalaženju lokalnih maksimuma u parametarskom prostoru u praksi vode na postupak koji zahtijeva apriorno znanje o približnom položaju nedogleda u slici;
3. preciznost dobivenih rezultata je ograničena rezolucijom parametarskog prostora i može ne postići preciznost koju dozvoljavaju ulazni podaci;
4. svim pravocrtnim segmentima slike se pridaje jednaka važnost.

Autori predlažu pristup u kojem se kao temeljne značajke koriste jednadžbe pravaca na kojima se pravocrtni segmenti slike nalaze, a ne njihove krajnje točke. Izravno izdvajanje jednadžbi pravocrtnih segmenata u slici se postiže originalnim postupkom pretprocesiranja kojim se u jednom koraku pronalaze izdužene povezane nakupine slikovnih elemenata sa sličnim smjerom gradijenta. Nedogledi se pronalaze eksplicitnim grupiranjem izdvojenih pravaca oko glavnih 2D smjerova gradijenta slike. Nedostatak takvog postupka grupiranja je što se ne može primijeniti za nedogled čiji položaj se nalazi unutar slike. Predložena je i metoda modeliranja nesigurnosti rezultata postupka koja se temelji na rezultatima pretprocesiranja (odstupanje slikovnih elemenata nakupine od interpoliranog pravca) i grupiranja (udaljenost pravaca od pripadajućeg nedogleda).

Shufelt [94] predlaže dvije tehnike za poboljšavanje klasičnog pristupa određivanju nedogleda. Tehnike su razmatrane u kontekstu slika sa slabim perspektivnim izobličenjima i kratkim pravocrtnim segmentima uz prisustvo šuma. Prva tehnika vodi postupak traženja nedogleda u parametarskom prostoru na temelju apriornog znanja o geometriji promatranih objekata. Radi se o poopćenju ograničenja koja su za ortogonalne scene uvedena u prethodnim radovima, npr. u [29]. Druga tehnika se odnosi na modeliranje nesigurnosti kratkih pravocrtnih segmenata uzrokovane diskretizacijom.

5.1.2 Idealizirane scene hodnika

Svojstvo stvarnih hodnika je da im je duljina bitno veća od visine i širine. Zato definiramo idealizirane scene hodnika kao podskup scena s ortogonalnom triedarskom strukturom u kojima su uzdužni bridovi znatno duži i brojniji od poprečnih. Za većinu smjerova gledanja, pravocrtni segmenti slike koji odgovaraju uzdužnim bridovima takve scene će biti znatno brojniji od segmenata koji odgovaraju poprečnim bridovima. Zbog toga se nedogled uzdužnih bridova takve scene može odrediti kao dominantni nedogled slike, odnosno nedogled definiran s najvećim brojem pravocrtnih segmenata.

U većini okoliša koji se nalaze na Zemlji, smjer pružanja lokalnog polja gravitacije se poklapa sa jednom osi strukture okoliša koja se onda naziva uspravnom osi. U takvim okolišima vodoravne ravnine se mogu definirati kao ravnine čije su normale paralelne sa uspravnom osi. Tako će kod većine ortogonalnih scena, jedna os strukture biti uspravna dok će druge dvije pripadati vodoravnoj ravnini.

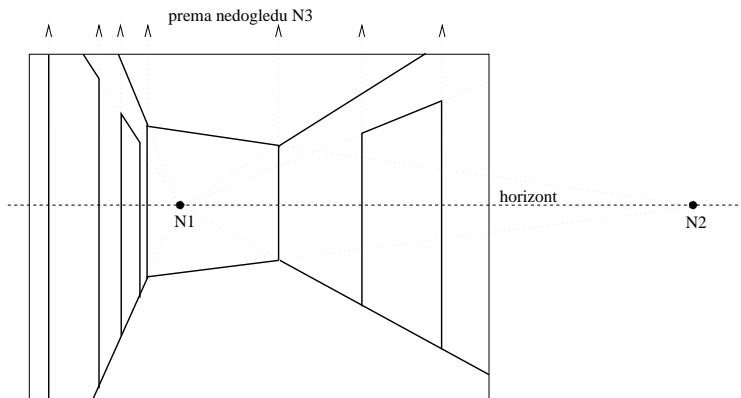
Položaji nedogleda uzdužnih i poprečnih bridova ortogonalne scene su međusobno ovisni. Pokazuje se da nedogledi svih vodoravnih bridova scene čine pravac u kojeg se preslikava horizont vodoravnih ravnina scene. Taj pravac se često naziva horizontom, iako je on zapravo samo slika horizonta u sceni. Ta će se konvencija koristiti i ovdje, u situacijama u kojim je iz konteksta jasno radi li se o pravcu projekcijskog prostora ili projekcijske ravnine. U konfiguraciji koja je opisana na sl. 1.1, 1.2 i 3.1, koordinatni sustav kamere ima dva rotacijska stupnja slobode u odnosu na koordinatni sustav robota čija se uspravna os podudara sa uspravnom osi koordinatnog sustava svijeta. U takvoj konfiguraciji, slijedeći podaci su robotu poznati jer su određeni nagibom kamere:

- položaj nedogleda uspravnih bridova scene;
- položaj horizonta u slici.

Eksperimenti su pokazali da je pogodno održavati nagib kamere unutar $\pm 10^\circ$ kako bi se uspravni bridovi preslikavali u približno uspravne segmente slike, a horizont bio u vidnom polju kamere. Sl. 5.3 prikazuje tipičnu sliku idealizirane scene hodnika dobivenu u skladu sa prethodnim razmatranjem.

5.1.3 Odabir pristupa pronalaženju nedogleda

U literaturi se mogu pronaći dva temeljna pristupa određivanju nedogleda:



Slika 5.3: Pravocrtni prikaz idealizirane scene hodnika sa označenim horizontom i nedogledima koji odgovaraju uzdužnoj (N1), poprečnoj (N2) i uspravnoj (N3) osi hodnika.

1. analiza parametarskog prostora dobivenog primjenom Houghove transformacije;
2. eksplicitno grupiranje hipotetskih nedogleda.

U većini objavljenih radova, korišten je pristup koji se temelji na Houghovoj transformaciji (HT). Postoje dvije temeljne varijacije tog pristupa koje se razlikuju u načinu popunjavanja polja akumulatora parametarskog prostora [39]:

- (a) “Jedan-na-mnogo” (1:m) transformacija, kojom svaki razmatrani pravocrtni segment doprinosi većem broju akumulatora u parametarskom prostoru. Takav pristup korišten je u radovima koji se temelje na Barnardovom članku iz 1983. godine.
- (b) “Mnogo-na-jedan” (m:1) transformacija, kojom više pravocrtnih segmenata (tipično $m=2$) doprinosi točno jednom akumulatoru u parametarskom prostoru. Takav pristup korišten je u radovima koji se temelje na radu Mageeja i Aggarwala iz 1984. godine.

Neka je broj izlučenih pravocrtnih segmenata jednak N i neka je, kod pristupa (a), broj akumulatora parametarskog prostora kojima doprinosi pojedini pravocrtni segment slike jednak M . Tada složenost popunjavanja polja akumulatora kod pristupa (1: M) iznosi $O(N \cdot M)$, dok složenost iste operacije kod pristupa (2:1) iznosi $O(N^2)$. Budući da tipično vrijedi $N \approx 100$ i $M \approx 100$, metode imaju vrlo sličnu asimptotsku složenost za uobičajene N i M . Može se očekivati međutim da će pristup (1:m) biti prikladniji u slučajevima kad je $N \gg M$ i obratno, da će pristup (2:1) biti bolji za slučajeve $N \ll M$.

Drugi temeljni pristup određivanju nedogleda podrazumijeva njihovo eksplicitno grupiranje, bez korištenja akumulatora i preslikavanja u parametarski prostor. Na prvi pogled, takav pristup nije privlačan jer je prvo potrebno odrediti $N^2/2$ hipotetskih nedogleda, a onda se još suočiti sa njihovim grupiranjem. Grupiranje se u pravilu obavlja heurističkim postupcima čiju je složenost teško egzaktno utvrditi, ali je vrlo malo vjerojatno da bi neki od tih algoritama bio manje složen od traženja lokalnih maksimuma u parametarskom prostoru koji se dobiva uz pomoć HT. Međutim, eksplicitno grupiranje je posebno pogodno kada su prisutna ograničenja na broj i položaje točaka nedogleda, kao što je to slučaj u problemu koji se razmatra u ovom radu. Navedena ograničenja mogu smanjiti broj hipotetskih nedogleda koje treba grupirati više desetina puta pa složenost grupiranja može postati zanemariva. Korak grupiranja hipotetskih nedogleda je moguće obaviti ili na slikovnoj ravnini [45] ili nad projekcijama nedogleda na Gaussovoj sferi. Konačno je odabran pristup u kojem se kao prostor za obavljanje grupiranja koristi Gaussova sfera zbog mogućnosti definiranja ujednačenije mjere udaljenosti.

5.1.4 Pronalaženje dominantnog nedogleda vodoravnih bridova scene

Predloženi algoritam na ulazu zahtijeva težinski pravocrtni prikaz analizirane slike, dok na izlazu daje položaj dominantnog nedogleda pravocrtnih segmenata koji odgovaraju vodoravnim bridovima scene. Algoritam je u kraćem obliku već opisan u [92], a oblikovan je na temelju prethodno opisanih rješenja iz literature, uz uvažavanje specifičnosti razmatranog okoliša. Činjenica da je uspravni smjer strukture okoliša poznat značajno olakšava posao budući da se onda, slično kao i u problemu kojeg su rješavali Liou i Jain [8], može odrediti idealni pravac na kojem se traženi nedogled sigurno nalazi. Predloženi algoritam se može opisati po slijedećim koracima, u skladu sa sl. 5.4.

1. Provjera konzistencije pravocrtnih segmenata — odbacuju se segmenti sa velikom procijenjenom mjernom nesigurnošću te segmenti koji ne mogu odgovarati vodoravnom bridu scene.
2. Određivanje homogenih jednadžbi pravaca — za svaki probrani pravocrtni segment, određuje se homogena jednadžba pravca nositelja na temelju krajnjih točaka segmenta.
3. Određivanje hipotetskih nedogleda — za svaki par probranih pravocrtnih segmenata za koje se može pretpostaviti da odgovaraju paralelnim vodoravnim bridovima scene, određuje se hipotetski nedogled i njegova težina.
4. Grupiranje hipotetskih nedogleda u nakupine te određivanje dominantnog nedogleda kao težišta nakupine s najvećom težinom.

Slijedi opis izvedbenih detalja za svaki od opisanih koraka.



Slika 5.4: Prikaz toka obrade algoritma za određivanje dominantnog nedogleda vodoravnih bridova scene. Pravokutnici označavaju ulazne i izlazne strukture podataka te međurezultate, dok su koraci obrade predstavljani krugovima (opširnije u tekstu).

Provjera konzistencije pravocrtnih segmenata

Pri traženju nedogleda vodoravnih bridova scene, razmatraju se samo pravocrtni segmenti za koje pretpostavka da se u njih preslikao vodoravan brid scene nije kontradiktorna. Kamera ima samo dva rotacijska stupnja slobode koji joj ne dozvoljavaju zauzimanje položaja kojim bi se vodoravni bridovi scene preslikali u uspravne segmente slike. Na temelju načela projekcijske geometrije, može se pokazati da ni pravocrtni segmenti koji presijecaju horizont ne mogu biti slike vodoravnih bridova scene.

Kao što je velik broj istraživača naglasio, nesigurnost smjera pružanja kratkih pravocrtnih segmenata je zbog diskretizacije vrlo velika. Na sreću, razmatrani okoliš se u potpunosti sastoji od ravnih bridova pa se u ovom slučaju, analiza može ograničiti samo na relativno duge pravocrtne segmente (dulje od dvadesetine manje dimenzije analizirane slike). Stoga se iz daljnjeg razmatranja izbacuju uspravni, relativno kratki te pravocrtni segmenti koji presijecaju horizont.

Određivanje homogenih jednadžbi pravaca

U ovom koraku se određuju homogene jednadžbe ili, ekvivalentno, interpretacijske ravnine pravaca slike koji su definirani izlučenim pravocrtnim segmentima (sl. 5.1). Ponovo, prikaz u homogenim koordinatama je pogodan jer se time zaobilazi mogućnost pojave singulariteta u kasnijim proračunima. Pretpostavljen je opis pravocrtnih segmenata 2D koordinatama njihovih krajnjih točaka. Stoga se homogena jednadžba pravca (3.11) svakog izlučenog segmenta određuje u slijedeća dva koraka:

1. krajnje točke pravocrtnog segmenta se prevode u homogene koordinate u skladu s injekcijom (3.9);
2. određuje se jednadžba pravca u homogenim koordinatama koji prolazi kroz krajnje točke segmenta slike u skladu sa jednadžbom (3.13).

Opisana metoda određivanja homogene jednadžbe pravca na temelju krajnjih točaka pravocrtnog segmenta je ekvivalentna određivanju interpretacijske ravnine pravca slike (sl. 5.1).

Određivanje hipotetskih nedogleda

Hipotetski nedogled za svaki par pravaca dobivenih u prethodnom koraku obrade se određuje u skladu s jednadžbom (3.14). Ako je položaj tako dobivenog nedogleda unutar jednog od dvaju pravocrtnih segmenata koji definiraju pravce, on se odbacuje jer je time dokazano da odgovarajući bridovi scene nisu paralelni. Nadalje, hipotetski nedogled se također odbacuje ako je značajno udaljen od horizonta, jer tada odgovarajući bridovi scene ne mogu biti i vodoravni i paralelni. Svakom preostalom elementu skupa hipotetskih nedogleda pridružuje se težina u iznosu umnoška težina pravocrtnih segmenata koji ga definiraju. Ako je broj elemenata tog skupa manji od tri, dodaju mu se presjecišta horizonta i pravaca na kojima leže probrani pravocrtni segmenti s velikom težinom. Težina takvih hipotetskih nedogleda jednaka je težini segmenta.

Iako se koristi drukčiji matematički formalizam, opisani postupak određivanja hipotetskog nedogleda dvaju pravaca slike je ekvivalentan pronalaženju presjecišta njihovih interpretacijskih ravnina (sl. 5.2).

Određivanje dominantnog nedogleda

Skup hipotetskih nedogleda se tek sada projicira na Gaussovu sferu kako bi se mogla definirati ujednačena mjera udaljenosti. To se postiže normiranjem vektora nedogleda na jediničnu apsolutnu vrijednost. Nad skupom se definira mjera udaljenosti kao kvadrirana L2 metrika u 3D Euklidskom prostoru. Skup nedogleda se podvrgava postupku nenadgledanog grupiranja modifikacijom jednostavnog heurističkog “maximin” algoritma [1], koja je definirana kako slijedi:

1. Kao klica prve nakupine, odabire se hipotetski nedogled sa najvećom težinom i briše se iz liste nedogleda.
2. Kao klica druge nakupine, odabire se hipotetski nedogled *drugi*, za kojeg je udaljenost do klice prve nakupine maksimalna. Ako sa $D_{i,j}$ označimo udaljenost nedogleda i do klice nakupine j , onda mora vrijediti:

$$D_{drugi,1} = \max_i D_{i,1}.$$

Neka je *prethodna_udaljenost* := $D_{drugi,1}$.
Nedogled *drugi* se briše se iz liste nedogleda.

3. Za svaki nedogled i iz liste nedogleda, pronalazi se udaljenost do najbliže klice nakupine j_i , tako da vrijedi:

$$D_{i,j_i} = \min_j D_{i,j}.$$

4. Odabire se nedogled n iz liste nedogleda, koji je najviše udaljen od sebi najbliže klice nakupine. Za taj nedogled mora vrijediti:

$$D_{n,j_n} = \max_i D_{i,j_i},$$

odnosno, kad se uvrsti izraz za D_{i,j_i}

$$D_{n,j_n} = \max_i \min_j D_{i,j}.$$

5. **Ako** vrijedi:

$$D_{n,j_n} < \frac{2}{5} \cdot \text{prethodna_udaljenost},$$

Onda

Nedogled n se iz liste nedogleda prebacuje u listu klica nakupina

Postavlja se $\text{prethodna_udaljenost} := D_{n,j_n}$.

Obrada se vraća na korak 3.

Inače

Obrada se nastavlja u slijedećem koraku.

6. Preostali nedogledi iz liste nedogleda se pridružuju najbližim klicama nakupina. Središta dobivenih nakupina se određuju kao težišta pripadnih nedogleda, na temelju njihovih položaja i težina.

Rezultat tog algoritma je razdjeljivanje hipotetskih nedogleda na veći broj grupa, a naknadno se dotjeruje stapanjem nakupina koje su na Gaussovoj sferi udaljene manje od 1° . Svakoj pojedinačnoj nakupini se pridružuje težina u iznosu zbroja težina hipotetskih nedogleda koji nakupini pripadaju. Konačno, dominantni nedogled se određuje kao težište nakupine s najvećom težinom.

Opisani postupak određivanja dominantnog nedogleda u slici se zasniva na pretpostavci da su nakupine hipotetskih nedogleda koji odgovaraju paralelnim bridovima scene brojnije te da imaju veću težinu od nakupina koje su posljedica slučaja. Primjenljivost postupka je zato ograničena na scene s velikim brojem paralelnih bridova. Međutim, čak i u scenama koje zadovoljavaju taj zahtjev, postoji mogućnost da vidljivi dio scene ne sadrži paralelne bridove. Takva situacija se u razmatranom okolišu javlja kad je orijentacija kamere okomita na smjer pružanja hodnika. U takvim slučajevima, postupak ne može pronaći traženu informaciju, jer temeljna pretpostavka za njegov ispravan rad nije zadovoljena.

5.1.5 Primjenljivost predloženog postupka za lokalizaciju

U ovom pododjeljku se razmatra mogućnost primjene opisanog postupka za pronalaženje dominantnog nedogleda vodoravnih bridova za određivanje smjera pružanja hodnika na temelju jedne slike. Temeljna pretpostavka za uspješno odvijanje postupka je međutim da scena sadrži veći broj vodoravnih paralelnih bridova. Empirijski je utvrđeno da je ta pretpostavka zadovoljena samo u slučajevima kada je robot već orijentiran u smjeru pružanja hodnika, s dozvoljenim kutnim odstupanjem od oko pola širine vidnog polja korištene kamere. U ovom pododjeljku se, na temelju empirijskih ispitivanja, gradi model ponašanja postupka za scene u kojima ta pretpostavka nije zadovoljena

Pristup lokalizaciji na temelju izravne primjene predloženog postupka bi podrazumijevao da je približna orijentacija robota u hodniku već poznata. To je protivno ciljevima

koji su zacrtani u poglavlju 1 — tražena procedura lokalizacije bi se trebala primjenjivati baš u tim, iznimnim slučajevima, kada su svi drugi izvori znanja o okolišu nedostupni. Međutim, ostvarivanje željene procedure na temelju samo jedne slike je vrlo težak zadatak. Čak i čovjek, nakon ulaska u nepoznati hodnik mora stati i osvrnuti se oko sebe kako bi zaključio gdje je, donio odluku kuda želi ići i počeo razmatrati način dolaska do cilja. Zato ima smisla lokalizacijski zadatak rješavati aktivnim pristupom [11], analizirajući niz slika koje se pribavljaju za veći broj smjerova gledanja kamere.

Da bi se mogla oblikovati procedura lokalizacije koja bi radila na nizu slika, potrebno je izgraditi model postupka na temelju kojeg se može procijeniti pouzdanost dobivenih rezultata. Traženi model u ovom slučaju može biti algoritam koji na temelju analize rezultata postupka daje mjeru njihove konzistencije. Predlaže se vrlo jednostavan pristup izvedbi tog algoritma, u kojem se zaključak o konzistenciji rezultata postupka donosi samo na temelju jednog parametra — broja pravocrtnih segmenata koji podržavaju dominantni nedogled. Prvi korak u modeliranju postupka je promatranje kretanja tog parametra, za slike svih mogućih scena na koje bi postupak mogao biti primijenjen u kontekstu lokalizacije u hodniku. Traže se zato odgovori na slijedeća pitanja:

1. Da li je vrijednost odabranog parametra proporcionalna preciznosti rezultata postupka?
2. Ako je odgovor na prethodno pitanje potvrđan, u kojim okvirima ta vrijednost mora biti da bi se rezultati postupka mogli proglasiti pouzdanim?

Odgovori na ta pitanja će biti dani na temelju eksperimenata u slijedećem odjeljku.

5.1.6 Eksperimentalni rezultati

Opisani postupak pronalaženja nedogleda je isproban na velikom broju slika hodnika dobivenih VHS kamerom. Obavljene su tri grupe eksperimenata. Cilj eksperimenata prve grupe je provjeriti rad algoritma za slike za koje je i projektiran tj. za slike koje prikazuju veći broj paralelnih vodoravnih bridova. Eksperimenti iz druge i treće grupe su međutim provedeni u cilju pronalaženja odgovora na pitanja iz pododjeljka 5.1.5. Tako je svrha eksperimenata druge grupe dobivanje empirijske podloge za modeliranje ponašanja postupka u hodniku kod slika dobivenih za razne smjerove zakreta kamere. Konačno, cilj eksperimenata treće grupe je ocijeniti mogućnost pojavljivanja “nedogleda” u generiranom skupu pseudo-slučajnih pravocrtnih segmenata. Ta ocjena će dati i mjeru vrijednosti čitavog pristupa lokalizaciji: ako su nedogledi značajke koje se vrlo često javljaju i na slučajno generiranom skupu pravocrtnih segmenata, tada njihovo pronalaženje i nema neku upotrebnu vrijednost. Suprotno, ako se nedogledi ne javljaju u slučajnom skupu segmenata, onda su oni intrinzično svojstvo scene i mogu poslužiti kao temelj za donošenje daljnjih zaključaka.

Eksperimenti su obavljani prije nabavljanja kamere sa mogućnošću postavljanja kutova zakreta i nagiba. Stoga su sve analizirane slike pribavljene uz pomoć VHS kamere čiji su unutrašnji parametri (vodoravni i uspravni kut vidnog polja kamere — matrica \mathbf{K} , odjeljak 3.1) te kut nagiba u odnosu na vodoravnu ravninu određeni eksperimentalnim putem, prije obavljanja eksperimenata. Kamera je relativno starije proizvodnje pa su utjecaji radijalnog izobličenja leće značajni. Takva izobličenja otežavaju obradu jer se ravni bridovi scene preslikavaju u više ili manje zakrivljene segmente slike. Popravljanje tih izobličenja [54] međutim nije razmatrano zbog računске složenosti postupka. Zbog brzine postupka, odabrana je rezolucija slike $320 \times 240 \times 24$ bita koja je manja od maksimalne rezolucije koju podržava korišteni digitalizator.

Pokazalo se da je eksplicitno grupiranje hipotetskih nedogleda prikladan pristup određivanja nedogleda vodoravnih bridova u sustavu hodnika. Za većinu parova izlučenih pravocrtnih segmenata, pretpostavka da oni odgovaraju vodoravnim i paralelnim bridovima

scene je kontradiktorna sa njihovim međusobnim položajem u slikovnoj ravnini. Stoga su postignuta vremena izvršavanja koja bi u vrlo skoroj budućnosti omogućila primjenu postupka u stvarnom vremenu. Na računalu sa cjelobrojnou performansom od 4 SPECint95, opisani postupak traženja dominantnog nedogleda vodoravnih bridova scene na temelju težinskog pravocrtnog prikaza scene traje oko 0,5s.

Rezultati za uzdužne smjerove gledanja

U ovoj grupi eksperimenata, analizirane slike su pribavljene za orijentacije kamere koje se od uzdužne osi hodnika ne razlikuju više od pola širine vidnog polja kamere. Za takve smjerove gledanja, nedogled uzdužnih bridova hodnika se nalazi unutar slike. Tipična slika dobivena u skladu sa spomenutim ograničenjima se nalazi u lijevom dijelu sl. 5.5, dok su kao rezultati obrade, u njenom desnom dijelu, pokazane slijedeće značajke:

- križići koji označavaju hipotetske nedoglede;
- kvadratići koji pokazuju težišta nakupina;
- skup pravocrtnih segmenata koji određuju dominantnu nakupinu (oni su ucrtani i u izvornoj slici);
- vodoravni pravac koji označava horizont¹.



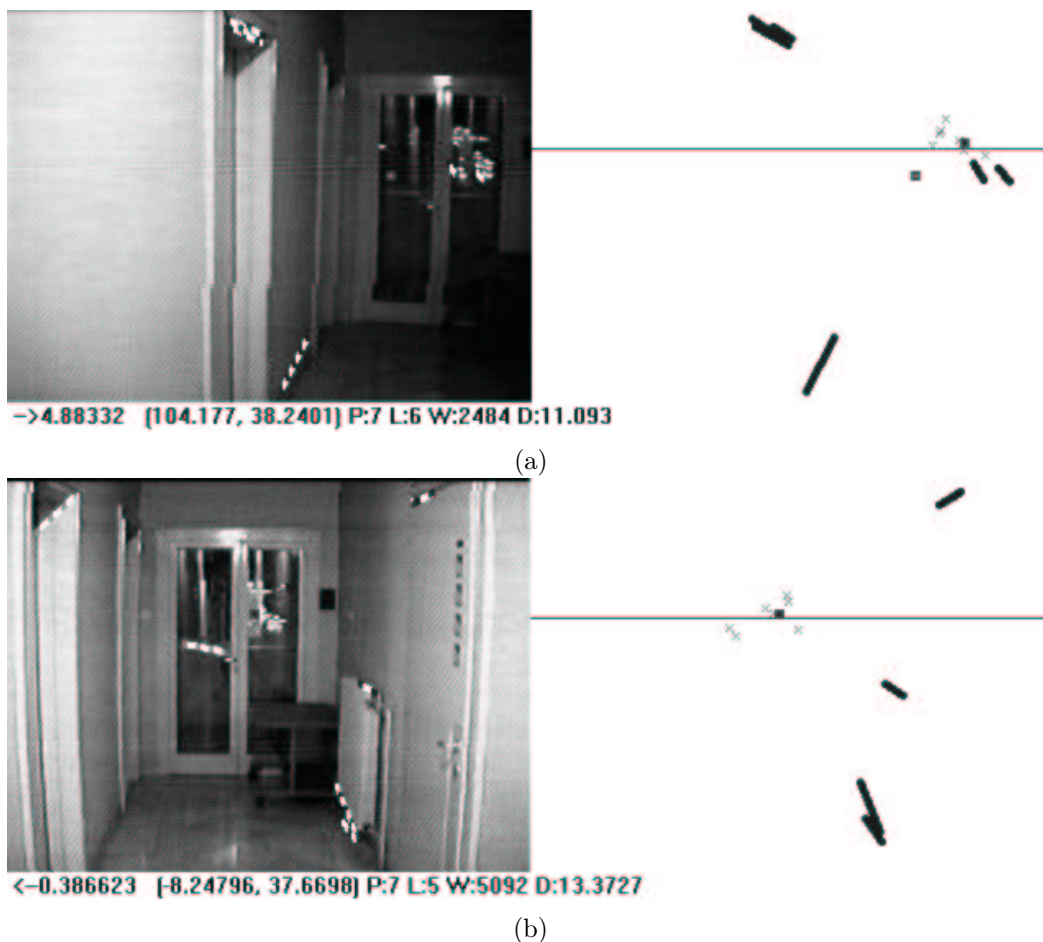
Slika 5.5: Slika hodnika (lijevo), te rezultati obrade (desno). U desnom dijelu prikazani su hipotetski nedogledi vodoravnih bridova scene (križići), njihove nakupine određene postupkom grupiranja (kvadratići), pravocrtni segmenti koji definiraju dominantnu nakupinu (dužine) te slika horizonta (vodoravna dužina).

Brojevi ispod slike označavaju kut između smjera gledanja kamere i procijenjenog smjera uzdužne osi hodnika te slijedeće podatke o dominantnoj nakupini:

- koordinate dominantne nakupine u slikovnoj ravnini (ishodište koordinatnog sustava je u središtu slike);
- broj pojedinačnih nedogleda koji čine dominantnu nakupinu;
- broj pravocrtnih segmenata koji određuju dominantnu nakupinu;
- prosječna međusobna udaljenost elemenata dominantne nakupine.

Rezultati obrade dviju slika koje su pribavljene iz istog položaja hodnika, ali za različite smjerove gledanja su pokazani na sl. 5.6.

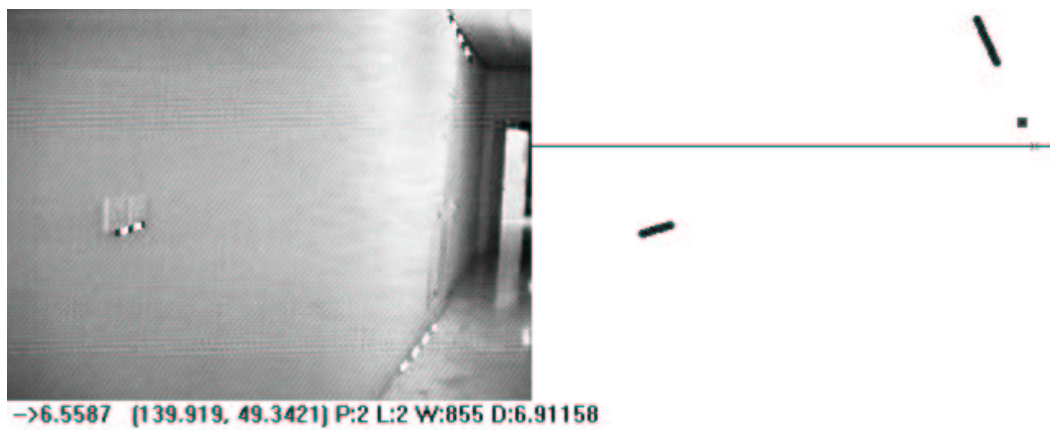
¹Položaj horizonta sustav određuje na temelju unesene ocjene nagiba VHS kamere.



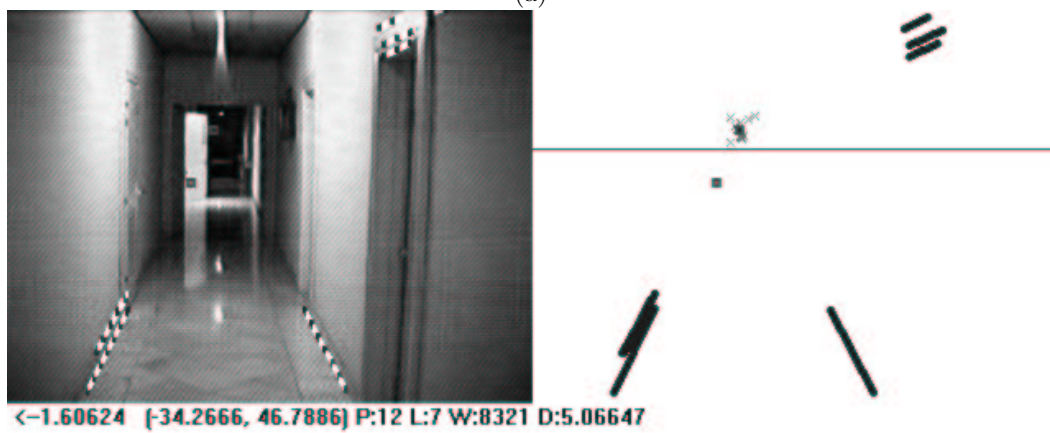
Slika 5.6: Rezultati obrade dviju slika koje su dobivene iz istog položaja hodnika kao i slika na sl. 5.5, ali za različite smjerove gledanja.

Dobiveni rezultati ukazuju kako je predloženom metodom moguće pouzdano određivanje uzdužne osi okoliša, ako je smjer gledanja kamere takav da se nedogled uzdužnih bridova scene nalazi unutar slike. Postupak je isproban na više lokacija hodnika, a neki od dobivenih rezultata su prikazani na sl. 5.7, sl. 5.8 i sl. 5.9.

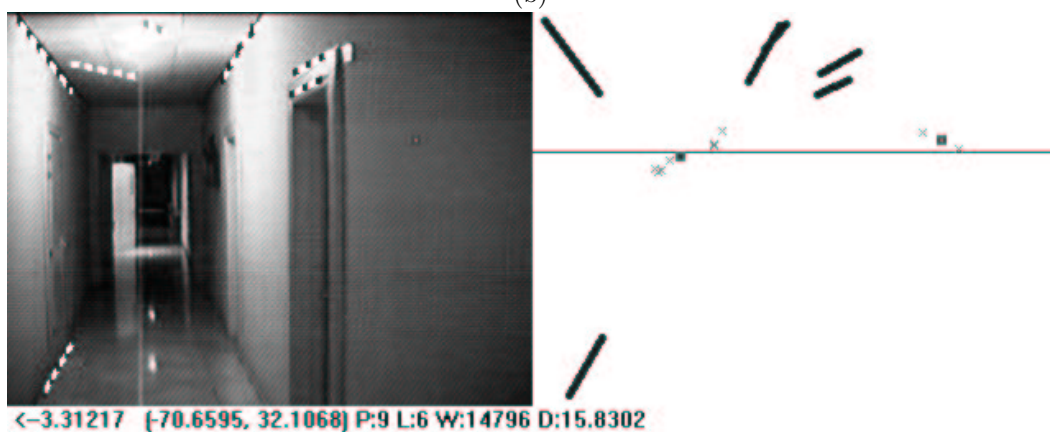
Vodoravni kut širine vidnog polja kamere je 15° pa se položaji dominantne nakupine i odgovarajućih hipotetskih nedogleda nalaze izvan granica slika na sl. 5.8(a) te sl. 5.9(a) i 5.9(c). Iz tog razloga, kvadratići i križići koji bi trebali označavati položaje tih hipotetskih nedogleda, ne mogu biti prikazani u granicama slike. Može se međutim zaključiti da je sustav ispravno pronašao smjer pružanja hodnika na temelju pravocrtnih segmenata koji definiraju dominantnu nakupinu te konačne ocjene smjera pružanja uzdužne osi hodnika (prvi u nizu brojeva ispod slika).



(a)

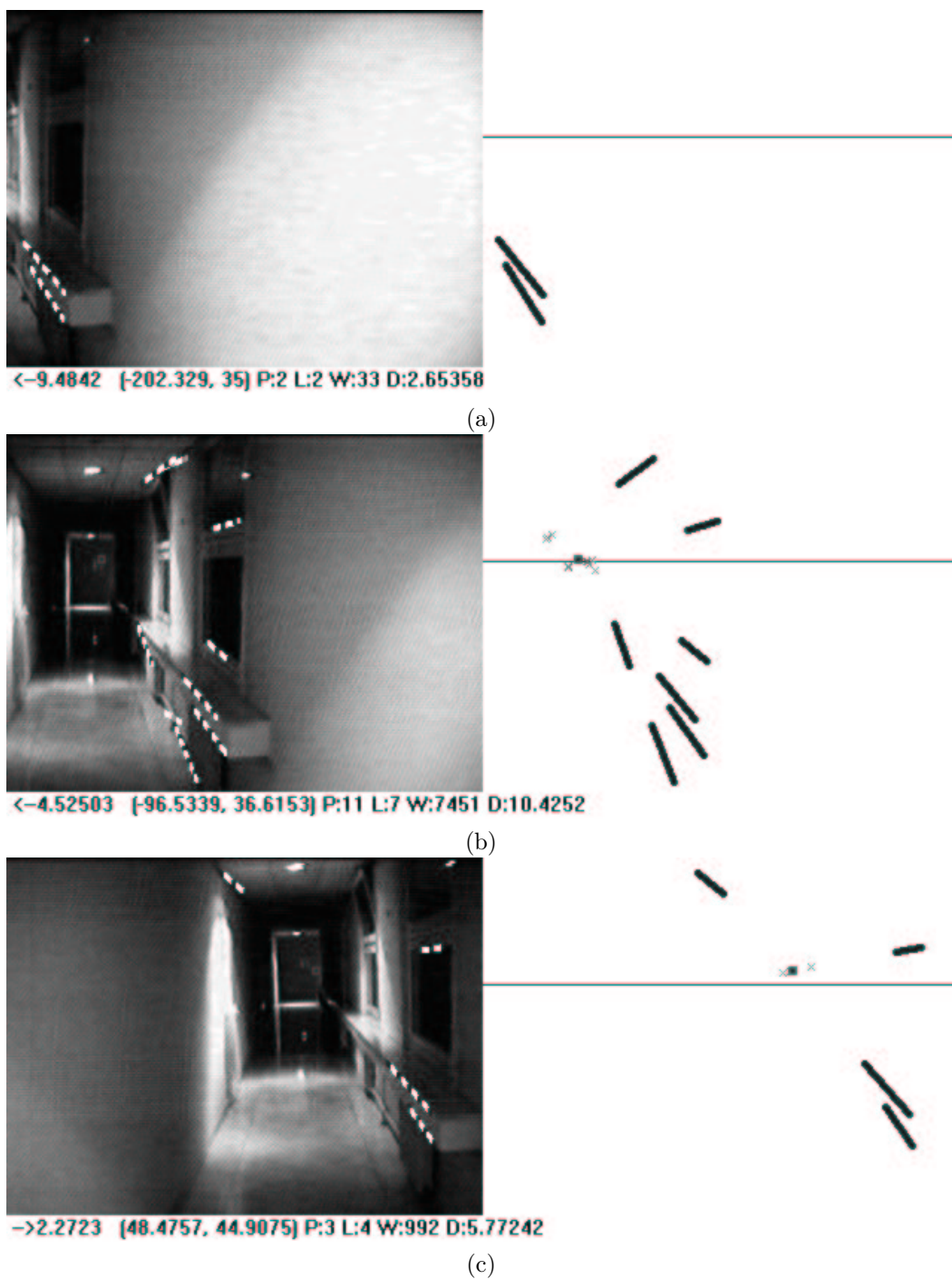


(b)

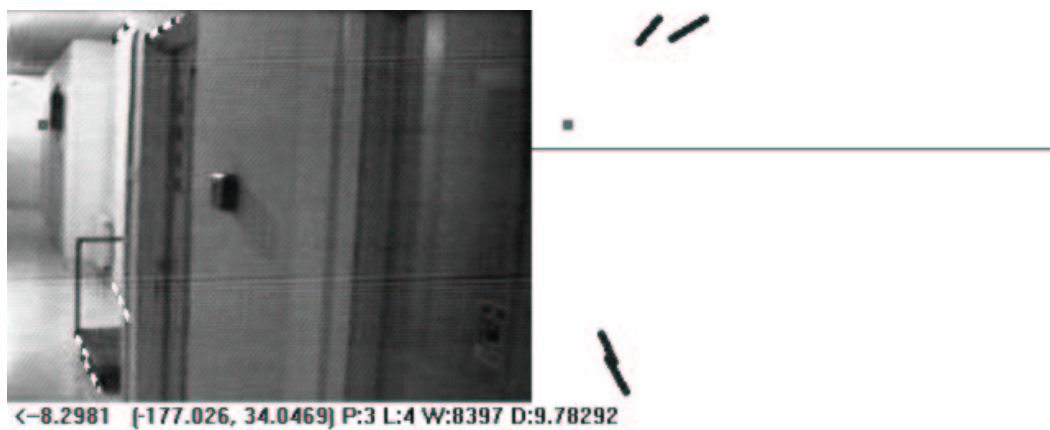


(c)

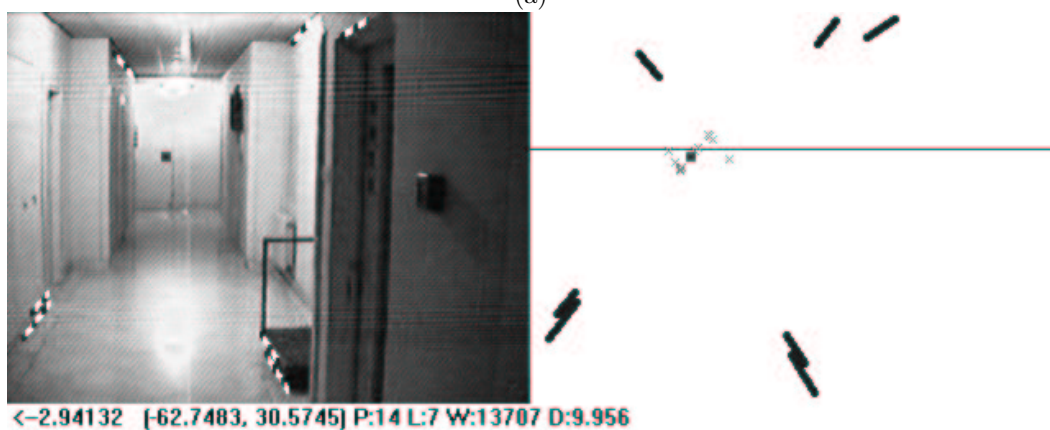
Slika 5.7: Rezultati obrade triju slika koje su dobivene iz istog položaja hodnika, za različite smjerove gledanja. Pokazane su iste značajke, kao i na sl. 5.5.



Slika 5.8: Rezultati obrade triju slika koje su dobivene iz istog položaja hodnika, za različite smjerove gledanja. Pokazane su iste značajke, kao i na sl. 5.5.



(a)



(b)



(c)

Slika 5.9: Rezultati obrade triju slika koje su dobivene iz istog položaja hodnika, za različite smjerove gledanja. Pokazane su iste značajke, kao i na sl. 5.9.

Rezultati pri raznim smjerovima gledanja

Svrha ovih eksperimenata je ispitivanje mogućnosti prepoznavanja slabe točnosti postupka za pronalaženje dominantnog nedogleda analizom broja pravocrtnih segmenata koji taj nedogled definiraju. U svakom od tri eksperimenta, postupak je primijenjen na nizu slika dobivenih za kutove zakreta kamere od 0° do 90° , na tri različita položaja u hodniku. Dobiveni rezultati su sažeti u tablici 5.1.

Tablica 5.1: Rezultati postupka za različite kutove zakreta kamere, za položaje 1, 2 i 3

γ_z	#	ukupan broj segmenata	broj segmenata dominantnog nedogleda	izračunati smjer pružanja hodnika	odstupanje od izmjerenog smjera
90°	1	0	0	-	-
	2	0	0	-	-
	3	0	0	-	-
60°	1	38	1	0°	60°
	2	0	0	-	-
	3	0	0	-	-
30°	1	22	1	20°	10°
	2	31	2	-15°	45°
	3	2	0	-	-
15°	1	4	1	10°	5°
	2	30	1	-5°	20°
	3	7	0	-	-
10°	1	29	3	10°	0°
	2	27	3	10°	0°
	3	60	5	10°	0°
5°	1	47	4	5°	0°
	2	62	13	5°	0°
	3	68	6	5°	0°
0°	1	68	9	0°	0°
	2	76	11	0°	0°
	3	78	11	0°	0°

Redci tablice sadrže rezultate eksperimenata za različite kutove zakreta kamere γ_z , za tri različita položaja kamere. Prvi stupac tablice sadrži izmjereni kut zakreta kamere u odnosu na uzdužnu os hodnika. Slijedeći stupac sadrži redni broj položaja na kojeg se odnose rezultati u redku. Treći stupac tablice sadrži ukupan broj pravocrtnih segmenata koji su izlučeni u fazi pretprocesiranja, dok se broj segmenata koji čine dominantni nedogled kao mjera pouzdanosti rezultata postupka nalazi u četvrtom stupcu. Posljednja dva stupca sadrže smjer pružanja hodnika koji je određen na temelju položaja dominantnog nedogleda te apsolutno odstupanje između izračunatog smjera pružanja hodnika (stupac 5) i izmjerenog kuta zakreta kamere (stupac 1).

Rezultati iz tablice 5.1 nisu iscrpni, dok je njihova procijenjena nesigurnost prilično visoka (oko 1°). Ipak, oni pokazuju kako jednostavna mjera pouzdanosti rezultata predloženog postupka ima izrazit maksimum u smjeru uzdužne osi hodnika, te da postupak daje precizne rezultate za smjerove gledanja koji su od uzdužne osi hodnika udaljeni manje od 10° . Bolji rezultati nisu mogli biti ni očekivani s obzirom da širina vidnog polja korištene kamere iznosi 15° . Dobiveni rezultati posebno naglašavaju važnost odabira pravog smjera gledanja u tumačenju scena, jer su za većinu smjerova gledanja, dobivene slike neupotrebljive u lokalizacijske svrhe (redci za 30° , 60° i 90°).

Rezultati na skupu slučajnih pravocrtnih segmenata

Ova grupa eksperimenata se sastoji u primjenjivanju predloženog postupka na skupu pseudo-slučajnih pravocrtnih segmenata. Napravljena su tri skupa eksperimenata, za različite brojnosti slučajnih segmenata. Svrha eksperimenata je ispitivanje ponašanja algoritma za slučajne ulazne podatke te mogućnosti oblikovanja kriterija konzistentnosti rezultata postupka. Traženi kriterij bi trebao dati mjeru pouzdanosti da je rezultat algoritma svojstvo promatrane scene, a ne posljedica slučaja. Dobiveni rezultati su prikazani u tablici 5.2

Tablica 5.2: Rezultati primjene postupka na skupu slučajnih pravocrtnih segmenata

skup eksperimenata	broj eksperimenata u skupu	broj generiranih pseudo-slučajnih segmenata	broj segmenata dominantnog nedogleda	
			μ	σ
A	20	40	4,1	2,0
B	20	60	6,1	1,5
C	20	80	8,7	2,2

Prvi stupac tablice 5.2 označava pojedini skup eksperimenata na kojeg se odnosi dotični redak. Drugi stupac sadrži broj obavljenih eksperimenata u skupu, dok je broj pripremljenih pseudo-slučajnih pravocrtnih segmenata u svakom eksperimentu prikazan u trećem stupcu tablice. Četvrti i peti stupac tablice prikazuju prosječni broj pravocrtnih segmenata koji podržavaju dominantni nedogled i odgovarajuću standardnu devijaciju, za sve eksperimente skupa.

Rezultati eksperimenata pokazuju zanimljivu pravilnost da je određeni nedogled posljedica strukture scene, ako ga definira više od $1/7$ svih pravocrtnih segmenata u sceni. Primjenom kriterija na rezultate iz tablice 5.1, pokazuje se da su dobiveni pouzdani rezultati samo za smjerove gledanja koji su približno paralelni sa uzdužnom osi hodnika (stupci za 0° i 5°). Glavni razlog za takve ne baš idealne rezultate su pravocrtni segmenti koji su uzrokovani nejednolikim osvjetljenjem, refleksijama i sjenama. Taj problem se teško može riješiti bez uvođenja neke vrste apriornog znanja kojim bi se mogla razlikovati područja visokog gradijenta slike uzrokovana promjenom orijentacije objekata od područja uzrokovanih neravnomjernim osvjetljenjem scene.

Provedeno ispitivanje je možda malo prestrogo jer uopće ne uzima u obzir težine pravocrtnih segmenata. Naime, osim u slučajevima ekstremno nejednolikog osvjetljenja, pravocrtni segmenti koji odgovaraju strukturi okoliša uglavnom imaju veću težinu od segmenata koji su posljedica neželjenih efekata. Nedogledi koji odgovaraju strukturi scene tako mogu biti traženi ne samo analizom brojnosti linijskih segmenata, nego i analizom težine dobivenih nakupina.

Uporabna vrijednost značajki scene se može mjeriti sigurnošću kojom one ukazuju na prisutnost neke strukture ili objekta u sceni. *Posljedična svojstva* scene (*engl.* nonaccidental properties) [51] se općenito definiraju kao slikovne značajke koje se često javljaju kao posljedica objekata u sceni, a vrlo rijetko kao posljedica slučaja. Rezultati eksperimenata ove grupe ukazuju na jednostavan postupak kojim se može odrediti u kojoj se mjeri dobiveni nedogled može tretirati kao posljedično svojstvo scene.

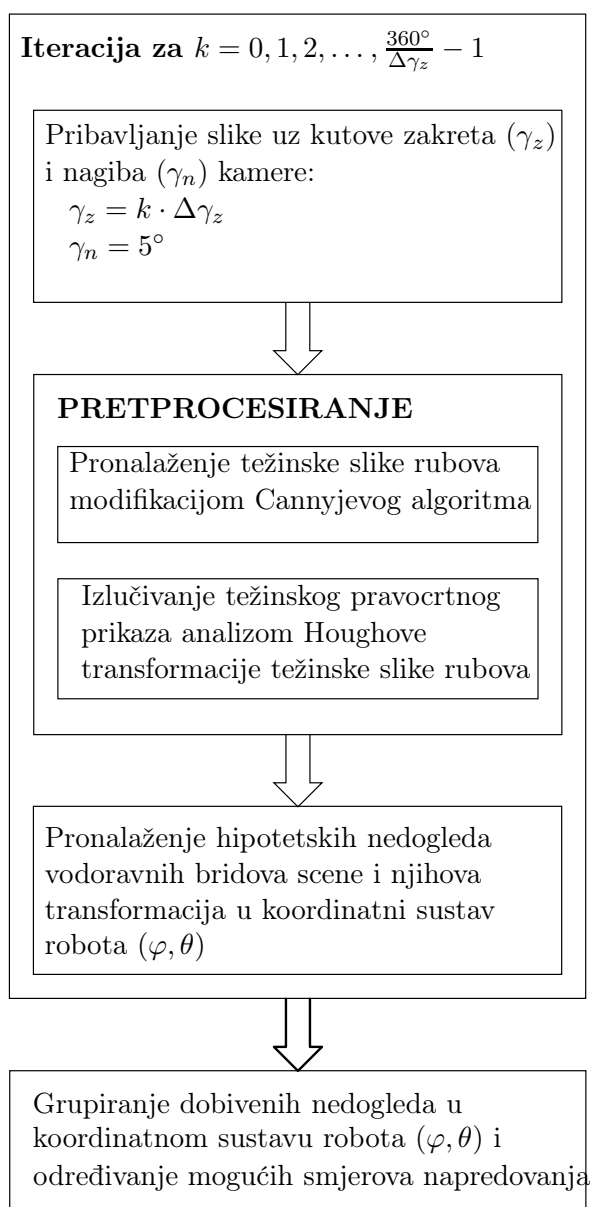
5.2 Lokalizacijska procedura

U ovom odjeljku se predlaže lokalizacijska procedura kojom je moguće odrediti apsolutnu orijentaciju robota u sustavu hodnika, odnosno moguće smjerove napredovanja. Procedura koja je u kraćem obliku već opisana u [93] se temelji na postupku pronalaženja nedogleda

horizontalnih bridova okoliša koji je opisan u prethodnom pododjeljku. Rezultati koji su tim postupkom postignuti na pojedinačnim slikama (tablice 5.1 i 5.2), sugeriraju određivanje smjera pružanja uzdužne osi hodnika na temelju većeg broja slika, za različite kutove zakreta kamere. Takvim pristupom mogli bi se pronaći mogući smjerovi napredovanja u sustavu u kojem se hodnici križaju pod proizvoljnim kutovima. Međutim, kao i kod većine ostalih zatvorenih prostora, hodnici u blizini laboratorija u kojem su izrađeni eksperimenti se križaju isključivo pod pravim kutom. Tako su eksperimenti provedeni na ortogonalnom sustavu hodnika, u kakvom ukupan broj smjerova napredovanja može biti jedan (završetak hodnika), dva (jednostavan hodnik), tri ili četiri (križanje hodnika "T" odnosno "+" oblika).

Blok dijagram lokalizacijske procedure je prikazan na sl. 5.10. Procedura se sastoji od dva koraka. U prvom koraku, iterativno se traže hipotetski nedogledi za sve kutove zakreta kamere. Dobivene točke se prebacuju u koordinatni sustav robota te spremaju za kasniju obradu. Zadatak drugog koraka procedure je određivanje smjerova pružanja hodnika na temelju rezultata grupiranja hipotetskih nedogleda dobivenih u prethodnom koraku.

U predloženoj proceduri, analiziraju se slike pribavljene za slijedeće kuteve zakreta



Slika 5.10: Blok-dijagram predložene lokalizacijske procedure.

kamere

$$\gamma_z = k \cdot \Delta\gamma_z, k = 0, 1, \dots, \frac{2\pi}{\Delta\gamma_z} - 1. \quad (5.3)$$

Pri tome se $\Delta\gamma_z$ postavlja na $\frac{1}{5}$ kuta širine vidnog polja kamere. U provedenim eksperimentima, širina vidnog polja kamere bila je 45° , pa je korišten $\Delta\gamma_z = 9^\circ$ što znači da je konačna odluka donošena na temelju analize 40 slika. Mramorni pod hodnika u kojem su se obavljali eksperimenti dobro reflektira svjetlost, a posebno rasvjetu na stropu hodnika. Posljedica takvih refleksija je da se bridovi koji se nalaze pri vrhu hodnika znatno bolje raspoznaju od bridova pri dnu hodnika. Kut nagiba kamere je zato, prilikom eksperimentiranja, bio postavljen na oko $+5^\circ$, kako bi takvi efekti što manje došli do izražaja. Kut nagiba nije pogodno postaviti na još veću vrijednost, kako bi se uspravni bridovi preslikavali u približno uspravne pravocrtne segmente koji se onda ne moraju razmatrati prilikom traženja nedogleda vodoravnih bridova scene.

Prvi korak procedure se u k -toj iteraciji sastoji iz slijedećih operacija:

1. Pribavljanje slike za analizu

Postavljaju se kutovi zakreta i nagiba kamere $\gamma_z = k \cdot \Delta\gamma_z$ i $\gamma_n = 5^\circ$.

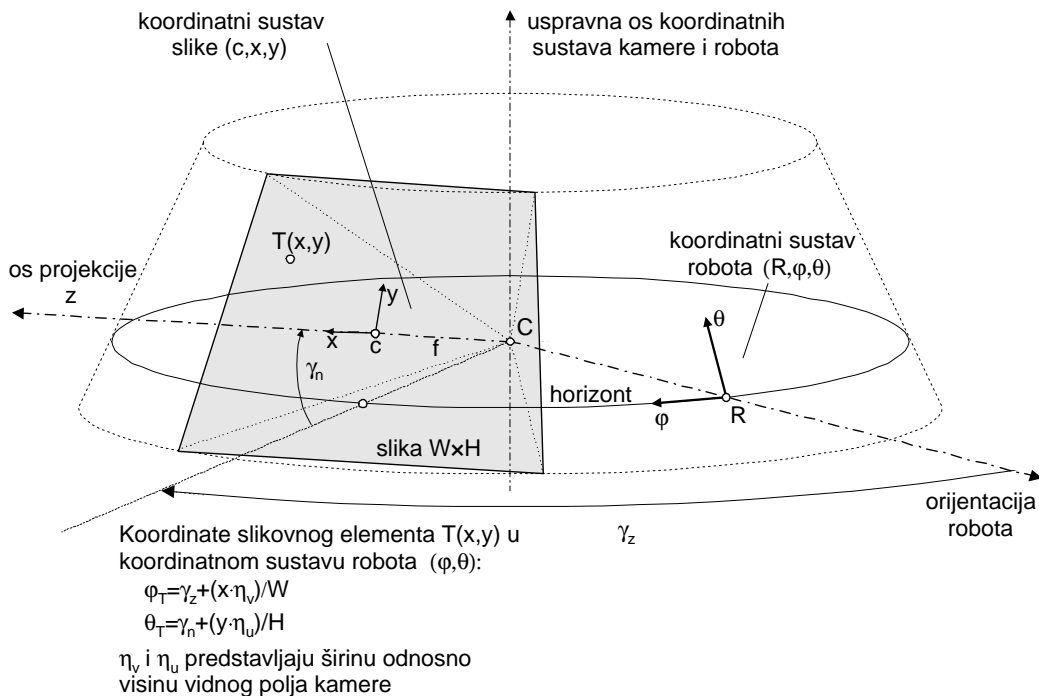
Pribavlja se slika dimenzija 320×240 , sa 24 bita po slikovnom elementu.

2. Pretprocesiranje

Za svaku pribavljenu sliku, pronalazi se težinski pravocrtni prikaz u dva koraka, u skladu s procedurom koja je opisana u odjeljku 4.2.

3. Pronalaženje hipotetskih nedogleda

Pronalaze se hipotetski nedogledi opisani težinskim faktorima, kao što je to opisano u pododjeljku 5.1.4. Tako dobivene točke se transformiraju u koordinatni sustav robota (φ, θ) (sl. 5.11) i spremaju za obradu u drugom koraku obrade.



Slika 5.11: Međusobni odnos koordinatnih sustava robota i slike.

Spomenuta transformacija preslikava hipotetske nedoglede iz kartezijevog koordinatnog sustava slike na plašt stošca čija se glavna os poklapa s uspravnom osi koordinatnih sustava svijeta, robota i kamere. Na plaštu stošca se upotrebljava koordinatni sustav (φ, θ) ,

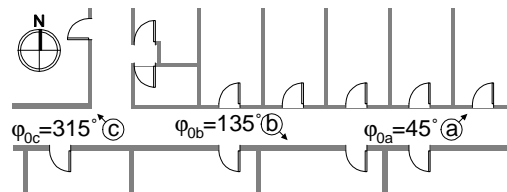
pri čemu pozitivna vrijednost koordinate φ odgovara zakretu u smjeru kazaljke na satu. Ishodište koordinatnog sustava robota (tj. plašta stošca) je definirano presjecištem plašta sa glavnom osi projekcije kamere za kutove zakreta i nagiba od 0° . Odnosi između dvaju koordinatnih sustava su prikazani na sl. 5.11.

U drugom koraku obrade, pronalaze se nakupine hipotetskih nedogleda uz pomoć istog "maximin" algoritma koji je korišten i u pododjeljku 5.1.4. Koordinatni vektori hipotetskih nedogleda se stoga prije grupiranja trebaju normalizirati, što odgovara njihovom preslikavanju na Gaussovu sferu. Mogući smjerovi napredovanja se na kraju određuju na temelju položaja i težine tako dobivenih nakupina.

Pored nedogleda, kao lokalizacijske značajke su isprobani i vodoravni te uspravni pravocrtni segmenti, u kombinaciji sa istom lokalizacijskom procedurom. Rezultati postignuti za takve značajke su međutim bili znatno slabiji od rezultata dobivenih analizom nedogleda.

5.3 Eksperimentalni rezultati

Predložena lokalizacijska procedura je isprobana za tri karakteristična položaja robota u hodniku Zavoda za elektroniku, mikroelektroniku računalne i inteligentne sustave, kao što prikazuje sl. 5.12. U eksperimentima je korištena kamera s mogućnošću podešavanja kutova zakreta i nagiba koja je prikazana na sl. 1.2. Računalo koje je obavljalo obradu je postavljeno na pokretna kolica, a kamera je postavljena na zaslon računala kao njegovu najistaknutiju točku. Razvijeno je programsko sučelje koje omogućuje postavljanje parametara kamere preko serijske veze, upotrebom protokola niske razine dokumentiranog od strane proizvođača kamere. Eksperimenti su izvedeni na računalu sa cjelobrojnomo performansom od oko 2 SPECint95 pa je obrada pojedinih slika trajala oko 4 s, dok je cjelokupna procedura trajala oko 3 minute.



Slika 5.12: Tlocrt Zavodskog hodnika sa ucrtanim položajima na kojima su provedeni eksperimenti te odgovarajućim orijentacijama robota (φ_{0a} , φ_{0b} , φ_{0c}).

Rezultati obrade za svaki od položaja robota su sažeti u tablici 5.3. Redci tablice odgovaraju položajima robota u hodniku **a**, **b** i **c**. Odgovarajuće apsolutne orijentacije robota (φ_{0a} , φ_{0b} , φ_{0c}) u svakom od eksperimenata su prikazane na sl. 5.12. Stupci u desnom dijelu tablice pokazuju rezultate lokalizacijske procedure, odnosno slijedeće podatke o četiri dominantne nakupine hipotetskih nedogleda:

1. koordinata φ položaja nakupine u koordinatnom sustavu plašta stošca — pozitivna vrijednost koordinate φ označava zakret u smjeru kazaljke na satu.
2. relativni težinski faktor nakupine (t);
3. greška izračunata uspoređivanjem položaja nakupine sa stvarnim smjerom pružanja hodnika (δ).

Slikovni prikaz rezultata iz tablice 5.3 je dan na sl. 5.13. Radi jednostavnosti prikaza, točke koordinatnog sustava plašta stošca (φ , θ) su transformirane u polarni koordinatni sustav (r , φ). Pri tome, uspravan smjer prema gore u polarnom koordinatnom sustavu

Tablica 5.3: Rezultati primjenjivanja predložene lokalizacijske procedure na položajima koji su prikazani na sl. 5.12

#	nakupine											
	1			2			3			4		
	φ	t	δ	φ	t	δ	φ	t	δ	φ	t	δ
a	-135	100	0	-98	8	-	150	1	-	62	0	-
b	137	100	2	-45	73	0	-122	0	-	-	-	-
c	-45	100	0	133	45	2	47	39	2	-102	1	-

odgovara kutu od 0° na plaštu stošca, a veći radijus r odgovara točkama sa većom koordinatom θ . Koordinata r se računa po formuli

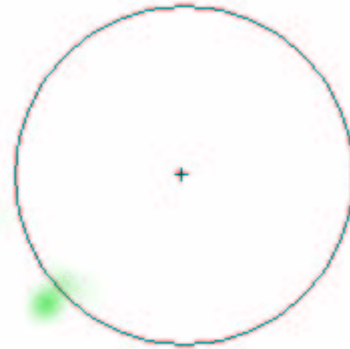
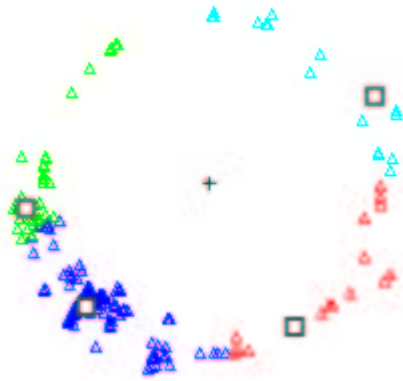
$$r = r_H + \frac{\theta - \gamma_n}{k},$$

gdje je r_H zadani radijus u kojeg se preslikavaju točke stošca koje se nalaze na horizontu (koordinata θ tih točaka iznosi γ_n), dok je k empirijski određena konstanta.

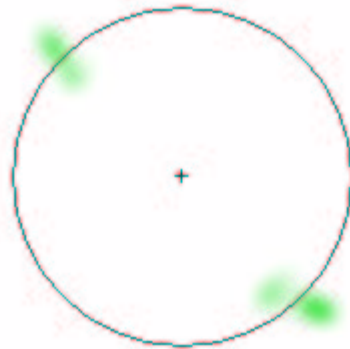
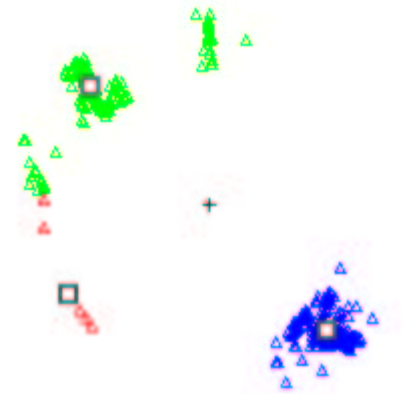
Za svaki eksperiment, prikazane su dvije slike u polarnom koordinatnom sustavu (r, φ) . Trokuti i kvadrati u lijevoj slici svakog para odgovaraju pronađenim hipotetskim nedogledima i određenim središtima nakupina, respektivno. Lijeve slike ne daju dobar kvalitativni prikaz rezultata obrade jer nedogledi koji su daleko od središta nakupina redovito imaju znatno manju težinu od prosjeka. Iz tog razloga, desne slike su predložene kao alternativan prikaz, a oblikovane su u slijedeća tri koraka:

- i) svaki nedogled doprinosi svojom težinom tamnoći odgovarajućeg slikovnog elementa u polarnom koordinatnom sustavu;
- ii) nad slikom se primjenjuje dvodimenzionalna konvolucija s Gausovim filterom uz veliku vrijednost parametra σ ;
- iii) u sliku se ucrtava kružnica s radijusom r_H koja odgovara horizontu.

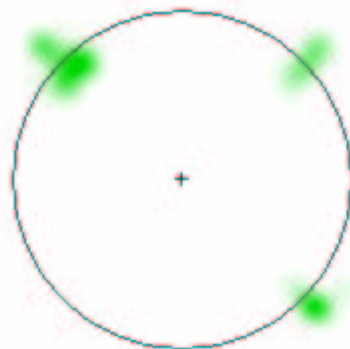
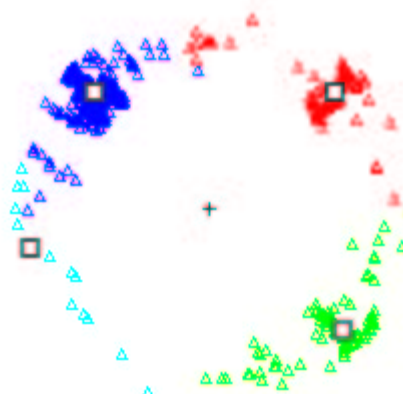
Primjerice, desna slika na sl. 5.13(c) ukazuje na tri moguća smjera napredovanja, što je u skladu sa položajem c na tlocrtu hodnika koji je prikazan na sl. 5.12.



(a)



(b)



(c)

Slika 5.13: Rezultati primjenjivanja predložene procedure na položajima koji su ucrtani na sl. 5.12.

Poglavlje 6

Rasprava

Većina autora se slaže kako je samostalno određivanje položaja (lokalizacija) robota u okolišu prvi i ključni korak za omogućavajuće samostalne navigacije. Unatoč brojnim iskušanim pristupima, međutim, zadovoljavajuća, elegantna i prilagodljiva tehnika lokalizacije još nije pronađena. Od svih senzorskih tehnika koje se danas koriste u robotici, kao potencijalno najmoćniji izvor informacija za lokalizaciju samostalnog robota u literaturi se navodi računarski vid. Ta ocjena vrijedi tim više kod okoliša koji su projektirani prvenstveno za potrebe ljudi čije se snalaženje u prostoru temelji na osjetu vida, te kod primjena u kojima postavljanje aktivnih referentnih objekata u okoliš nije prihvatljivo.

Ipak, u kontekstu trenutnog stanja tehnologije, još uvijek su češće primjene u kojima glavnu ulogu igraju drugi senzori. Tumačenje stvarnih scena je za računarski vid još uvijek presložen, višeznačan i često nerješiv problem. Prednost ljudi pri njegovom rješavanju je znatna i prije početka obrade jer se prilagodljivost modernih kamera neravnomjernom osvjetljenju u sceni još uvijek ne može mjeriti s ljudskim okom. Međutim, razlika u sposobnosti tumačenja ostaje gotovo nepromijenjena čak i kad se apstrahira razlika u metodi pribavljanja slike: ljudi nedvosmisleno tumače i *slike* stvarnih scena, kada na raspolaganju nisu ni stereo vid ni pokret. Snalaženje u prostoru na temelju slika je toliko prirodno čovjeku da se čini nevjerojatnim kako računala na tom području postizu relativno skromne rezultate.

Priroda veze između vidnog podražaja i konačnog uporabnog opisa scene kod ljudske percepcije je vrlo slabo poznata, pa ju je potrebno istraživati na čim lakšim problemima. Vrlo dobar poligon za takva istraživanja je hodnik — arhitektonsko rješenje koje je u zapadnoj kulturi univerzalno prihvaćeno kao okoliš koji omogućuje čim praktičnije i jednostavnije snalaženje ljudi. Mi vjerujemo da se u takvom, jednostavnom okolišu, prije nego u bilo kom drugom, može pronaći put prema naprednijim samostalnim pokretnim tehničkim sustavima čiji bi glavni alat za snalaženje u prostoru bio računarski vid.

6.1 Mogući pristupi lokalizaciji

Prvi pristup koji se nameće za lokalizaciju samostalnog robota jest primjena metoda s područja raspoznavanja objekata. Rezultati preliminarnog istraživanja na tom području međutim pokazuju kako te metode postizu bitno slabije rezultate kad parametri pribavljanja slike (osvjetljenje, položaj kamere, smjer gledanja kamere) nisu potpuno određeni. Pokazuje se da ti parametri kod stvarnih scena u velikoj mjeri utječu na količinu informacije u pribavljenoj slici. Eksperimentalni rezultati su pokazali da u razmatranom okolišu, za većinu smjerova gledanja iz zadanog položaja kamere, pribavljene slike ne sadrže informaciju koja bi bila korisna za snalaženje u prostoru. Vjerujemo da slični odnosi vrijede i kod velikog broja ostalih tipova okoliša.

Zbog navedenih razloga, kod stvarnih je scena vrlo teško primijeniti metode poklapanja značajki slike sa trodimenzionalnim modelima poznatih ili nepoznatih objekata. Prihvaćen

je stoga inkrementalni pristup tumačenju scene u kojem robot usmjerava svaki novi korak obrade u cilju upotpunjavanja svog znanja o Svijetu. U svakom koraku obrade, postupci pribavljanja i obrade slika se propisuju ravnopravno, u skladu sa načelima aktivnog vida. Zbog nekontroliranih uvjeta osvjetljenja u sceni, koraci obrade trebaju biti jednostavni i robustni kako bi postupak snalaženja u prostoru bio stabilan. Temeljni korak obrade u takvom sustavu je lokalizacijska procedura koja se primjenjuje kad nije nikakvo znanje o položaju i orijentaciji robota u okolišu nije poznato.

U sklopu rada, ostvarena je lokalizacijska procedura kojom se mogu odrediti smjerovi napredovanja robota u složenom sustavu hodnika. Analiza smjera pružanja hodnika u trenutku kada nikakvo znanje o strukturi scene nije dostupno je prikladna jer je riječ o najizraženijem svojstvu strukture razmatranog okoliša. Pored toga, prikupljeno znanje može biti korisno i u daljnjim koracima interpretacije okoliša: tumačenje stvarnih scena se u radovima [30, 34, 56] temelji na apriorno poznatoj glavnoj strukturnoj osi okoliša.

6.2 Međuprikaz

Zbog pravilne strukture okoliša, svaka pribavljena slika se u fazi pretprocesiranja opisuje skupom pravocrtnih segmenata. Pravocrtni opis je kompromisno rješenje između prikaza koji sadrži čim više informacije o ulaznoj slici i prikaza nad kojim se mogu oblikovati efikasni algoritmi. Crtež je jedan od osnovnih načina ljudskog komuniciranja (stripovi, tehnički crteži) pa je njegova prikladnost što se tiče mogućnosti prikaza potrebnih informacija neupitna. S druge strane, crtež ipak nije ikonički, slikovni prikaz — njegovi građevni elementi su izdvojene crte i ima ih bitno manje (tipično, oko 50) nego slikovnih elemenata izvornih slika (preko 50 000). Stoga je složenost obrade crteža u pravilu bitno manja od složenosti obrade slika.

6.3 Odabrani pristup lokalizaciji

Odabrani pristup lokalizaciji je napravljen po ugledu na proceduru koju prilikom snalaženja u sustavu hodnika obavljaju ljudi. Kada čovjek kroz vrata izađe u nepoznat hodnik, on se prije svega mora osvrnuti oko sebe kako bi doznao smjerove pružanja tog hodnika. U predloženoj lokalizacijskoj proceduri se, analogno proceduri koju koriste ljudi, analizira veći broj slika pribavljenih za ravnomjerno raspoređene kutove zakreta kamere. U svakoj pribavljenoj slici, traže se točke nedogleda vodoravnih pravaca okoliša kao značajke koje najviše karakteriziraju razmatrani okoliš. Mogući smjerovi napredovanja u sustavu hodnika kao konačni rezultat lokalizacijske procedure, određuju se tek nakon pronalaženja nedogleda u svakoj pribavljenoj slici, na temelju najizraženijih nakupina točaka nedogleda u koordinatnom sustavu robota.

Loše svojstvo opisanog pristupa je da se temelji na pretpostavci da okoliš ima ortogonalnu strukturu pa ne može biti primijenjen u okolišu koji takvu strukturu nema. Alternativna rješenja u ovom trenutku su određivanje strukture na temelju kretanja odnosno stereo vida, a zasnivaju se na analizi slika koje su dobivene iz različitih položaja prostora. Ta rješenja međutim zahtijevaju složenije procedure i sklopovlje za pribavljanje slika te složene algoritme podudaranja značajki¹ u različitim slikama.

¹U ovom slučaju vjerojatno bi bilo najpogodnije pronalaziti podudaranje uspravnih pravocrtnih segmenata slika dobivenih za položaje kamere sa istom Z koordinatom, a različitim X ili Y koordinatama u koordinatnom sustavu svijeta.

6.4 Eksperimentalni rezultati

Obavljeni eksperimenti se mogu svrstati u dva skupa. U prvom skupu eksperimenata, tražena je dominantna točka nedogleda vodoravnih pravaca u pojedinačnim slikama. Eksperimenti iz drugog skupa su se sastojali iz primjene predložene lokalizacijske procedure u raznim lokacijama hodnika.

Prvi skup se sastojao od tri grupe eksperimenata. Prva grupa eksperimenata je pokazala da se određivanjem dominantnog nedogleda pouzdano može odrediti smjer pružanja hodnika u slikama koje su dobivene za smjerove gledanja kamere približno paralelne uzdužnoj osi hodnika. U drugoj grupi eksperimenata, analizirani su dominantni nedogledi slika dobivenih iz zadanog položaja, za razne kuteve zakreta kamere. Rezultati su pokazali da velika većina slika koje su dobivene za kutove zakreta koji od smjera uzdužne osi hodnika odstupaju više od širine vidnog polja kamere ne sadrži informacije korisne za lokalizaciju. Također, pokazano je da se broj pravocrtnih segmenata koji određuju dominantni nedogled može koristiti kao mjera pouzdanosti rezultata postupka. Konačno, u trećoj grupi eksperimenata prvog skupa, analizirani su rezultati algoritma za slučajne ulazne podatke. Dobiveni rezultati su potvrdili pouzdanost eksperimenata iz prethodne dvije grupe, a na temelju njih je predložen heuristički postupak za ocjenu pouzdanosti rezultata algoritma.

Drugi skup eksperimenata se sastojao u primjeni predložene lokalizacijske procedure na različitim položajima hodnika. Rezultati su pokazali stabilan rad procedure unatoč slabim uvjetima osvjetljenja, na svim karakterističnim lokacijama hodnika (završetak hodnika, jednostavan hodnik, križanje). Za sve obavljene eksperimente iz skupa, određeni mogući smjerovi napredovanja su odstupali od pravih vrijednosti za manje od 2° .

6.5 Budući rad

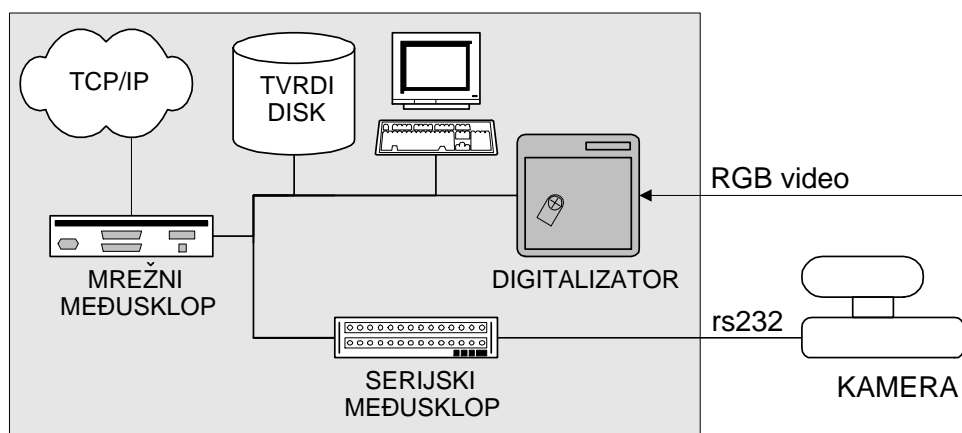
Cilj budućeg rada jest daljnje približavanje samostalnom snalaženju u prostoru uz pomoć računarskog vida. Buduće procedure bi se trebale naslanjati na znanje o strukturi scene koje se može steći procedurom koja je opisana u ovom radu. Rezultat novih procedura bi trebao biti proširivanje znanja o Svijetu u kojem se robot nalazi. Mogući ciljevi u tom smislu bi mogli biti određivanje visine i širine hodnika, udaljenosti robota od uzdužnih zidova, određivanje položaja vrata i sl.

Također, bilo bi zanimljivo istražiti mogućnost određivanja udaljenosti bridova scene od kamere na temelju stereo vida ili određivanja strukture iz kretanja. Najbolji kandidati za to su naravno bridovi čiji je smjer pružanja okomit na optičku os kamere, a u razmatranom okolišu se preslikavaju u uspravne ili vodoravne pravocrtne segmente slike. Na temelju dobivenih udaljenosti, također bi se mogli odrediti mogući smjerovi napredovanja pa bi se dva različita pristupa mogla usporediti ili međusobno potpomagati.

Dodatak A

Upravljanje kamerom i prijenos slika

U ovom dodatku, opisani su programski podsustavi koji čine sučelje prema specifičnom sklopovlju na kojem se temelji predložena izvedba lokalizacijske procedure. Shematski prikaz sklopovlja u sustavu je prikazan na sl. A.1.



Slika A.1: Shematski prikaz sklopovlja u sustavu.

Pri oblikovanju obaju podsustava postavljeni su slični redosljedi prioriteta: efikasnost, lakoća testiranja, lakoća održavanja, mogućnost višestrukog korištenja i prenosivost [49]. Postizanje tih prioriteta je razrađeno u odjeljku A.1. Odjeljak A.2 opisuje notaciju koja je korištena za prikaz arhitektura podsustava u skladu sa objektno orijentiranim pristupom oblikovanju programske podrške. Konačno, u odjeljcima A.3 i A.4, postavljaju se precizni zahtjevi na svaki od programskih podsustava i u skladu s tim zahtjevima predlažu se njihove arhitekture i izvedbe u programskom jeziku C++.

A.1 Prioriteti oblikovanja programske podrške

Temeljni problem u izgradnji programske podrške je borba sa složenošću velikih sustava [48]. Međutim, glavno svojstvo programske podrške je njena upotrebna vrijednost pa je osiguravanje maksimalne efikasnosti na kritičnom putu izvođenja često prioritarna. Stoga su u oblikovanju programske opreme poštovani slijedeći prioriteti pod uvjetom da njihovo zadovoljenje ne šteti brzini izvođenja kritičnih operacija:

1. lakoća održavanja i testiranja;
2. mogućnost višestrukog korištenja;
3. prenosivost;

Tako postavljeni prioriteti povoljno utječu na vrijeme razvoja aplikacije jer smanjuju njenu složenost, a time istovremeno i pojednostavljaju održavanje. Programski podsustavi koji su izgrađeni u skladu s tim prioritetima smanjuju troškove budućih projekata jer se mogu jednostavno proširivati i koristiti u drugim kontekstima. U slijedećim pododjeljcima, bit će opisane tehnike programiranja i konteksti u kojima se pojedini prioriteti mogu postići, a posebno će biti istaknute i razrađene konfliktne situacije među prioritetima.

A.1.1 Lakoća održavanja i testiranja

Održavanje se općenito odnosi na unošenje promjena u program, bilo to sa svrhom nadograđivanja ili ispravljanja grešaka. Nije slučajno da su ta dva pojma objedinjena pojmom “održavanje” budući da su tijekom izgradnje programskog sustava oni vrlo usko povezani, ponekad čak i cikličnim uzročno-posljedičnim vezama. Unošenje promjena u program je, s druge strane, obično vezano za ponovno testiranje dijela sustava u kojem je promjena unesena. Pokazuje se da je taj iterativni postupak (nadogradnja - testiranje - ispravljanje grešaka) kritična faza u izgradnji većih programskih sustava. Upravo zato je jednostavno održavanje i testiranje postavljeno ovako visoko na listi prioriteta kôdiranja, odmah iza efikasnosti kritičnih operacija aplikacije. U širem obliku, “lakoća” održavanja i testiranja označava da se održavanje može izvesti uz:

1. čim manje izmjene u izvornom kôdu;
2. odražavanje unesenih promjena na čim manje prevedenih datoteka koje se povezuju u izvršni kôd.

Minimiziranje propagacije grešaka u prevedenim datotekama je poželjno kako bi se testiranje moglo obaviti na hijerarhijski čim nižoj razini, uz minimalno vrijeme prevođenja i povezivanja izmijenjenog programskog sustava.

Prilikom oblikovanja programskih sustava, uspostavljaju se odnosi među objektima iz logičke i fizičke domene. Logički entiteti su pri tome tipovi podataka, potprogrami, klase i sl. Logička organizacija definira funkcionalnost programa i potpuno je nezavisna o fizičkoj organizaciji. Fizičko oblikovanje se odnosi na organizaciju fizičkih entiteta programskog sustava — datoteka sa izvornim i prevedenim kôdom, direktorija i biblioteka [52]. Pažljiva fizička organizacija postaje značajna kod programskih sustava srednje veličine (preko 25 tisuća linija izvornog kôda), a postaje izvanredno bitna kod velikih sustava (preko 100 tisuća linija izvornog kôda) kada se vremena prevođenja i povezivanja cijelog sustava počinju mjeriti danima. Temeljna jedinica fizičke organizacije je *komponenta* i odgovara paru datoteka izvornog kôda od kojih jedna definira sučelje, a druga izvedbu komponente. Kao što se može i pretpostaviti, propagaciju izmjena u izvornom kôdu određuje logička organizacija sustava, dok će fizička organizacija imati presudni utjecaj na propagaciju izmjena u izvršnom kôdu.

Minimiziranje propagacije izmjena u izvornom kôdu

U skladu sa prethodnom argumentacijom, prvo poželjno svojstvo programskog sustava je mogućnost njegovog mijenjanja i nadograđivanja uz minimalne i čim više lokalne izmjene u postojećem izvornom kôdu. Ta svojstva se postižu pažljivim logičkim oblikovanjem programskog sustava, upotrebom *enkapsulacije* (*engl.* encapsulation) i *polimorfizma* (*engl.* polymorphism) [42]. Obje tehnike programiranja se prirodno izražavaju u jezicima koji podržavaju *objektno orijentirano* oblikovanje (npr. Smalltalk, Eiffel, C++), iako se mogu izvesti i u proceduralnim jezicima uz stanoviti dodatni napor.

Enkapsulacija označava skrivanje izvedbenih detalja pojedinih jedinica logičkog oblikovanja programskog sustava iza njihovih proceduralnih sučelja [52]. Tom tehnikom se

omogućuje mijenjanje izvedbenih detalja programskih podsustava bez potrebe za mijenjanjem hijerarhijski viših dijelova sustava. Većina modernih programskih jezika podržava automatizirano unošenje manjih potprograma u izvršni kôd (*engl.* inline expansion) kako bi se zaobišla dugotrajna operacija grananja na potprogram. Kod tih jezika, enkapsulacija se u najvećem broju slučajeva može postići bez utjecaja na efikasnost konačnog sustava. Stoga je to jedna od rijetkih tehnika programiranja koja nije u pravilu vezana za neželjene efekte i kompromise.

Jednostavno nadograđivanje sustava se vrlo efikasno može osigurati polimorfizmom. Polimorfizam označava mogućnost primjene istih procedura na različite objekte sa identičnim sučeljem. Tako se može postići da *stare* procedure rade sa *novim* komponentama bez potrebe za mijenjanjem prethodno napisanog kôda. Polimorfizam, za razliku od enkapsulacije, ima svoju cijenu jer je *poziv sa dinamičkim vezivanjem* kojim se ostvaruje polimorfizam skuplji od klasičnog, *statički vezanog poziva*. Štoviše, polimorfni poziv onemogućuje unošenje potprograma u izvršni kod (*engl.* inline expansion) što se može još štetnije odraziti na efikasnost. Tako polimorfni potprogram koji radi vrlo jednostavan posao, npr. dohvaća podatak koji je izvedbeni detalj nekog objekta, može biti višestruko sporija od statički vezanog potprograma. U tim slučajevima potrebno je napraviti kompromis između prilagodljivog i efikasnog kôda ili pokušati dobiti i jedno i drugo generičkim programiranjem [76]. Generičko programiranje je relativno novo područje pa je vezano sa značajnim poteškoćama u razvoju zbog nedostatka literature, neistitranih standarda te prevodioca i biblioteka koji kasne za standardom.

Minimiziranje propagacije izmjena u izvršnom kôdu

Osobine dobre fizičke organizacije su minimalne međusobne ovisnosti pojedinih komponenta sustava prilikom prevođenja, povezivanja i testiranja. Time se dobiva hijerarhijski sustav koji se može oblikovati, prevesti, testirati i shvatiti metodom “podijeli pa vladaj”. Ključni koncept fizičkog oblikovanja je *izolacija* (*engl.* isolation) [52] koji se odnosi na uklanjanje ovisnosti ostatka sustava o izvedbi odabranih komponenta. Razlikuje se pri tome potpuna izolacija i različiti oblici djelomične izolacije. Izvedba potpuno izolirane komponenta se može po volji mijenjati bez potrebe za ponovnim prevođenjem ijedne druge komponente sustava. Tako se veliki sustav može graditi od zasebno testiranih komponenti sa dobro definiranim sučeljima čime se složenost postupka značajno smanjuje.

Glavni alat za postizanje izolacije je polimorfizam. Izolacija stoga ima svoju cijenu jer je komunikacija preko granica izoliranih komponenti nešto manje efikasna nego u slučaju kad komponente nisu izolirane. Zato nije prikladno primijeniti potpunu izolaciju na sve komponente iako bi to vodilo najprilagodljivijem sustavu. U sadašnjem vremenu kada je procesorsko vrijeme još uvijek kritičan resurs, potrebno je pažljivo odabrati komponente koje su kritične za performansu i na njih primijeniti stupanj izolacije koji neće smanjiti efikasnost sustava.

A.1.2 Mogućnost višestrukog korištenja

Izgradnja programske podrške je skup i dugotrajan proces. To je toliko izraženo da se u gotovo svakom komercijalnom projektu više isplati kupiti neku komponentu sustava nego je iznova graditi [68]. Tada je jasno da je tim više povoljno ponovo iskoristiti neku od prethodno izgrađenih komponenti u usporedbi sa njenom ponovnom izgradnjom. Pokazuje se da su komponente sustava koji se lako održava i testira (vidi prethodni pododjeljak) vrlo dobri kandidati za višestruku upotrebu. Međutim, višestruko korištenje komponenti programskog sustava obično ima svoju cijenu, a plaća se ponekad efikasnošću i glomaznošću komponente te dodatnom međusobnom ovisnošću komponenta sustava, a uvijek dodatnim naporom pri njihovom oblikovanju. Dok je spomenuti dodatni napor u najvećem broju

slučajeva dobra investicija, povećanje ovisnosti među komponentama je često ograničavajući faktor višestrukog korištenja komponente.

Neke komponente nije isplativo upotrijebiti na više mjesta unutar aplikacije iako će izvedba sa više komponenti možda rezultirati značajnom količinom redundantnog kôda. Višestruko korištenje komponente može biti povezano sa ugrađivanjem prevelikog broja svojstava u komponentu i potrebom za robustnim procedurama ispravljanja grešaka koje se mogu pojaviti zbog upotrebe u različitim kontekstima. Takve pojave povećavaju cijenu razvoja i održavanja te mogu smanjiti efikasnost i povećati veličinu izvršnog kôda.

A.1.3 Prenosivost

Prenosivost označava mogućnost prenošenja nekog programskog sustava na novi računalni sustav (platformu), sa različitim prevodiocem, operacijskim sustavom ili procesorom. Jasna je prednost koju pruža konstrukcija programskog sustava koji zadovoljava taj kriterij: neovisnost o nedostacima bilo kojeg konkretnog proizvođača sklopovlja ili programske opreme. Ipak, u praksi su rijetki programi koji se istovremeno mogu prevesti na različitim računalima. Postoje četiri glavna uzroka pojave neprenosivog kôda:

1. Potreba za specifičnim optimizacijama;
2. Korištenje jezičnih konstrukta koji su specifični za dani prevodioc;
3. Potreba za korištenjem biblioteka koje su specifične za danu platformu;
4. Potreba za pristupom specifičnom sklopovlju;

Specifične optimizacije su do prije deset godina bile glavna prepreka prenosivim programima. Kritični dijelovi programa su se tada u pravilu pisali u strojnom kôdu koji je specifičan za dani procesor. Međutim, u novije vrijeme su se pojavili optimizirajući prevodioci koji proizvode izvanredno dobar izvršni kôd, jednako kvalitetan ili čak bolji od onog kojeg bi napravio čovjek sa dobrim znanjem arhitekture ciljnog procesora. Pored toga, programiranje novih procesora sa protočnom ili superskalarnom arhitekturom je iznimno složen posao koji još više obeshrabruje ručno dotjeravanje izvršnog kôda.

Drugi uzrok neprenosivosti se u najvećem broju slučajeva može zaobići jednostavnim ignoriranjem nestandardnih ekstenzija korištenog prevodioca. Iskustvo pokazuje da se takav pristup dugoročno najviše isplati čak i ako se postojeći program ne namjerava prenositi na novu platformu.

Proizvođači razvojnih alata često uz klasične aplikacije koje čine razvojnu okolinu (prevodioc, *linker*, *debugger*) nude i svoje specifične biblioteke za jednostavniji i brži razvoj korisničkih aplikacija. Tu se nude dva rješenja i jako je bitno na samom početku odabrati ispravno od njih. Ako ponuđena funkcionalnost značajno može ubrzati razvojni proces, onda se može donijeti odluka da se prenosivost žrtvuje zbog jeftinijeg razvoja. Suprotno, ako je funkcionalnost ponuđenih biblioteka od marginalnog značaja za razvoj aplikacije, isplati se razmotriti odbacivanje biblioteka koje nisu propisane standardom korištenog jezika. U ovom slučaju, može se dogoditi da prenosivost bude besplatna jer su standardne biblioteke modernih jezika su iznimno bogate vrlo iskoristivim konstruktima¹ koji su ili jednako kvalitetni ili čak i efikasniji od vlasničkih rješenja.

Konačno, posljednji uzrok neprenosivosti se ne može zaobići, ali se zato mogu minimizirati njegovi efekti. Neprenosivi dio kôda se može lokalizirati i tako postići da najveći dio aplikacije ipak ostane prenosiv. Metode za lokaliziranje “osjetljivih” dijelova aplikacije su dobro poznate upravo zato što je to iznimno važno arhitektonsko rješenje u velikim sustavima [42]. O tim metodama će biti više riječi u slijedećem pododjeljku.

¹Primjerice, samo dio standardne biblioteke jezika C++ sa tzv. generičkim klasama i algoritmima je vrlo šuro opisan u knjizi od preko 500 stranica [76].

Fundamentalni problem sa ostvarivanjem prenosivosti leži u tome što, tržište radi protiv nje kada ista tvrtka razvija i operacijski sustav i prevodioc. Takvoj tvrtki nije u interesu da korisnici razvijaju aplikacije za konkurentni operacijski sustav na njenoj razvojnoj okolini. U tim slučajevima postizanje prenosivosti se otežava nepodržavanjem standardiziranih specifikacija programskog jezika i odgovarajućih biblioteka te promoviranjem vlasničkih biblioteka i formata datoteka.

A.2 Objektno orijentirano oblikovanje

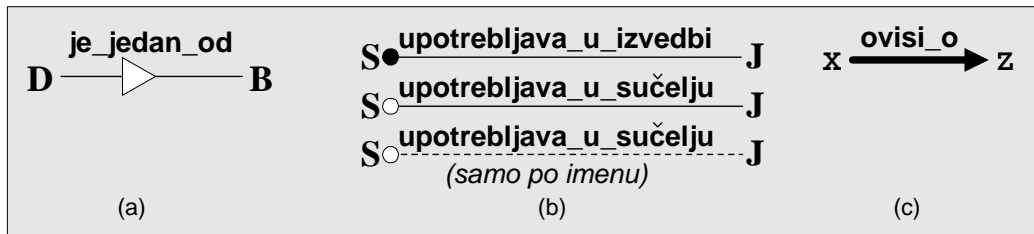
U podsustavima koji se u ovom dodatku opisuju, javlja se potreba za korištenjem polimorfizma i enkapsulacije. Stoga je odabran objektno orijentirani pristup oblikovanju programske podrške i shematski prikaz arhitekture koji se kod takvog pristupa često prakticira. Tako se za oblikovanje odnosa među logičkim i fizičkim entitetima koristi notacija predložena u [52], dok se logički entiteti prikazuju u skladu sa tzv. OMT notacijom [42].

U objektno orijentiranom oblikovanju, koncepti se opisuju *klasama* (*engl.* class) koje su temeljne jedinice logičkog oblikovanja. Klasa je struktura koja objedinjuje skup podataka i operacija nad njima pa se njom prirodno ostvaruje enkapsulacija. Građenje novih koncepata specijalizacijom postojećih se može oblikovati *nasljeđivanjem* (*engl.* inheritance) klase koja opisuje postojeći, općenitiji koncept. Tim postupkom, klasa koja opisuje specijalizirani koncept deklarira da zadržava sučelje postojeće klase, a mijenja izvedbene detalje ili proširuje sučelje i izvedbu općenitijeg koncepta. Operacije klase mogu biti *statički* i *dinamički* vezane. Dinamički vezani pozivi operacija klase omogućuju polimorfizam, tj. da isti poziv efektivno poziva različite dijelove kôda, u ovisnosti o tipu objekta nad kojim se operacija obavlja. Često se novi koncepti grade na temelju postojećih komponenata, ali bez zadržavanja njihovih sučelja. Takav postupak se naziva *ugrađivanjem* (*engl.* aggregation) i njime se prirodno ostvaruje višestruka upotreba postojećih koncepata.

Prilikom oblikovanja arhitekture sustava, prikladno je obratiti pažnju na slijedeće relacije među klasama [52]:

1. **A je jedan od B** (*engl.* is_a, is_a_kind_of)
Klasa **A** nasljeđuje sučelje klase **B**. Koncept kojeg opisuje **A** je specijalizacija koncepta **B**. Objekti klase **A** se moraju moći pojaviti u svim kontekstima u kojima se mogu pojaviti i objekti klase **B**.
2. **A upotrebljava u sučelju B** (*engl.* uses_in_the_interface)
Klasa **B** se pojavljuje u sučelju klase **A**, kao jedan od argumenata ili rezultata operacija te klase ili kao jedan od njenih sastavnih dijelova. Korisnici klase **A** ne mogu biti izolirani od promjena u sučelju klase **B** te u slučaju takvih izmjena moraju ponovo prevesti izvorni kôd.
3. **A upotrebljava u sučelju (samo po imenu) B** (*engl.* uses_in_the_interface (in name only))
Ime klase **B** se pojavljuje u sučelju klase **A**, međutim ne i njena definicija. Korisnici klase **A** su time izolirani od promjena u sučelju klase **B**.
4. **A upotrebljava u izvedbi B** (*engl.* uses_in_the_implementation)
Klasa **B** se javlja u implementaciji klase **A** kao njen izvedbeni detalj. Korisnici klase **A** su izolirani od promjena u sučelju klase **B**.

Fizički odnosi se prikazuju relacijom međusobne ovisnosti komponenti kao temeljnih jedinica fizičkog oblikovanja. Uvodi se binarna relacija **ovisi_o** koja označava da se funkcionalnost klase iz prvog argumenta ne može testirati bez prethodnog testiranja klase iz drugog argumenta. Relacija **ovisi_o** se uvijek javlja kao posljedica relacija **je_jedan_od** i **upotrebljava_u_sučelju**, a gotovo uvijek kao posljedica ostalih oblika relacije **upotrebljava**. Grafički prikaz opisanih relacija je pokazan na sl. A.2.



Slika A.2: Grafički prikaz relacija **je_jedan_od** (a) i tri oblika relacije **upotrebljava** (b) među klasama, te relacije **ovisi_o** među komponentama (c).

A.3 Programski podsustav za upravljanje kamerom

U ovom odjeljku, opisuju se zahtjevi, arhitektura i izvedba programskog podsustava za upravljanje kamerom.

A.3.1 Zahtjevi

Programski podsustav za upravljanje kamerom trebao bi omogućiti podešavanje parametara upravljive kamere. U kontekstu ovog rada, najzanimljivije je upravljanje kutevima zakreta i nagiba kamere, ali su u perspektivi zanimljivi i žarišna daljina, širina vidnog polja kamere te otvor objektiva, trajanje ekspozicije i sl. Podsustav bi trebao biti primjenljiv u raznim aplikacijama, od interaktivnih do potpuno automatiziranih. Stoga je potrebno izraditi prijateljsko sučelje prema čovjeku, ali potpuna funkcionalnost mora biti dostupna neovisno o tom sučelju.

Najvažnije je da podsustav ispravno radi sa kamerom EVI D-31 tvrtke Sony koja je bila na raspolaganju prilikom izvedbe ovog rada. Lijepa osobina upravljivih kamera tvrtke Sony je da za njihovo upravljanje preko serijske veze (RS232) postoji standardni protokol koji je dobro dokumentiran. Radi se o tzv. VISCA protokolu (*engl.* Video System Control Architecture) koji je razvijen u tvrtki Sony za računalno upravljanje višestrukim kamerama. Tako se upotrebom VISCA protokola preko serijske veze može upravljati svim uređajima koji taj protokol podržavaju. Ipak, taj protokol nije općenito priznat i ne može se očekivati da će ga podržavati i svi ostali proizvođači. Zato bi oblik komunikacije sa kamerom (protokol) trebao biti izolirani izvedbeni detalj o kojem ostatak sustava ne bi smio ovisiti. Nadogradnja podsustava u smislu podržavanja kamera sa drukčijim sučeljem, ne bi se smjela propagirati na korisnike: postojeći korisnički kôd bi trebao raditi i sa novom podrškom za novu kameru.

Pored upravljanja kamerom koja je priključena na lokalno računalo, u perspektivi su zanimljive primjene u kojima je kamera značajno udaljena od računala na kojem se odvija obrada. TCP/IP je trenutno najzastupljeniji protokol kako kod lokalnih računalnih mreža tako i kod globalne mreže (Interneta) pa se nameće kao komunikacijsko rješenje za takvu primjenu. Zato se postavlja dodatni zahtjev, da se kamerom transparentno može upravljati i preko TCP/IP protokola.

Budući da je pokretanje kamere mehanička operacija te da se komunikacija sa kamerom odvija preko VISCA protokola pri 9600bps, očekuje se da brzina obavljanja traženih operacija neće biti kritična. Stoga se oko efikasnosti postupaka ne treba pretjerano truditi i više se može inzistirati na prilagodljivosti, mogućnosti višestrukog korištenja i prenosivosti.

A.3.2 Arhitektura

U opisanim zahtjevima, javljaju se se slijedeći ključni koncepti:

- **apstraktna kamera** sa mogućnošću podešavanja zakreta, nagiba, žarišne daljine, širine vidnog polja itd;

- **stvarna kamera** tvrtke Sony kojom se upravlja preko serijske veze, po protokolu VISCA;
- **udaljena kamera** koji dobivene zahtjeve prenosi konkretnoj kameri na drugom računalu, preko protokola TCP/IP;
- prijateljsko korisničko **sučelje** kojim se omogućava upravljanje kamerom upotrebom miša;

Ključni problem je osigurati transparentnost pristupa udaljenoj kameri. To je čest i dobro dokumentiran problem u izgradnji programske podrške, a rješava se *obrascem oblikovanja* (*engl.* design pattern) [42] koji se naziva *Proxy*. Taj obrazac se primjenjuje u situacijama u kojima je potreban općenitiji ili sofisticiraniji način predstavljanja zadanog objekta od običnog pokazivača. U ovom slučaju, primjenjiva je varijanta obrasca koja se naziva *udaljeni proxy* [42] ili *ambasador* (*engl.* Ambassador) [26]. Tom varijantom, oblikuje se lokalni predstavnik objekta koji se nalazi u različitom adresnom prostoru. U skladu sa tim obrascem, **udaljena kamera** je predstavnik **stvarne kamere** na lokalnom računalu, a oba koncepta nasljeđuju sučelje (**je jedan od**) **apstraktne kamere**. Objekt koji predstavlja **stvarnu kameru** zajedno sa programom poslužiocem “živi” na računalu na koje je kamera spojena, dok objekt koji predstavlja **udaljenu kameru** “živi” na računalu na kojem se obavlja obrada slike i komunicira sa **stvarnom kamerom** preko poslužioca i TCP/IP protokola. Korisnik je u ovom slučaju program za obradu slike, a **udaljenoj kameri** pristupa preko sučelja **apstraktne kamere** i zapravo ne zna da li upravlja lokalnom ili udaljenom kamerom.

A.3.3 Izvedba

Koncept **stvarna kamera** mora imati mogućnost slanja i primanja podataka preko serijske veze, dok **udaljena kamera** mora komunicirati preko TCP/IP protokola. Pogodno je serijsku i TCP/IP komunikaciju izdvojiti iz **stvarne kamere** i **udaljene kamere** i opisati zasebnim konceptima kako bi se omogućila njihova kasnija ponovna upotreba². Time se dobivaju nova dva koncepta:

- **serijska veza** — omogućavanje komunikacije preko serijske veze RS232;
- **TCP/IP veza** — omogućavanje komunikacije preko TCP/IP protokola;

Arhitektura programskog podsustava za upravljanje kamerom oblikovana je u skladu s međusobnim odnosima izdvojenih koncepata. Pri tome je svakom od koncepata pridružena klasa kao što je prikazano u tablici A.1. Predložena arhitektura podsustava je prikazana na sl. A.3.

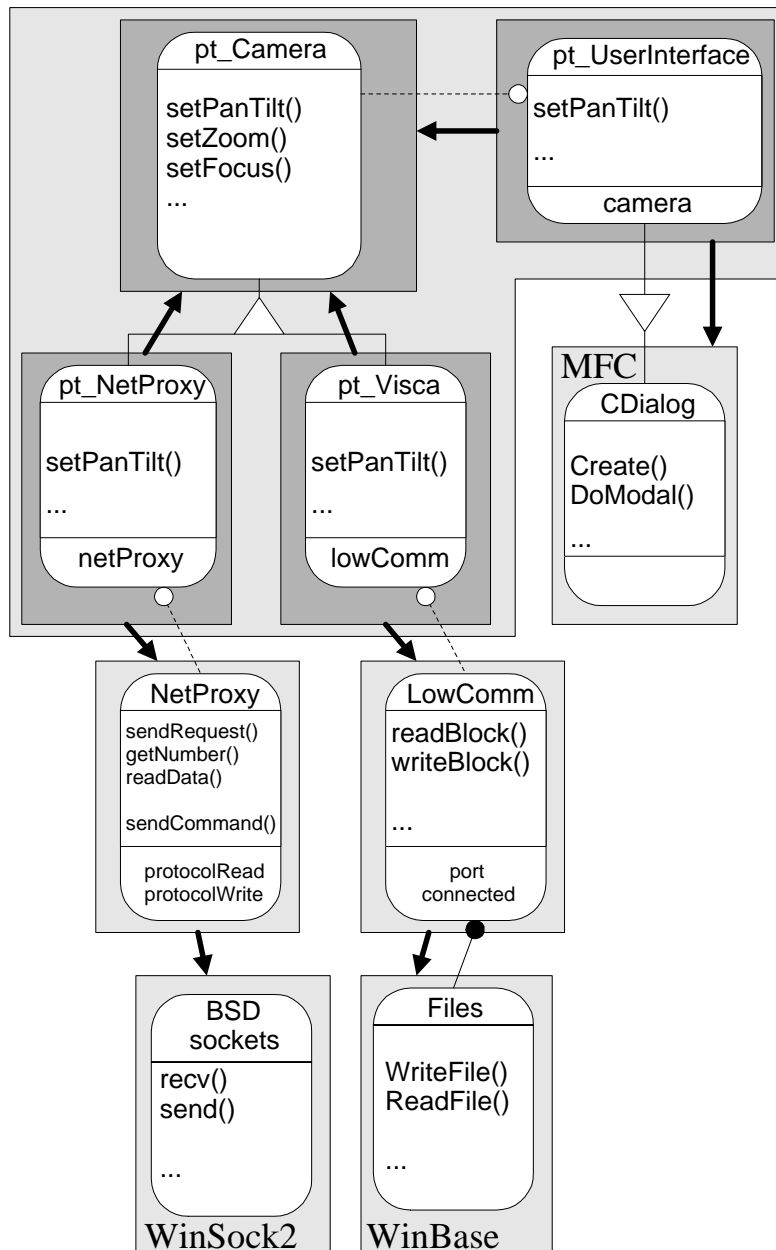
Tablica A.1: Klase izvornog kôda koje su pridružene pojedinim konceptima podsustava za upravljanje kamerom

apstraktna kamera	pt_Camera
stvarna kamera	pt_Visca
udaljena kamera	pt_NetProxy
sučelje	pt_UserInterface
serijska veza	LowComm
TCP/IP veza	NetProxy

²Prva izvedba koncepta serijske veze datira iz 1996. godine.

Složenost testiranja komponente sustava se u [52] procjenjuje kao broj ostalih komponentata o kojima ona ovisi (relacija **ovisi o**), tj. broj komponentata koje se moraju testirati prije nje. Vrlo važno svojstvo predložene arhitekture u tom kontekstu jest izolacija ključne komponente koja sadrži koncept **apstraktne kamere**. Time se testiranje korisničkog kôda može izvesti odvojeno od serijske ili TCP komunikacije, što bitno smanjuje složenost testiranja.

Plod pažljivog odabira koncepata u predloženoj arhitekturi je velika mogućnost ponovne upotrebe svakog od njih u novim aplikacijama i različitim kontekstima. Primarna ciljna platforma se nažalost temelji na MS Windows95 grafičkom korisničkom sučelju jer za korištenje digitalizator slike ne postoji programska podrška za druge platforme. Komponenta **sučelje** je jedina skoro pa inherentno neprenosiva u cijelom podsustavu jer su detalji izvedbe korisničkih sučelja na različitim platformama značajno različiti. Komponenta **serijska veza** bi teoretski mogla (trebala) biti prenosiva, ali iz nekih čudnih razloga podrška za ostvarivanje serijske veze je na MS Windows95 platformi izvedena različito od



Slika A.3: Arhitektura programskog podsustava za upravljanje kamerom.

biblioteka koje se dugi niz godina uspješno koriste na UNIX platformama.

Srećom, komunikacija preko TCP/IP protokola se na svim platformama temelji na BSD socketima pa je izvedba koncepta **TCP/IP veza** u potpunosti prenosiva. Podsustav sa nepromijenjenim komponentama **apstraktna kamera**, **udaljena kamera** i **TCP/IP veza** je uspješno preveden i testiran na platformi sa operacijskim sustavom SunOS 5.5 sa procesorom UltraSparc, čime su ostvarene pretpostavke za prenošenje i ostatka aplikacije.

A.4 Programski podsustav za prijenos slike

U ovom odjeljku, opisuju se zahtjevi, arhitektura i izvedba programskog podsustava za prijenos slike.

A.4.1 Zahtjevi

Programski podsustav za prijenos slike trebao bi definirati apstrakciju postupaka pribavljanja i slanja niza slika u stvarnom vremenu. Slično kao i kod podsustava za upravljanje kamerom, potrebno je pažljivo projektirati sučelja na način da budu neovisna o konkretnom mehanizmu prijenosa. Kao rezultat toga, konačni podsustav će moći biti korišten i testiran transparentno, neovisno o komponentama u kojima je definirana izvedba prijenosa.

Najvažnije je da podsustav ispravno radi sa digitalizatorom slike Comet tvrtke Matrox koji je bio namijenjen za izvedbu ovog rada. Ružna osobina tog digitalizatora (pored ostalih) jest da programska podrška za njega postoji isključivo za platformu sa korisničkim sučeljem MS Windows95. Ružna osobina biblioteka kojima se ostvaruje pristup digitalizatoru je da su one zapravo beta verzija napuštenog projekta i kao takve prilično nestabilne. Stoga je ključni zadatak izgraditi čim veći dio podsustava neovisno o tipu digitalizatora kako bi se taj posao mogao iskoristiti i u budućnosti, kad se nabavi drugi digitalizator. Kao i ranije, želi se izgraditi podsustav koji će se moći nadograđivati bez propagiranja promjena na korisnike.

Pored klasičnog pribavljanja slike sa čvrstog diska, u perspektivi su još zanimljive i primjene u kojima su kamera i digitalizator značajno udaljeni od računala na kojem se odvija obrada. Zato se, kao i ranije, postavlja zahtjev da se postupak pribavljanja i slanja slike, transparentno može odvijati i preko TCP/IP protokola.

Iako se pribavljanje slike nalazi na kritičnom putu obrade u sustavu, ono nije kritična operacija jer je složenost pribavljanja višestruko manja od složenosti višestrukih prolaza kroz sliku prilikom njene obrade. Ipak, potrebno je pripaziti da u tom postupku ne bude suvišnog prenošenja i kopiranja slike. Ponovo se ispostavlja da efikasnost nije kritična te da se može bez straha koristiti elegantan polimorfni pristup.

Krivo bi se mogao donijeti zaključak da je primjena polimorfizma uvijek opravdana. Koncept **slike** je primjer u kojem se polimorfni pristup zbog gubitka efikasnosti morao odbaciti. Naime, slika se može opisati apstraktnim konceptom sa polimorfnim operacijama **setAt** i **getAt** kojima se slikovni element na zadanoj lokaciji slike postavlja odnosno dohvaća. Lijepa strana takvog prikaza je da se algoritmi mogu elegantno zapisati i testirati nad apstraktnom slikom, a funkcionalnost takvih poziva u kontekstu danog formata slike bi definirali specijalizirani koncepti poput **slika u boji**, **siva slika s 8 bita po slikovnom elementu** ili **siva slika s 16 bita po slikovnom elementu**. Ružna strana takvog pristupa je što prevodioc do samog trenutka poziva operacije (npr. **setAt**) *ne može znati* na koju konkretnu izvedbu će izvedba programa preći. Kao posljedica toga, pozvana operacija se ne može izravno proširiti u izvršni kôd, odnosno ne može se zaobići poziv potprograma. Operacija **setAt** se prilikom obrade slike poziva onoliko puta koliko ima slikovnih elemenata u slici (oko 100 000) i koliko god grananje na potprogram bio malen dodatni posao, on se mora odraziti na konačnu performansu programa. To pokazuju i eksperimentalni rezultati. Kada se **slika** opisala generičkom klasom parametriziranom po

tipu slikovnog elementa, prevodioc je dobio mogućnost izravnog proširivanja potprograma koji su definirani nad slikom u izvršni kôd. Rezultat toga je bilo skoro dvostruko ubrzanje programa (oko 40%).

A.4.2 Arhitektura

U opisanim zahtjevima, javljaju se se slijedeći ključni koncepti:

- **izvor slike**³ koji apstrahira pribavljanje slike;
- **ponor slike** koji apstrahira slanje (prosljeđivanje) slike;
- **konkretni izvori slike** koji opisuju pribavljanje slike sa diska, digitalizatora, ...
- **udaljeni izvor slike** koji zahtjeve za pribavljanje slike prenosi preko TCP/IP protokola konkretnom izvoru slike na drugom računalu;
- **konkretni ponori slike** koji opisuju slanje slike na disk, mrežu, ...

Kao i ranije, ideja je da korisnik piše program u terminima poziva apstraktnih koncepata **izvor slike** i **ponor slike**. Ponovo je ključni problem osiguravanje transparentnog pristupa udaljenom digitalizatoru, a rješava se obrascem oblikovanja **Proxy** koji je opisan u prethodnom odjeljku. I u ovom slučaju, objekt koji predstavlja **konkretni izvor slike** zajedno sa programom poslužioem “živi” na računalu na kojeg je digitalizator fizički spojen. Objekt koji predstavlja **udaljeni izvor slike** “živi” na računalu na kojem se obavlja obrada slike, a komunicira sa **konkretnim izvorom slike** preko TCP/IP protokola i programa poslužioaca.

U okviru ovog rada, razrada koncepta **ponor slike** nije potrebna pa zbog vremenskih ograničenja nije obavljena. Ta razrada bi trebala biti jednostavna budući da su već razvijene implementacije slanja slike preko TCP/IP protokola (poslužioac koji se izvodi na računalu na kojeg je digitalizator fizički spojen) i upisa slike u datoteku sa formatom AVI.

A.4.3 Izvedba

Koncept **udaljeni izvor slike** mora imati mogućnost komuniciranja preko TCP/IP protokola. Ovdje je prikladno ponovo iskoristiti komponentu kojom se u podsustavu za upravljanje kamerom opisivao koncept **TCP/IP veza**.

Konkretni izvori slike mogu biti **digitalizator** i **datoteka**, pri čemu su podržani različiti formati datoteka. Time se dobivaju novi koncepti:

- **datoteka** — apstrakcija dohvata slika iz datoteke;
- **AVI datoteka** — dohvata niza slika iz AVI datoteke;
- **višestruka Raster datoteka** — dohvata niza slika iz datoteke sa višestrukim SUN Raster formatom⁴;
- **datoteka s jednom slikom** — apstrakcija dohvata slike iz datoteke sa samo jednom slikom;
- **datoteka BMP** — dohvata niza slika iz BMP datoteke;
- **datoteka Raster** — dohvata jedne slike iz SUN Raster datoteke;
- **digitalizator** — apstrakcija dohvata slike iz digitalizatora;

³Koncept **izvor slike** je prvi identificirao Vladimir Stanisavljević

⁴Izvedba čitanja i pisanja slike u Sun Raster formatu se temelji na izvornom kôdu Zorana Kalafatića

- **digitalizator MCI** — apstrakcija dohvata slike iz digitalizatora preko sučelja MCI;

Arhitektura programskog podsustava za prijenos slike oblikovana je u skladu s međusobnim odnosima izdvojenih koncepata. Pri tome je svakom od koncepata pridružena klasa kao što je prikazano u tablici A.2. Predložena arhitektura podsustava je prikazana na sl. A.4.

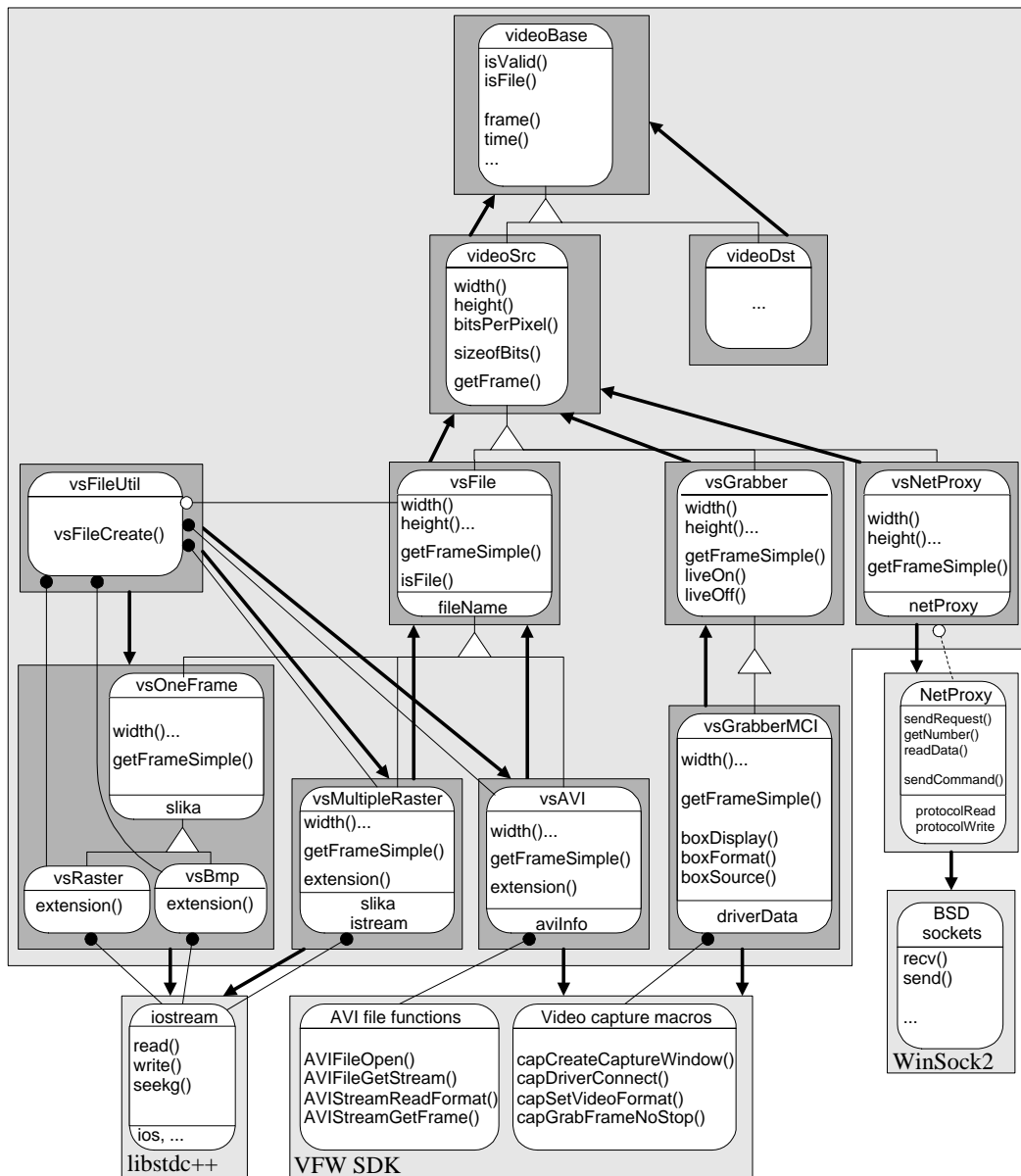
Tablica A.2: Klase izvornog kôda koje su pridružene pojedinim konceptima podsustava za prijenos slike

izvor slike	videoSrc
ponor slike	videoDst
datoteka	vsFile
digitalizator	vsGrabber
udaljeni izvor slike	vsNetProxy
datoteka s jednom slikom	vsOneFrame
datoteka BMP	vsBmp
datoteka Raster	vsRaster
datoteka AVI	vsAVI
višestruka Raster datoteka	vsMultipleRaster
digitalizator MCI	vsGrabberMCI

Slično kao i u podsustavu za upravljanje kamerom, korisnički kôd se lako testira jer je izoliran od izvedbenih detalja konkretnog mehanizma za prijenos slika. Komponente sustava predstavljaju jasne koncepte i stoga su dobri kandidati za ponovnu upotrebu. Korisnici podsustava mogu naravno birati razinu apstrakcije na kojoj će pristupati pojedinim konceptima. Tako je objektu klase `vsSingleBmp` moguće pristupati i preko sučelja klase `vsSingleFrame`, `vsFile` i `videoSource`. Pristup preko apstraktnijeg sučelja pri tome ima prednost da bi ispravno radio i sa drugim konkretnim mehanizmom pribavljanja slika i to bez ponovnog prevođenja. Najosjetljiviji trenutak u takvom polimorfnom pristupu podsustavu za prijenos slike je trenutak odluke koji će se mehanizam odabrati. Ta odluka se donosi u trenutku kreiranja objekta koji predstavlja neki konkretni ili udaljeni **izvor slike**, a temelji se na odluci krajnjeg korisnika podsustava. Pri tome je pogodno korisniku olakšati izbor njegovim svođenjem na tri alternative: slika se može pribavljati konceptima **datoteka**, **digitalizator** i **udaljeni izvor slike**. Ako se jednom odluči uvesti četvrtu alternativu, jedino mjesto u cijeloj aplikaciji u kojem će trebati unijeti izmjenu biti će izbornik za inicijalizaciju gdje bi se ponudila dodatna mogućnost.

Svaki od koncepata u prvoj razini hijerarhijske podjele koncepta **izvor slike** je odgovoran za daljnje, finije grananje postupka kreiranja novog objekta (ako je takvo grananje moguće). Višestruki koncepti u drugoj razini postoje kod koncepta **datoteka** pa je oblikovana dodatna klasa `vsFileUtil`. Jedina metoda te klase `vsFileCreate` kreira i inicijalizira novi objekt prikladnog tipa (`vsAvi`, `vsMultipleRaster`, `vsBmp` ili `vsRaster`) na temelju argumenta koji predstavlja ime datoteke iz koje bi se slike trebale pribavljati. U prvoj izvedbi, `vsFileCreate` obavlja zadatak samo na temelju ekstenzije zadanog imena datoteke, što je sasvim zadovoljavajuće za velik broj primjena. U međuvremenu je razvijena bolja metoda kojom se pročitaju prvih osam bajtova zadane datoteke i onda se postavlja pitanje svakoj od konkretnih klasa da li takvo zaglavlje odgovara formatu kojeg klasa predstavlja. Zbog vremenskih ograničenja, ta metoda nije integrirana u konačnu verziju podsustava za prijenos slike.

Izvedba komponente koja opisuje **datoteka AVI** nažalost nije prenosiva jer se temelji na bibliotekama koje postoje samo na platformama tvrtke Microsoft. Izvedba komponente koja opisuje **digitalizator MCI** također nije prenosiva jer programska podrška za korišteni



Slika A.4: Arhitektura programskog podsustava za prijenos slike.

digitalizator postoji samo za platformu MS Windows95. Izvedba ostalih komponenta bi trebala biti prenosiva međutim prenošenje podsustava na drugu platformu zbog vremenskih ograničenja još nije obavljeno.

Dodatak B

Uputstva za instalaciju i korištenje razvijenih programa

U ovom dodatku, daju se kratka uputstva za instalaciju i korištenje razvijene programske podrške. Tokom razvoja provedeno je dosta eksperimenata sa različitim sučeljima ključnih koncepata. Rezultat toga je da konačne arhitekture programskih podsustava za upravljanje digitalizatorom i pokretnom kamerom nisu kompatibilne sa ostatkom aplikacije zbog naprednijeg sučelja ključnog koncepta za opis slike.

Stoga se uz rad prilažu dvije aplikacije, od kojih prva obuhvaća funkcionalnost opisanu u glavnom dijelu rada (izlučivanje linija, pronalaženje dominantnog nedogleda vodoravnih bridova scene i lokalizacijska procedura), dok druga sadrži konačnu izvedbu podsustava za upravljanje digitalizatorom i pokretnom kamerom (dodatak A).

B.1 Program za obavljanje lokalizacijske procedure

Aplikacija je pisana za platforme Windows95 i NT tvrtke Microsoft. Njena instalacija je vrlo jednostavna i svodi se na kopiranje datoteke sa izvršnim kôdom na računalo na kojem se želi obavljati obrada.

Korištenje aplikacije je također jednostavno. Prije svega, potrebno je odabrati izvor slike — u ovoj verziji tog podsustava, može se birati između pribavljanja slike izravno iz digitalizatora ili iz datoteke koja sadrži jednu ili više prethodno digitaliziranih slika. Taj odabir se vrši izbornicima

Video gource → Open →

Ako se odabere digitalizator, aplikacija će pokušati uspostaviti vezu sa programom za upravljanje digitalizatorom preko MCI sučelja. U slučaju uspješne inicijalizacije, unutar aplikacije će se otvoriti prozor sa “živom” slikom, dok će u suprotnom biti dojavljena dijagnostička poruka. Program za upravljanje digitalizatorom je jedva funkcionalan i povremeno se ne da pravilno iskonfigurirati. U tim slučajevima, za nastavak rada je potrebno resetirati sklopovlje, a jedini poznati način za to je sklopovsko resetiranje računala.

Ako se odabere pribavljanje slike iz datoteke, otvorit će se izbornik za njen odabir. Podržani su formati AVI, BMP i SUN Raster, a ekstenzija datoteke mora odgovarati njenom formatu (`avi`, `bmp`, `rfs`). Ako se radi o datoteci sa višestrukim slikama, moguće se pozicionirati na pojedini okvir uz pomoć naredbe izbornika:

Video gource → Set Frame.

Inicijalizacija podsustava za upravljanje kamerom se obavlja pokretanjem prijateljskog korisničkog sučelja, izbornikom

View → Camera control.

Budući da protokol VISCA ne predviđa komunikaciju bržu od 9600 bps, ta operacija traje više sekundi.

Obrada se pokreće iz izbornika

Process → ... ,

pri čemu je moguće obavljati obradu na jednoj ili više slika. Na temelju jedne slike, može se zatražiti izlučivanje pravocrtnih segmenata

Process → Single Image → Lines,

ili pronalaženje nedogleda vodoravnih bridova scene

Process → Single Image → Vanishing points.

Izlučivanje pravocrtnih segmenata je moguće obaviti i bez inicijalizacije programskog podsustava za upravljanje kamerom. Međutim ako se nad slikama iz datoteke želi obaviti i pronalaženje nedogleda, prethodno je potrebno “reći” sustavu nagib i širinu vidnog polja (u stupnjevima) kamere kojom su dobivene slike iz datoteke. To se može obaviti dodavanjem ili mijenjanjem redaka

`tiltAngleDegs=xx`

odnosno

`horAngleDegs=yy`

u datoteci `mr.ini` koja se nalazi u istom direktoriju kao i izvršna datoteka `mr.exe`.

Lokalizacijska procedura se pokreće izbornikom

Process → Look around → Live.

Da bi se ona pokrenula, potrebno je prethodno ručno inicijalizirati podsustave za upravljanje kamerom i digitalizatorom. Kako korištena kamera ima ograničenu mogućnost postavljanja kuta zakreta ($\pm 100^\circ$), lokalizacija se obavlja u dva koraka, za “prednjih” i “stražnjih” 180° . Tako nakon obrade prednjeg polukruga, aplikacija traži od korisnika (u krajnjoj primjeni, to bi bio robot), da okrene kameru za 180° kako bi se mogao obraditi i stražnji polukrug.

Nakon obavljanja prvog dijela lokalizacijske procedure, dobiveni hipotetski nedogledi u koordinatnom sustavu robota se upisuju u datoteku `c:/temp/out.txt`. Ti hipotetski nedogledi mogu biti naknadno podvrgnuti samo drugom koraku lokalizacijske procedure (grupiranju) odabirom izbornika

Process → Look around → From file.

Prvi korak procedure se mora obaviti na MS Windows95 platformi zbog programske podrške korištenog digitalizatora. Drugi korak je međutim bolje napraviti na MS NT platformi jer se format slike u kojem se prikazuju konačni rezultati (16 bita po slikovnom elementu) ne da prikazati na Windows95 platformi.

B.2 Konačna izvedba upravljanja digitalizatorom i kamerom

Ovaj program proširuje funkcionalnost upravljanja digitalizatorom i pokretnom kamerom iz prethodne aplikacije mogućnošću pristupa udaljenoj kameri TCP/IP protokolom.

Da bi se program mogao uspješno izvršavati, potrebno je instalirati i pokrenuti po jedan program poslužioc za svaki servis koji se želi koristiti (pristup kameri i pristup digitalizatoru). Program poslužioc je pisan za platforme Windows95 i NT tvrtke Microsoft. Njegova instalacija se svodi na kopiranje izvršne datoteke `netserver.exe` na udaljeno računalo na koje su spojeni kamera i digitalizator (sl. A.1). Pokretanje poslužioca se izvodi iz ljuške, a jedini argument u komandnoj liniji označava željeni mrežni ulaz (*engl.* port) na kojem će poslužioc očekivati zahtjeve. Trenutak zauzimanja mrežnog ulaza je osjetljiv jer se teoretski može dogoditi da je traženi ulaz već zauzela druga aplikacija. Kako udaljeni klijenti očekuju da će server oslušivati unaprijed određeni ulaz, tada je potrebno zatvoriti ili prekonfigurirati konfliktnu aplikaciju, ili izabrati neki drugi mrežni ulaz. Na Windows95 platformi nije moguće pokrenuti poslužioc u pozadini pa je zato, u slučaju kad se želi upravljati i digitalizatorom i pokretnom kamerom, potrebno pokrenuti dvije ljuške (`4nt.exe`) kako bi se iz njih moglo pokrenuti dva mrežna poslužioca.

Instalacija glavnog programa je trivijalna i svodi se na kopiranje datoteke sa izvršnim

kôdom aplikacije (`mr2.exe`) na udaljeno računalo. Glavni program je isproban na NT platformi tvrtke Microsoft. Slično kao i u prethodnoj aplikaciji, na samom početku je potrebno inicijalizirati podsustave za pribavljanje slike i upravljanje kamerom. Ovdje se međutim pruža dodatna mogućnost upravljanja udaljenim sklopovljem preko protokola TCP/IP. Pri tome se pokušava uspostaviti komunikacija sa mrežnim ulazima udaljenih računala u skladu sa podacima

```
netServer=racunalo.domena
```

odnosno

```
netPort=xxxx.
```

Ti podaci o računalu i mrežnom ulazu na kojima “čekaju” poslužioци za upravljanje digitalizatorom i kamerom su upisani u sekcijama VIDEO odnosno VISCA tekstualne datoteke `mr2.ini` koja se nalazi u istom direktoriju kao i datoteka sa izvršnim kôdom programa.

U glavnom programu pruža se mogućnost upravljanja kamerom, dohvata slike iz digitalizatora i njenog spremanja na čvrsti disk — izbornik

Image → Save.

Pored toga, omogućen je i periodički dohvat slike kojim se dobiva kontinuirano osvježavanje. Zавodska mreža omogućuje prijenos slike najvećeg formata kojeg podržava digitalizator ($640 \times 480 \times 24$) u vremenu od 2,2 s što odgovara propusnosti od oko 3,5 Mbps. Odatle proizlazi da bi za postizanje kvalitete televizijskog prijenosa bez korištenja kompresije bilo potrebno ubrzati mrežnu infrastrukturu 50 puta.

Dodatak C

Izvorni i izvršni kôd

Konačnoj verziji rada će biti priložen i CD ROM sa izvornim te izvršnim kôdom razvijenih programa te slikama na kojima su obavljani eksperimenti.

Bibliografija

- [1] J.T. Tou i R.C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley, 1974.
- [2] D. Waltz, Understanding line drawings of scenes with shadows, u P.H. Winston, ur., *Psychology of Computer Vision*, Computer Science, str. 19–91, McGraw-Hill, 1975.
- [3] P.H. Winston, *The Psychology of Computer Vision*, Computer Science, McGraw-Hill, 1975.
- [4] Jack Sklansky, On the hough transform for curve detection, *IEEE Transactions on Computers*, sv. c-27(10):923–926, listopad 1978.
- [5] John Canny, Finding edges and lines in images, Tech. Rep., Laboratory of Artificial Intelligence, Massachusetts Institute of Technology, Cambridge, USA, 1983.
- [6] P.J. Besl i R.C. Jain, Three-dimensional object recognition, *ACM Computing Surveys*, str. 75–145, 1985.
- [7] J.B.Burns, A.R.Hanson i E.M.Riseman, Extracting straight lines, *IEEE Transactions on Pattern recognition and Machine Intelligence*, sv. 8(7):425–456, srpanj 1986.
- [8] S.P. Liou i R.C. Jain, Road following using vanishing points, *Computer Vision Graphics and Image Processing*, sv. 39(1):116–130, srpanj 1987.
- [9] Roger Y. Tsai, A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf cameras and lenses, *IEEE Journal of Robotics and Automation*, sv. 3(4):323–345, kolovoz 1987.
- [10] J. Aloimonos, I. Weiss i A. Bandyopadhyay, Active vision, *International Journal of Computer Vision*, sv. 1:333–356, 1988.
- [11] Ruzena K. Bajcsy, Active perception, *Proceedings of the IEEE*, sv. 76(8):996–1005, kolovoz 1988.
- [12] Mario Bertero i Tomaso A. Poggio, Ill-posed problems in early vision, *Proceedings of the IEEE*, sv. 76(8):869–889, kolovoz 1988.
- [13] J. Illingworth i J.V. Kittler, A survey of the hough transform, *Computer Vision Graphics and Image Processing*, sv. 44(1):87–116, listopad 1988.
- [14] Shigeki Ishikawa, Hideki Kuwamoto i Shinji Ozawa, Visual navigation of an autonomous vehicle using white line recognition, *IEEE Transactions on Pattern recognition and Machine Intelligence*, sv. 10(5):743–749, rujana 1988.
- [15] Kokichi Sugihara, Some location properties for robot navigation using a single camera, *Computer Vision, Graphics and Image processing*, sv. 42(1):112–129, travanj 1988.
- [16] Eric P. Krotkov, *Active computer vision by cooperative focus and stereo*, Springer-Verlag, 1989.
- [17] Darko Veljan, *Kombinatorika s teorijom grafova*, Školska knjiga, Zagreb, Hrvatska, 1989.
- [18] R.C. Arkin i R.R. Murphy, Autonomous navigation in a manufacturing environment, *IEEE Transactions on Robotics and Automation*, sv. 6(4):445–454, kolovoz 1990.
- [19] Weng-Nung Lie i Yung-Chang Chen, Robust line-drawing extraction for polyhedra using weighted polarized hough transform, *Pattern Recognition*, sv. 23(3/4):261–274, 1990.
- [20] B. Brillault-O'Mahony, New method for vanishing point detection, *Computer Vision Graphics and Image Processing – Image Understanding*, sv. 54(2):289–300, rujana 1991.

- [21] J.R.Parker, Gray level thresholding in badly illuminated images, *IEEE Transactions on Pattern recognition and Machine Intelligence*, sv. 13:813–819, 1991.
- [22] K.I. Kanatani, Computational projective geometry, *Computer Vision Graphics and Image Processing – Image Understanding*, sv. 54(3):333–348, studeni 1991.
- [23] John J. Leonard i Hugh F. Durrant-Whyte, Mobile robot localization by tracking geometrical beacons, *IEEE Transactions on Robotics and Automation*, sv. 7(3), lipanj 1991.
- [24] M.J. Swain i M.Stricker, Promising directions in active vision, Tech. Rep. 91-27, University of Chicago, Department of Computer Science, 1100 E. 58th street, studeni 1991.
- [25] Margrit Betke, *Algorithms for exploring an unknown graph*, magistrski rad, Massachusetts Institute of Technology, Cambridge, Department of Electrical Engineering and Computer Science, veljača 1992.
- [26] James O. Coplien, *Advanced C++: Programming Styles and Idioms*, Addison-Wesley, Reading, Massachusetts, USA, 1992.
- [27] Yong K. Hwang i Narendra Ahuya, Gross motion planning — a survey, *ACM computing surveys*, sv. 24(3), rujan 1992.
- [28] David W. Jacobs, *Recognizing 3-D Objects Using 2-D Images*, doktorska disertacija, Massachusetts Institute of Technology, Cambridge, Laboratory of Artificial Intelligence, 1992.
- [29] Takeshi Shakunaga, 3-d corridor scene modeling from a single view under natural lighting conditions, *IEEE Transactions on Pattern recognition and Machine Intelligence*, sv. 14(2):293–298, veljača 1992.
- [30] M. Straforini, C. Coelho, M. Campani i V. Torre, The recovery and understanding of a line drawing from indoor scenes, *IEEE Transactions on Pattern recognition and Machine Intelligence*, sv. 14(2):298–303, veljača 1992.
- [31] Ching-Cheng Wang, Extrinsic calibration of a vision sensor mounted on a robot, *IEEE Transactions on Robotics and Automation*, sv. 8(2):161–176, travanj 1992.
- [32] S. Atiya i G.D. Hager, Real-time vision-based robot localization, *IEEE Transactions on Robotics and Automation*, sv. 9(6):785–800, prosinac 1993.
- [33] O.D. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, 1993.
- [34] X. Lebegue i J.K. Aggarwal, Significant line segments for an indoor mobile robot, *IEEE Transactions on Robotics and Automation*, sv. 9(6):801–815, prosinac 1993.
- [35] P. L. Palmer, J. Kittler i M. Petrou, A hough transform algorithm with a 2d hypothesis kernel, *Computer Vision Graphics and Image Processing – Image Understanding*, sv. 58:221–234, 1993.
- [36] Nagewara S.V. Rao, Weimin Shi i S. Sitharama Iyengar, Robot navigation in unknown terrains: Introductory survey of non-heuristic algorithms, Tech. Rep. ORNL/TM-12410, Oak Ridge National Laboratory, srpanj 1993.
- [37] R. Schuster, N. Ansari i A. Bani-Hashemi, Steering a robot with vanishing points, *IEEE Transactions on Robotics and Automation*, sv. 9(4):491–498, travanj 1993.
- [38] Gianluca Foresti, Vittorio Murino, Carlo S. Regazzoni i Gianni Vernazza, Grouping of rectilinear segments by the labeled hough transform, *Computer Vision Graphics and Image Processing – Image Understanding*, sv. 58(3):22–42, studeni 1994.
- [39] E. Lutton, H. Maitre i J. Lopez-Krahe, Contribution to the determination of vanishing points using hough transform, *IEEE Transactions on Pattern recognition and Machine Intelligence*, sv. 16(4):430–438, travanj 1994.
- [40] Yoav Shoham, *Artificial Intelligence Techniques in Prolog*, Morgan Kaufmann Publishers, San Francisco, California, 1994.
- [41] Margrit Betke, *Learning and Vision Strategies for Robot Navigation*, doktorska disertacija, Massachusetts Institute of Technology, Cambridge, lipanj 1995.

- [42] Erich Gamma, Richard Helm, Ralph Johnson i John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented systems*, Professional Computing Series, Addison-Wesley, Reading, Massachusetts, USA, 1995.
- [43] J. Hynoski i H. R. Wu, Active vision - a survey of the field and research directions, Tech. Rep. 95-04, Monash University, Faculty of Computing and Information Technology, Department of Robotics and Digital Technology, Clayton, Australia, svibanj 1995.
- [44] Akio Kosaka i Juiyao Pan, Purdue experiments in model-based vision for hallway navigation, u *Proceedings of Workshop on Vision for Robots, IROS'95 Conference*, Pittsburgh, PA, 1995.
- [45] G.F. McLean i D. Kotturi, Vanishing point detection by line clustering, *IEEE Transactions on Pattern recognition and Machine Intelligence*, sv. 17(11):1090–1095, studeni 1995.
- [46] Stuart J. Russel i Peter Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, New Jersey, 1995.
- [47] Zhengyou Zhang, Estimating motion and structure from correspondence of line segments between two perspective images, *IEEE Transactions on Pattern recognition and Machine Intelligence*, sv. 17(12):1129–1139, prosinac 1995.
- [48] Frederick P. Brooks Jr., *The Mythical Man-Month: Essays on Software Engineering*, Addison Wesley Longman, Reading, Massachusetts, USA, 2 izd., 1995 (1975).
- [49] Steve Maguire, *Kako upravljati razvojnim procesom*, ZNAK, Zagreb, Hrvatska, 2 izd., 1995 (1994).
- [50] J. Borenstein, H. R. Everett i L. Feng, *Navigating Mobile Robots: Sensors and Techniques*, A. K. Peters, Ltd., Wellesley, MA, siječanj 1996.
- [51] David W. Jacobs, Robust and efficient detection of salient convex groups, *IEEE Transactions on Pattern recognition and Machine Intelligence*, sv. 18(1):23–37, 1996.
- [52] John Lakos, *Large Scale C++ software Design*, Professional Computing Series, Addison-Wesley, Reading, Massachusetts, USA, 1996.
- [53] M.D.Heath, *A robust visual method for assessing the relative performance of edge detection algorithms*, magistarski rad, College of Engineering, University of South Florida, Department of Computer Science and Engineering, prosinac 1996.
- [54] Roger Mohr i Bill Triggs, Projective geometry for image analysis, srpanj 1996, a tutorial given at the International Symposium of Photogrammetry and Remote Sensing, Vienna.
- [55] Pietro Parodi, The complexity of understanding line drawings of origami scenes, *International Journal of Computer Vision*, sv. 18(2):139–170, 1996.
- [56] Pietro Parodi i Giulia Piccioli, 3d shape reconstruction by using vanishing points, *IEEE Transactions on Pattern recognition and Machine Intelligence*, sv. 18(2):211–217, 1996.
- [57] Dean Pomerleau i Todd Jochem, Rapidly adapting machine vision for automated vehicle steering, *IEEE Expert*, sv. 11(2), travanj 1996.
- [58] Kristian T. Simsarian, Thomas J. Olson i N. Nandhakumar, View-invariant regions and mobile robot self-localization, *IEEE Transactions on Robotics and Automation*, sv. 12(5), listopad 1996.
- [59] R. Talluri i J.K Aggarwal, Mobile robot self-location using model-image feature correspondence, *IEEE Transactions on Robotics and Automation*, sv. 12(1), veljača 1996.
- [60] S. Baten, M. Lutzeler, E.D. Dickmanns, R. Mandelbaum i P.J. Burt, Techniques for autonomous, off-road navigation, *IEEE Intelligent Systems*, sv. 13(6), prosinac 1997.
- [61] M. Betke i L. Gurvits, Mobile robot localization using landmarks, *IEEE Transactions on Robotics and Automation*, sv. 13(2):251–263, travanj 1997.
- [62] J. Borenstein, H. R. Everett, L. Feng i D. Wehe, Mobile robot positioning: Sensors and techniques, *Journal of Robotic Systems*, sv. 14(4):231–249, travanj 1997.
- [63] J. Borenstein i I. Ulrich, The guidecane - a computerized travel aid for the active guidance of blind pedestrians, u *Proceedings of the IEEE International Conference on Robotics and Automation*, str. 1283–1288, Albuquerque, NM, travanj 1997.

- [64] S. J. Dickinson, H. I. Christensen, J. K. Tsotsos i G. Olofsson, Active object recognition integrating attention and viewpoint control, *Computer Vision and Image Understanding*, sv. 67(3):239–260, 1997.
- [65] H.S. Dulimarta i A.K. Jain, Mobile robot localization in indoor environment, *Pattern Recognition*, sv. 30(1):99–111, siječanj 1997.
- [66] Sara Hedberg, Ai coming of age: Nasa uses ai for autonomous space exploration, *IEEE Expert*, sv. 12(3), lipanj 1997.
- [67] N.I. Katevas, N.M. Sgouros, S.G. Tzafestas, G. Papakonstantinou, P. Beattie, J.M. Bishop, P. Tsanakas i D. Koutsouris, The autonomous mobile robot senario: a sensor aided intelligent navigation system for powered wheelchairs, *IEEE Robotics and Automation magazine*, sv. 4(4), prosinac 1997.
- [68] Jesse Liberty, *Clouds to Code*, WROX Press, Chicago, Illinois, USA, 1997.
- [69] Claus B. Madsen i Henrik I. Christensen, A viewpoint planning strategy for determining true angles on polyhedral objects by camera alignment, *IEEE Transactions on Pattern recognition and Machine Intelligence*, sv. 19(2):158–163, 1997.
- [70] A. Mandow, J.M. Gomez-de Gabriel, J.L. Martinez, V.F. Munoz, A. Ollero i A. Garcia-Cerezo, The autonomous mobile robot aurora for greenhouse operation, *IEEE Robotics and Automation magazine*, sv. 3(4), prosinac 1997.
- [71] Michael M. Marefat, Liwei Wu i Cristopher C. Yang, Gaze stabilization in active vision-1 vergence error extraction, *Pattern Recognition*, sv. 30(11):1829–1842, 1997.
- [72] P. L. Palmer, J. Kittler i M. Petrou, An optimizing line finder using a hough transform algorithm, *Computer Vision Graphics and Image Processing – Image Understanding*, sv. 67(1):1–23, srpanj 1997.
- [73] James R. Parker, *Algorithms for Image Processing and Vision*, Johnw Wiley & Sons, Inc., New York, U.S.A., 1997.
- [74] Changqing Wang i David R. Musser, Dynamic verification of C++ generic algorithms, *IEEE Transactions on Software Engineering*, sv. 23(5), svibanj 1997.
- [75] Mark C.K. Yang, Jong-Sen Lee, Cheng-Chang Lien i Chung-Lin Huang, Hough transform modified by line connectivity and line thickness, *IEEE Transactions on Pattern recognition and Machine Intelligence*, sv. 19(8):905–910, kolovoz 1997.
- [76] Matthew H. Austern, *Generic Programming and the STL: Using and Extending the C++ Standard*, Professional Computing Series, Addison-Wesley, Reading, Massachusetts, USA, 1998.
- [77] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner i S. Thrun, The interactive museum tour-guide robot, u *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, Wisconsin, 1998.
- [78] J. C Clarke, Visual beacons for robot localisation, u *Proc. of the EU joint TMR Networks Workshop on Computer Vision and Mobile Robotics*, str. 9–14, 1998.
- [79] A.J. Davison i D.W. Murray, Mobile robot localisation using active vision, u *Proceedings of the 5th European Conference on Computer Vision*, sv. II, str. 809–825, Freiburg, Germany, lipanj 1998.
- [80] Mike D. Heath, Sudeep Sarkar, Thomas Sanocki i Kevin Bowyer, Comparison of edge detectors: A methodology and initial study, *Computer Vision and Image Understanding*, sv. 69(1), siječanj 1998.
- [81] Lana Lipošinović, *Nacrtna geometrija — udžbenik za građevinske tehničke škole*, Element, Zagreb, Hrvatska, 1998.
- [82] Jacob Matijevic, Autonomous navigation and the sojourner microrover, *Science*, sv. 280(5362):454–455, travanj 1998.
- [83] Jun Miura i Katsushi Ikeuchi, Task-oriented generation of visual sensing strategies in assembly tasks, *IEEE Transactions on Pattern recognition and Machine Intelligence*, sv. 20(2):126–138, 1998.

- [84] Nagewara S. V. Rao, A fusion method that performs better than best sensor, u *Proceedings of the First International Conference on Multisource-Multisensor Information Fusion*, Las Vegas, Nevada, srpanj 1998.
- [85] Siniša Šegvić, Robust extraction of line segments from colour images by an iterative weighted polarised hough transform, u *Proceedings of 21th International Convention MIPRO '98*, sv. 2, str. 35–38, Opatija, Hrvatska, svibanj 1998.
- [86] Robert Sim, *Mobile robot localisation using learned landmarks*, magistrski rad, McGill University, Montreal, Department of Computer Science, srpanj 1998.
- [87] Camillo J. Taylor i David J. Kriegman, Vision-based motion planning and exploration algorithms for mobile robots, *IEEE Transactions on Robotics and Automation*, sv. 14(3), lipanj 1998.
- [88] Drago Torkar, *Prijemanje in premikanje predmetov v neumerjenih in šibko umerjenih sistemih aktivnega stereo vida*, doktorska disertacija, Fakulteta za elektrotehniko, Univerza v Ljubljani, srpanj 1998.
- [89] Eric Weisstein, *The CRC Concise Encyclopedia of Mathematics*, CRC Press, Boca Raton, Florida, 1998.
- [90] R. Araujo i A.T. de Almeida, Learning sensor-based navigation of a real mobile robot in unknown worlds, *IEEE Transactions on Systems, Man and Cybernetics, Part B*, sv. 29(2):164–178, travanj 1999.
- [91] J. Neira, J.D. Tardos, J. Horn i G. Schmidt, Fusing range and intensity images for mobile robot localization, *IEEE Transactions on Robotics and Automation*, sv. 15(1):76–84, veljača 1999.
- [92] Siniša Šegvić, Using vanishing points for purposive adjustment of viewing direction, u *Proceedings of the IEEE International Symposium on Industrial Electronics*, sv. 1, str. 199–204, Bled, Slovenija, srpanj 1999.
- [93] Siniša Šegvić i Slobodan Ribarić, Self-localisation procedure for autonomous navigation through corridors, u *Proceedings of the Eighth Electrotechnical and Computer Science Conference*, sv. A, str. 275–278, Portorož, Slovenija, rujan 1999.
- [94] Jefferey Shufelt, Performance evaluation and analysis of vanishing point detection techniques, *IEEE Transactions on Pattern recognition and Machine Intelligence*, sv. 21(3):282–288, ožujak 1999.
- [95] Robert Sim i Gregory Dudek, Learning and evaluating visual features for pose estimation, u *Proceedings of the International Conference on Computer Vision*, str. 1217–1222, Kerkyra, Greece, rujan 1999.

Kratki sadržaj

Samostalno određivanje položaja u okolišu (lokalizacija) je preduvjet za postizanje samostalne navigacije. Predložen je pristup lokalizaciji samostalnog robota u sustavu hodnika, koji se temelji na projekcijskoj geometriji i jednookom aktivnom računarskom vidu. Količina korisne informacije za snalaženje koja je sadržana u slici razmatranog okoliša jako ovisi o položaju i orijentaciji kamere kojom je slika pribavljena. Predložena lokalizacijska procedura stoga, pored postupaka za obradu slike, propisuje i postupke za njihovo pribavljanje. Tako se smjerovi pružanja hodnika odnosno mogući smjerovi napredovanja određuju obradom niza slika pribavljenih iz zadanog položaja robota, za različite smjerove gledanja kamere. Obrada pribavljenih slika se sastoji od traženja hipotetskih točaka nedogleda izlučenih pravocrtnih segmenata, u skladu sa principima projekcijske geometrije. Konačno, smjerovi pružanja hodnika se određuju grupiranjem pronađenih nedogleda u koordinatnom sustavu robota. Pojedinačni postupci i cjelokupna lokalizacijska procedura su isprobani na stvarnim scenama.

Ključne riječi:

Računarski vid, tumačenje trodimenzionalnih scena, projekcijska geometrija, aktivna percepcija, rubna segmentacija, Houghova transformacija, pravocrtni segmenti, nedogledi, ortogonalna triedarska struktura.

Summary

The capability of a mobile robot to determine its position in the environment (self-localisation) is a prerequisite for achieving autonomous navigation. The proposed approach to self-localisation in a system of corridors is based on the projective geometry and active computer vision using a single camera. The quantity of navigationally useful information contained in an image of the considered environment strongly depends on position and orientation of the camera used to acquire the image. The presented self-localisation procedure therefore specifies both image processing algorithms and methods for controlling the image acquisition process. Thus, the directions of longitudinal corridor axes in the environment (or, equivalently, the viable directions of advancement) are determined by processing the sequence of images acquired from the given position, with different orientations of the camera. The processing of acquired images consists of finding hypothetical vanishing points of extracted line segments, in accordance with principles of the projective geometry. Finally, the viable advancement directions in the corridor are determined by grouping of obtained vanishing points in the coordinate system of the robot. Both isolated techniques and the whole self-localisation procedure are tested on real scenes.

Keywords:

Computer vision, 3D scene analysis, projective geometry, active perception, edge segmentation, Hough transform, straight line segments, vanishing points, orthogonal trihedral structure.

Životopis

Siniša Šegvić rođen je 1971. godine u Splitu. Matematičku gimnaziju je završio u Zadru 1990. godine. Iste godine upisao je studij na Elektrotehničkom fakultetu Sveučilišta u Zagrebu i diplomirao u veljači 1995. godine. Za izuzetan uspjeh, i na trećoj i na četvrtoj godini studija primio je Priznanje “Josip Lončar”. Od 1996. godine zaposlen je na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva u Zagrebu kao mlađi asistent. Područja njegovog znanstvenog i istraživačkog interesa uključuju aktivni i trodimenzionalni računarski vid te obradu slike. Samostalno odnosno kao koautor objavio je više članaka na međunarodnim znanstvenim skupovima.