

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1015

**Pronalaženje prohodnog tla
stereoskopskim računalnim vidom**

Slavko Grahovac

Zagreb, veljača 2015.

Zahvaljujem svima koji su me podržavali u izradi diplomskog rada, te posebno mentoru prof.dr.sc. Šegviću na stručnoj pomoći i savjetima

POPIS SLIKA

2.1. Projekcija točke u prostoru na stereo slike	4
2.2. Koordinatni sustavi svijeta i kamere	4
2.3. Koordinatni sustavi kamere i slike	5
2.4. Epipolarna ravnina	6
2.5. Vektori \mathbf{q}_r i \mathbf{q}_l	7
2.6. Rektificirane slikovne ravnine	8
2.7. Proces triangulacije	9
2.8. Oblak točaka	11
2.9. Smjerovi agregacije cijene kod poluglobalnog podudaranja	13
2.10. Mapa dispariteta izgrađena algoritmom poluglobalnog podudaranja	13
2.11. Karta visina sa ćelijama grupiranim po koordinatnim osima	15
2.12. Karta visina sa ćelijama grupiranim po stupcima i disparitetu	16
2.13. Karta visina sa ćelijama točaka grupiranih po stupcima i dubinskoj koordinati	16
2.14. Karta visina sa slike iz KITTI ispitnog skupa	17
2.15. Koraci u detekciji rubova	18
2.16. Akumulacijsko polje Houghove transformacije	19
2.17. Model prometne scene	20
2.18. Oblak točaka u modelu prometne scene	21
2.19. Prikaz funkcije f_{rubnik}	22
2.20. Prikaz funkcije g_{rubnik}	24
2.21. Grafički model	25
2.22. Usmjereni grafički model	26
2.23. Neusmjereni grafički model	27
2.24. Struktura uvjetnog slučajnog polja na slici	28
2.25. Vjerojatnost izmjerene visine h_i uz danu oznaku l_i i parametre Θ	30
2.26. Funkcija g_{rubnik}	31

2.27. Modeliranje potencijala bridova, vjerojatnost različitih oznaka (crvena crta) i istih oznaka (plava crta) susjednih ćelija u ovisnosti o razlici njihove visine	32
2.28. Ažuriranje poruka u algoritmu propagacije vjerojatnosti	32
3.1. Projekcija točke u prostoru na stereo slike	34
3.2. Putanja koordinatnog sustava automobila	35
3.3. OpenCV logo	36
3.4. NLopt logo	37
3.5. Oblak točaka	38
4.1. Problem nedovoljne osvijetljenosti okoline	41
4.2. Koraci u rekonstrukciji rubnika analizom gradijenta karte visina . . .	42
4.3. Lažno pozitivne rekonstrukcije	42
4.4. Primjer problema površine susjedne pločniku	42
4.5. Prepreke označene kao rubnik	43
4.6. Uspješna detekcija	43
4.7. Uspješna detekcija	43
4.8. Primjer označenog rubnika	44
4.9. Početno označavanje granice interpolacijom pravca	44
4.10. Krivulje sa manjim kubnim i kvadratnim članovima	45
4.11. Krivulje sa manjim kubnim i kvadratnim članovima	45
4.12. Dobro rekonstruiran rubnik uz problem nejednolike visine pločnika .	46
4.13. Loše označen rubnik zbog male visinske razlike pločnika i ceste . . .	46
4.14. Precizna rekonstrukcija rubnika	46
4.15. Precizna rekonstrukcija rubnika	46
4.16. Estimirane površine	47
4.17. Oznake ćelija karte visina, bijelo za pločnik, crno za cestu	47
4.18. Razlika ručno i programski rekonstruiranog rubnika	47

SADRŽAJ

Popis slika	iv
1. Uvod	1
2. Tehnike stereo vida i algoritmi detekcije rubnika	3
2.1. Rekonstrukcija strukture scene	3
2.1.1. Stereoskopski računalni vid	3
2.1.2. Stvaranje mape dispariteta i oblaka točaka	9
2.2. Predstavljanje scene kartom visina	13
2.2.1. Reprzentacije oblaka točaka	14
2.3. Rekonstrukcija rubnika analizom gradijenata karte visina	17
2.3.1. Cannyjev detektor rubova	18
2.3.2. Houghova transformacija	19
2.4. Rekonstrukcija rubnika korištenjem uvjetnog slučajnog polja	20
2.4.1. Estimiranje parametara modela prometnice	20
2.4.2. Uvjetno slučajno polje kao grafički model	25
3. Programska implementacija i ispitni skupovi	34
3.1. Ispitni skupovi	34
3.2. Vanjske biblioteke	36
3.2.1. Biblioteka OpenCV	36
3.2.2. Biblioteke optimizacije parametara površina	36
3.2.3. Biblioteka za grafičke modele	37
3.2.4. Biblioteka PCL	37
3.3. Programska implementacija	38
4. Eksperimentalni rezultati	40
4.1. Rezultati rekonstrukcije rubnika analizom gradijenata karte visina	41
4.2. Rezultati rekonstrukcije rubnika korištenjem uvjetnog slučajnog polja	44

5. Zaključak	49
Literatura	51

1. Uvod

Sustavi za pomoć u vožnji su bitno polje u razvoju primjene računalnog vida na probleme analize okoline. Rješenja koja pružaju ovakvi sustavi mogu imati ulogu u povećanju sigurnosti u prometu te olakšati vozaču poštivanje prometnih pravila. Dio sustava za pomoć u vožnji čini i analiza prohodnosti ceste i njezine okoline. Kako bismo spriječili vožnju neprilagođenu uvjetima na cesti potrebno je detektirati rubnike, vozila i ostale prepreke na cesti. Problem detekcije rubnika zahtijeva specifične podatke iz okoline vozila. Kako bismo detektirali povišenje u odnosu na cestu kojom vozilo prometuje, potrebno je imati percepciju trodimenzionalnog prostora. Informaciju o dubini, koju ljudsko oko lako procesira, nije jednostavno transformirati u oblik prihvatljiv računalnoj obradi. Trodimenzionalni model okoline u analiziranim radovima generiran je na različite načine i iz različitih senzora. Jedna je opcija skeniranje optičkim uređajem LIDAR. Takav uređaj mjeri udaljenosti na osnovu analize odbijanja emitirane laserske zrake. Korištenje takvih mjernih podataka u cilju detekcije rubnika opisano je u radovima (Maye et al., 2012) i (Zhao i Yuan, 2012). Mjerni uređaji LIDAR daju kvalitetnu i preciznu procjenu stanja okoline. Cjenovno pristupačnija opcija je korištenje stereo para kamera i metoda računalnog vida. Aplikacije računalnog vida kao ulazne podatke koriste slike ili njihove sekvence. U konkretnoj primjeni slike pribavlja stereo par kamera postavljen na vozilu. Takav model prostora koristi se u analiziranim člancima (Siegemund et al., 2011), (Oniga et al., 2007) i (Turchetto i Manduchi, 2003). Postoje i radovi koji predlažu prikupljanja podataka iz više senzora radi preciznije aproksimacije scene. Pri analizi snimke često može biti korisna i informacija o poziciji vozila, dobivena primjerice iz GPS uređaja. Takve ideje šire su objašnjene u (Aufreere et al., 2003) i (McCall i Trivedi, 2006). Rad prezentira metode analize slike iz stereo para kamera.

Metode stereoskopskog računalnog vida pružaju rješenje za rekonstrukciju trodimenzionalne scene iz slika. Pronalazimo slikovne elemente jedne slike podudarne sa slikovnim elementima druge slike, odnosno stereo korespondenciju među slikama. Nastojimo kao rezultat dobiti gustu stereoskopsku rekonstrukciju sa većim brojem po-

dudarnih slikovnih elemenata. Pomoću takvih metoda iz ulaznih slika je moguće rekonstruirati strukturu oblaka točaka. Oblak točaka je niz točaka u trodimenzionalnom prostoru, gdje svaka točka ima svoju X , Y i Z prostornu koordinatu. Koordinata Y predstavlja visinsku, a Z dubinsku koordinatu u prostoru. Analiza cjelokupnog oblaka točaka za detekciju rubnika može biti vremenski zahtjevna, te stoga grupiramo točke. Model digitalne karte visina (engl. Digital Elevation Map, DEM) pojašnjen je u (Oniga et al., 2007). Točke iz oblaka grupiraju se s obzirom na neke od njihovih prostornih koordinata, smještaju se u ćelije karte visina te se za ćelije odredi prosječna visinska koordinata.

U sklopu rada implementirali smo dvije metode detekcije rubnika. Prva metoda uključuje konstrukciju visinske karte te pronalaska gradijenata na karti. Gradijenti pritom predstavljaju razlike u visini između dviju ćelija karte. Među pronađenim gradijentima, pronalazimo pravce kojima su gradijenti povezani. Detekciju pravaca vršimo Houghovom transformacijom. Navedena metoda se temelji na algoritmu predloženom u (Oniga et al., 2007). U članku (Oniga et al., 2008) predstavljena je nadogradnja na metodu u vidu uvođenja ovisnosti među okvirima snimke radi lakše detekcije rubnika u sekvenci slika. Drugo rješenje koje prezentiramo uključuje rekonstrukciju dviju površina, ceste i pločnika, te pronalazak rubnika kao granice među tim površinama. Struktura karte visina se modelira slučajnim uvjetnim poljem, pri čemu ćelijama izračunavamo vjerojatnost pripadnosti nekoj od površina na temelju njihove visinske koordinate. Takva metoda detekcije rubnika prezentirana je u (Siegemund et al., 2010) i (Maye et al., 2012). Nadogradnja na ovaj algoritam moguća je ako uvedemo vremensko filtriranje, tj. ovisnost među okvirima snimke kao što je predloženo u (Siegemund et al., 2011). Metoda rekonstrukcije rubnika modeliranjem prometnice uvjetnim slučajnim poljem predstavljena je kao robusnija i osjetljivija na problem lažnih rubnika te omogućuje rekonstrukciju rubnika kao krivulje višeg reda.

U 2. poglavlju rada prezentirani su algoritmi korišteni u programskoj implementaciji zadatka. Najprije se opisuju korištene metode rekonstrukcije scene te izgradnje karte visina. Za algoritam analize gradijenata karte visina pojašnjavaju se detekcija gradijenata Cannyjevim detektorom i Houghova transformacija. Zatim se definira stvaranje modela okoline i projekcija istog u strukturu slučajnog polja za potrebe drugog algoritma. Treće je poglavlje posvećeno programskoj implementaciji algoritama i uporabi vanjskih biblioteka za rješenje problema. Ispitni skupovi su također opisani u ovom poglavlju. Eksperimentalni rezultati zauzimaju 4. poglavlje rada. Nizom slika sa prikazanim rezultatima i mjerama kvalitete rješenja prezentira se uspješnost algoritama. Zaključak daje buduće mogućnosti rada na problemu i ostala poboljšanja.

2. Tehnike stereo vida i algoritmi detekcije rubnika

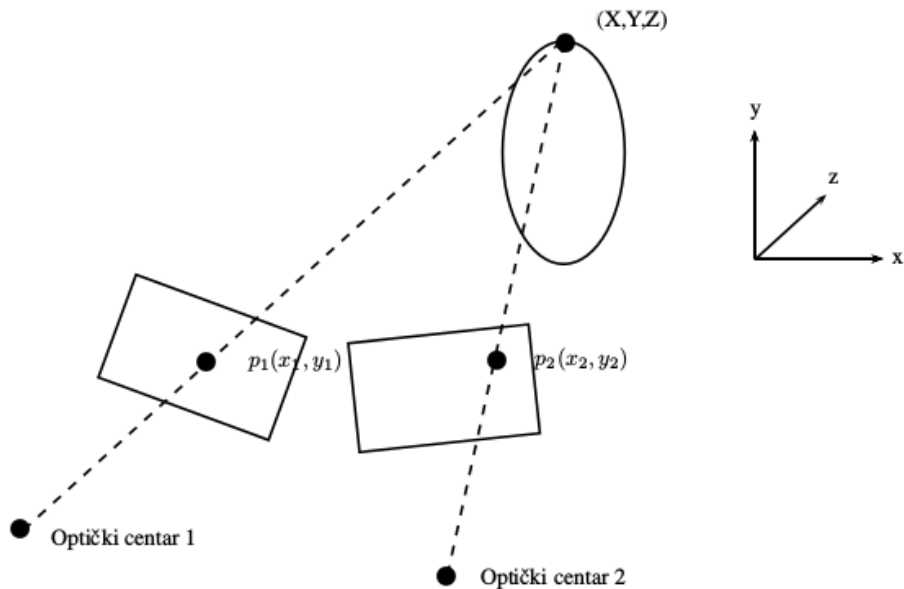
2.1. Rekonstrukcija strukture scene

Kako bi bilo moguće pribaviti podatke o trodimenzionalnom prostoru potrebno je analizirati slike prostora. Za rekonstrukciju 3D scene postoji niz metoda. Neke od tehnika se oslanjaju na sjene prisutne na slici, analizu teksture objekata na slici ili snimanja prostora uz različita osvjetljenja. U ovome se radu analizira uporaba stereoskopskog računalnog vida kao metode rekonstrukcije 3D modela prostora. Slično ljudskom vidu, za percepciju dubine u stereoskopskom vidu trebamo barem dvije slike. Za neku općenitu točku u prostoru s koordinatama (X, Y, Z) postoje projekcije na dvjema slikama, s koordinatama (x_1, y_1) i (x_2, y_2) . Primjer je dan na slici 2.1. Raznim tehnikama stvaraju se mape dubine koje predstavljaju preslikavanje iz domene koordinata ulaznih slikovnih elemenata u domenu prostora. Pouzdani algoritmi stereo vida ključni su u aplikacijama vezanim uz robotsku i autonomnu navigaciju.

Metode stereoskopskog računalnog vida uključuju proces pronalaska sličnosti i podudaranja piksela na dvjema slikama, te proces rekonstrukcije trodimenzionalnog modela iz podataka o podudarnim slikovnim elementima. Prvi algoritmi ovakve vrste fokusirali su se na podudaranje značajki (npr. rubovi, kutovi) na različitim slikama. Danas je područje stereoskopije fokusiranije na metode gustog podudaranja. Takve su metode računalno zahtjevnije, ali daju gušću mapu dubina i time bolju procjenu trodimenzionalne scene.

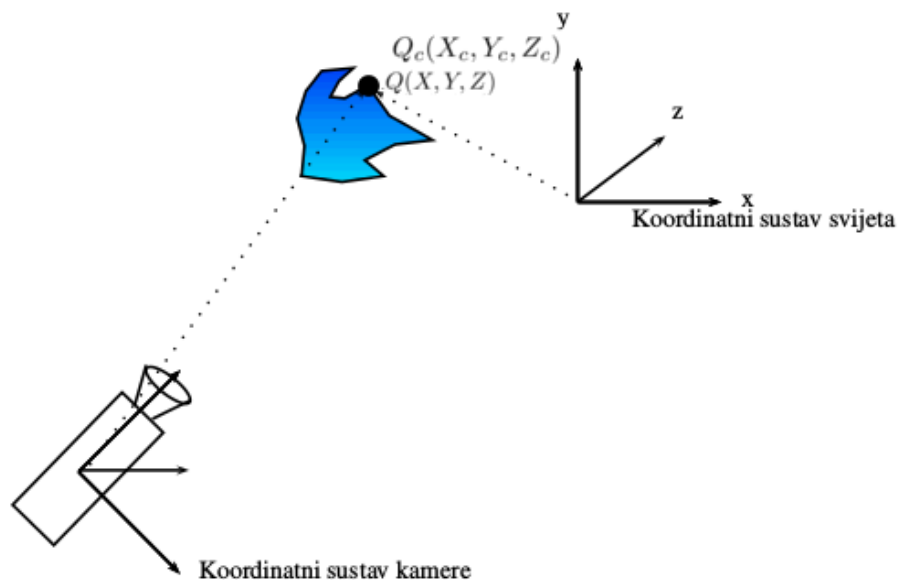
2.1.1. Stereoskopski računalni vid

Prije analize podudarnih piksela, potrebno je imati informacije o nekim parametrima kamere koja je sliku snimila. Jedan od najbitnijih parametara je udaljenost optičkih centara lijeve i desne kamere u stereo paru (engl. baseline). Svaka od kamera ima



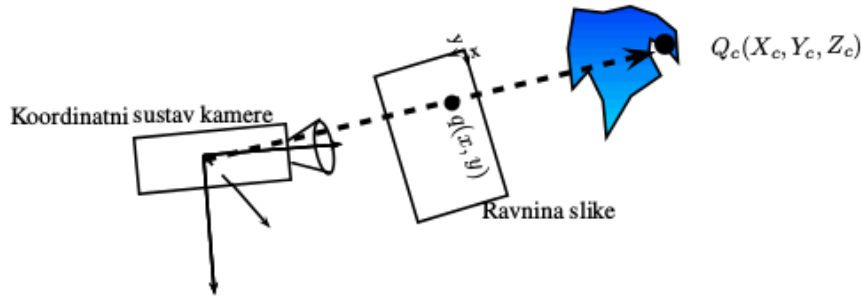
Slika 2.1: Projekcija točke u prostoru na stereo slike

svoju fokalnu duljinu. Radi jednostavnosti, transformacije se rade s kamerama jednake fokalne duljine. Također, koordinatni sustav kamere razlikuje se od koordinatnog sustava slike. Prikaz koordinatnih sustava i njihovih odnosa prikazan je na slikama 2.2 i 2.3.



Slika 2.2: Koordinatni sustavi svijeta i kamere

Koordinatni sustav prostora (svijeta) služi za prikaz točaka u prostoru. Neka točku Q u koordinatnom sustavu svijetu definiraju koordinate (X, Y, Z) . Točka P se iz koordinatnog sustava svijeta projicira u koordinatni sustav kamere. Točku u koordinatnom



Slika 2.3: Koordinatni sustavi kamere i slike

sustavu kamere označimo sa \mathbf{Q}_c sa koordinatama (X_c, Y_c, Z_c) . Transformaciju između ta dva koordinatna sustava čini rotacija te translacija između koordinatnih osi. Rotacijska matrica i translacijski vektor čine ekstrinzični skup parametara kamere. Ekstrinzični parametri definiraju odnos 2 sustava. Rotacijsku matricu označimo sa \mathbf{R} , a translacijski vektor sa \mathbf{t} . Tada vrijedi:

$$\mathbf{Q}_c = \mathbf{R}\mathbf{Q} + \mathbf{t} \quad (2.1)$$

Sljedeći je korak točku iz koordinatnog sustava kamere projicirati na sliku. Točka u slikovnom koordinatnom sustavu označena je kao q , te ima koordinate (x, y) . Odnos između koordinatnog sustava kamere i slike može se prikazati transformacijskom matricom \mathcal{M} . Matrica \mathcal{M} spada u skup intrinzičnih parametara kamere i definirana je kao:

$$\mathcal{M} = \begin{bmatrix} k * f & 0 & x_0 \\ 0 & l * f & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Tada vrijedi transformacija:

$$q' = \mathcal{M}\mathbf{Q}_c \quad (2.2)$$

$$q' = \begin{bmatrix} x' \\ y' \\ \omega' \end{bmatrix} = \begin{bmatrix} k * f & 0 & x_0 \\ 0 & l * f & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \quad (2.3)$$

Vrijednost f je fokalna duljina kamere, $\frac{1}{k}$ i $\frac{1}{l}$ su dimenzije slikovnih elemenata, a x_0 i y_0 su koordinate centra slike koje definiraju translaciju pozicije piksela. Konačno za koordinate točke p u koordinatnom sustavu slike vrijedi:

$$x = \frac{x'}{w'}; y = \frac{y'}{w'} \quad (2.4)$$

Ako koristimo stereo par kamera, neka točka \mathbf{Q} u prostoru u koordinatnom sustavu lijeve kamere je označena kao \mathbf{Q}_{cl} , a u koordinatnom sustavu desne kamere kao \mathbf{Q}_{cr} .

Za točku lijeve kamere vrijedi:

$$\mathbf{Q}_{cl} = \mathbf{R}_l \mathbf{Q} + \mathbf{t}_l \quad (2.5)$$

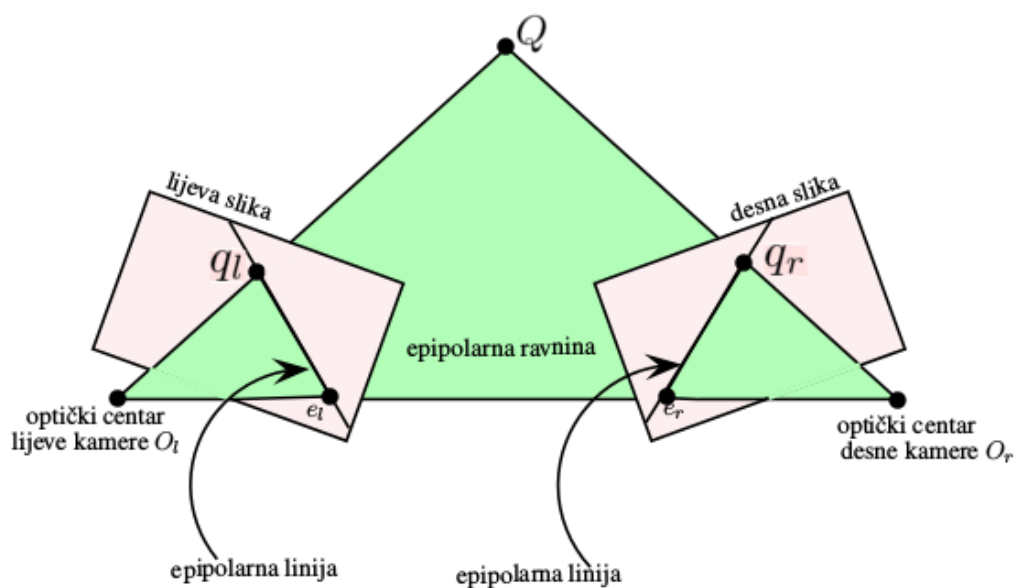
Slijedi:

$$\mathbf{Q} = \mathbf{R}_l^{-1} \mathbf{Q}_{cl} - \mathbf{R}_l^{-1} \mathbf{t}_l \quad (2.6)$$

Slično se može napisati i za transformaciju iz \mathbf{Q}_{cr} u \mathbf{Q} , pa vrijedi:

$$\mathbf{Q}_{cr} = \mathbf{R}_l \mathbf{R}_r^{-1} \mathbf{Q}_{cl} - \mathbf{R}_l \mathbf{R}_r^{-1} \mathbf{t}_l + \mathbf{t}_r = \mathbf{A} \mathbf{Q}_{cl} + \mathbf{b} \quad (2.7)$$

Nakon što smo iz parametara kamere odredili sve informacije bitne za rekonstruk-



Slika 2.4: Epipolarna ravnina

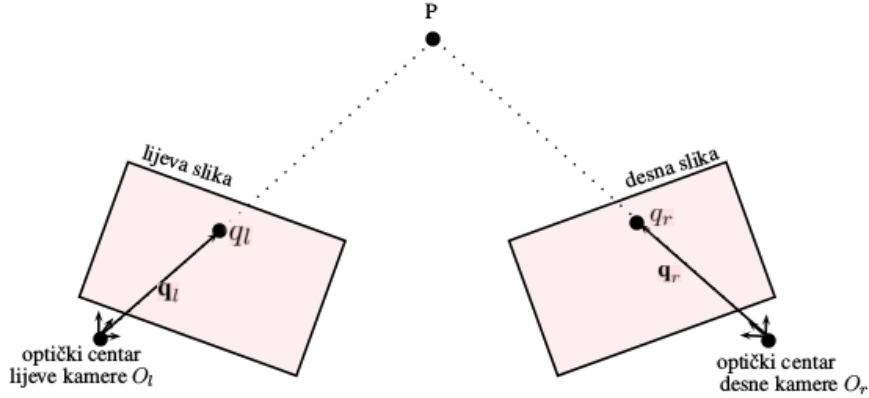
ciju scene, problem se svodi na pronalazak podudarnih piksela na dvije slike. Neka se točke \mathbf{Q}_{cl} i \mathbf{Q}_{cr} redom preslikavaju u točke q_l na slikovnoj ravnini lijeve kamere i q_r na slikovnoj ravnini desne kamere. Točkama O_l , O_r , q_l , q_r i \mathbf{Q} određena je epipolarna ravnina kao na slici 2.4. Pravac $\overrightarrow{q_r e_r}$ na slici je epipolarni pravac točke q_l . Točke e_l i e_r nazivamo epipolovima. Iz slike je očito da se točka q_l u lijevoj slici može preslikati samo u neku od točaka na epipolarnom pravcu na desnoj slici. Tim ograničenjem smanjujemo prostor na kojem tražimo podudarne piksele između dvije slike na jedan pravac. Ograničenje se naziva epipolarnim ograničenjem. Bitna implikacija epipolarnog ograničenja je da pri postupku podudaranja piksela možemo pretraživati samo jednodimenzionalni epipolarni pravac. pravac $\overrightarrow{O_l O_r}$ te pravci $\overrightarrow{O_l q_l}$ i $\overrightarrow{O_r q_r}$ leže u istoj, epipolarnoj ravnini, iz čega slijedi:

$$\overrightarrow{O_r q_r} \cdot (\overrightarrow{O_l O_r} \times \overrightarrow{O_l q_l}) = 0 \quad (2.8)$$

Uz ranije definirane \mathbf{A} i \mathbf{b} , te \mathbf{q}_l i \mathbf{q}_r kao vektore točki q_l i q_r (slika 2.5), pišemo:

$$\mathbf{q}_r \cdot (\mathbf{b} \times (\mathbf{A}\mathbf{q}_l)) = 0 \quad (2.9)$$

jer je vektor $\overrightarrow{O_l O_r}$ jednak translaciji između dva koordinatna sustava.



Slika 2.5: Vektori \mathbf{q}_r i \mathbf{q}_l

Ako sada definiramo matricu \mathbf{B} za koju vrijedi:

$$\mathbf{B}\mathbf{x} = \mathbf{b} \times \mathbf{x}, \forall \mathbf{x} \in \mathcal{R}^3 \quad (2.10)$$

možemo pisati:

$$\mathbf{q}_r^T \mathbf{E} \mathbf{q}_l = 0; \mathbf{E} = \mathbf{B}\mathbf{A} \quad (2.11)$$

Matricu \mathbf{E} nazivamo esencijalnom matricom. Neka je matrica \mathcal{M}_l projekcijska matrica između koordinatnog sustava lijeve kamere i koordinatnog sustava lijeve slike, a \mathcal{M}_r ista matrica za desnu kameru i sliku. Sada definiramo fundamentalnu matricu \mathbf{F} kao:

$$\mathbf{F} = \mathcal{M}_r^{-T} \mathbf{E} \mathcal{M}_l^{-1} \quad (2.12)$$

Pošto su \mathbf{p}_l i \mathbf{p}_r vektori točaka \mathbf{P}_{cl} i \mathbf{P}_{cr} , respektivno, vrijedi:

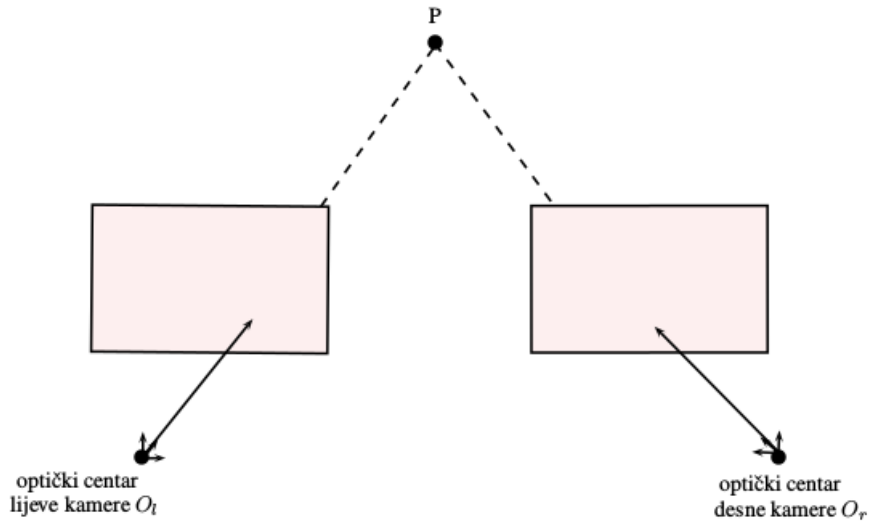
$$p_l = \mathcal{M}_l \mathbf{p}_l; p_r = \mathcal{M}_r \mathbf{p}_r \quad (2.13)$$

Iz navedenog slijedi:

$$p_r^T \mathbf{F} p_l = 0; \quad (2.14)$$

Stvaranje trodimenzionalne scene bitno je olakšano ukoliko se koriste rektificirane slike. Procesom rektifikacije slike dviju kamera projiciraju se na zajedničku slikovnu ravninu, paralelnu sa pravcem koja spaja optičke centre lijeve i desne kamere. Time se i epipolarni pravci svode na jedan redak slike, tj. paralelne su sa osi koja spaja optičke

centre kamera. Jednostavan postupak rektifikacije uključuje rotaciju kamere u poziciju u kojoj su im optičke osi okomite na spojnicu njihovih optičkih centara. Ovaj se proces izvodi nakon pronalaska fundamentalne matrice, tj. slike se podvrgavaju rotaciji. Kada bi se kamere rotirale fizički, prije pribavljanja slika, moguće je da ne bi dovoljno dobro snimile područje od interesa. Raspored slikovnih ravnina nakon rektifikacije prikazan je na slici 2.6.

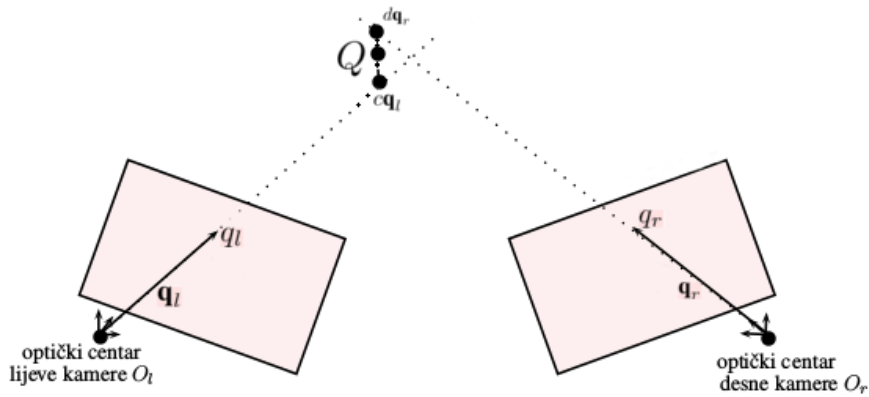


Slika 2.6: Rektificirane slikovne ravnine

Nakon što je su pronađene korespondentne točke, tražimo vrijednost dispariteta. Disparitet za neki piksel definira udaljenost horizontalnih koordinata podudarnih piksela na dvije slika. Uz definirani disparitet moguće je provesti triangulaciju. Triangulacija je postupak pronalaska trodimenzionalne koordinate točke iz pozicija točaka na stereo slikama. Ponekad se zrake sa smjerovima vektora \mathbf{q}_l i \mathbf{q}_r neće sjeći u 3D prostoru, kao na slici 2.7. Neka su c i d skalari za koje vrijedi da definiraju najbliže točke na zrakama u 3D prostoru kao $c\mathbf{q}_l$ i $d\mathbf{q}_r$. Nastojimo pronaći točku Q između navedenih točaka. Nekom od metoda minimizacije greške se pronađu optimalne vrijednosti za c i d . Primjer funkcije greške dan je u (Szeliski, 2010). Nakon toga, definiramo točku Q kao srednju točku između $a\mathbf{p}_l$ i $b\mathbf{p}_r$.

$$Q = \frac{(b\mathbf{q}_r) + \mathbf{A}(a\mathbf{q}_l) + \mathbf{b}}{2} \quad (2.15)$$

Ukoliko se zrake sijeku i obje kamere imaju jednaku fokalnu duljinu f , računamo



Slika 2.7: Proces triangulacije

točku $P(X, Y, Z)$ prema jednadžbama (2.16), gdje je d disparitet.

$$\begin{aligned} X &= \frac{x_l |\overrightarrow{O_l O_r}|}{d} \\ Y &= \frac{y_l |\overrightarrow{O_l O_r}|}{d} \\ Z &= \frac{|\overrightarrow{O_l O_r}| f}{d} \end{aligned} \quad (2.16)$$

2.1.2. Stvaranje mape dispariteta i oblaka točaka

Korištenjem ograničenja opisanih u prethodnom pododjeljku nastojimo naći podudarne piksele. Rezultat postupka podudaranja je nađena vrijednost dispariteta (razlike u horizontalnim koordinatama podudarnih piksela). Najprije treba definirati mjere sličnosti između slikovnih elemenata. Kako je navedeno u prošlom poglavlju, imamo piksel q_l na lijevoj slici I_l sa koordinatama (x_l, y_l) , te piksel q_r na desnoj slici I_r sa koordinatama u slikovnom sustavu (x_r, y_r) . Određujemo funkciju dispariteta $d(x, y)$. Ta funkcija svakom od slikovnih elemenata sa koordinatama (x, y) pridjeljuje vrijednost dispariteta. Rad (Scharstein i Szeliski, 2002) dijeli algoritam korespondencije na 4 koraka - izračun cijene podudaranja (engl. matching cost computation), agregacija cijene, računanje i optimizacija disparitetnih vrijednosti, te uglađivanje dispariteta. U prvom koraku definiramo prozor točaka W unutar kojeg računamo sličnost, kako je navedeno u (Szeliski, 2010). Najjednostavnija mjera je srednja kvadratna pogreška (engl. Mean Squared Error, MSE ili squared intensity differences, SD), definirana kao:

$$MSE = \sum_{i \in W} (I_r(x_l^i + d(x_l^i, y_l^i), y_l^i) - I_l(x_l^i, y_l^i))^2 \quad (2.17)$$

MSE se računa kao kvadratna razlika intenziteta piksela. Neke od ostalih metoda su suma apsolutnih razlika (engl. mean absolute difference, MAD ili absolute intensity

differences, AD), te normalizirana kros-korelacija (engl. normalized cross-corelation, NCC). Mjere su definirane na sljedeći način:

$$MAD = \sum_{i \in W} |I_r(x_l^i + d(x_l^i, y_l^i), y_l^i) - I_l(x_l^i, y_l^i)| \quad (2.18)$$

$$NCC = \frac{\sum_{i \in W} (I_r(x_l^i + d(x_l^i, y_l^i), y_l^i) \times I_l(x_l^i, y_l^i))}{\sum_{i \in W} (I_r(x_l^i + d(x_l^i, y_l^i), y_l^i) \times I_l(x_l^i, y_l^i))} \quad (2.19)$$

Osim metoda za dvije slike (lijevu i desnu), postoje i metode koje uspoređuju nekoliko slika za stvaranje disparitetne mape. Korak agregacije cijene može biti jednostavno zbrajanje cijena po prozoru točaka ili računanje prosječne cijene. Korak računanja i optimizacije vrijednosti dispariteta ovisi o tome koristimo li lokalnu ili globalnu metodu podudaranja. Kod lokalnih metoda naglasak je na računanju cijene podudaranja i agregaciji cijene. Definiramo li $d(x, y)$ kao disparitetnu funkciju za piksele s koordinatama (x, y) , možemo računati disparitete kao:

$$d(x, y) = \arg \min_d MSE(d(x, y), x, y) \quad (2.20)$$

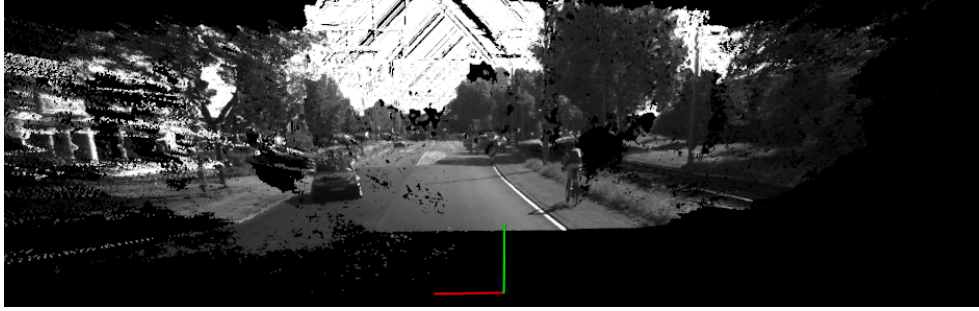
ukoliko koristimo prosječnu kvadratnu pogrešku kao funkciju cijene. Nedostatak lokalnih metoda je to što se pronađena jedinstvena podudaranja odnose samo na referentnu sliku, dok se jedan piksel druge (nereferentne) slike može preslikati u više piksela referentne slike.

S druge strane globalne metode u trećem koraku rade najznačajniji posao i često ne uključuju agregaciju cijene. Kod globalnih metoda cilj je funkcijom dispariteta minimizirati globalnu energiju koja se u (Scharstein i Szeliski, 2002) računa prema jednadžbi:

$$E(d) = E_{data} + \lambda E_{smooth} \quad (2.21)$$

Gdje je d funkcija dispariteta $d(x, y)$. Tako prvi član jednadžbe E_{data} daje informaciju o ukupnoj cijeni podudaranja ukoliko se koristi funkcija $d(x, y)$. Drugi faktor energije, E_{smooth} osigurava glatke prijelaze između dispariteta tako da zbraja razlike dispariteta između susjednih piksela. Nakon što smo definirali funkciju energije, možemo je minimizirati s obzirom na različite funkcije $d(x, y)$. Koriste se brojni algoritmi traženja minimuma (kaljenje, rezanje grafa...). Korak ugađivanja dispariteta osigurava na razini slikovnih elemenata manjih od piksela glatke prijelaze koji se na višoj razini gube zbog diskretizacije vrijednosti dispariteta.

Algoritam koji je korišten i implementiran u (Braut, 2014) je poluglobalnog podudaranja (engl. Semi-Global Block Matching), prezentiran u (Hirschmuller, 2005). Nakon stvaranja mape dispariteta, triangulacijom stvorimo točke i projiciramo ih u oblak točaka. Primjer je dan na slici 2.8.



Slika 2.8: Oblak točaka

Algoritam poluglobalnog podudaranja Implementirani algoritam je kombinacija lokalnih i globalnih metoda podudaranja. Ulazne podatke čine dvije slike (rektificirane) pribavljene kamerama sa poznatim parametrima projekcije. U prvoj verziji algoritma kao funkcija cijene podudaranja korištena je srednji uzajamni sadržaj informacije (transinformacija, engl. Mutual information, MI). Srednji uzajamni sadržaj informacije na domeni slika se definira kao razlika zbroja pojedinačnih entropija slika i zajedničke entropije:

$$MI(I_1, I_2) = H(I_1) + H(I_2) - H(I_1, I_2) \quad (2.22)$$

Ako označimo vjerojatnosnu distribuciju intenziteta slike I sa P_I , entropiju možemo definirati kao:

$$H_I = - \sum_{i=0}^{255} P_I(i) \log P_I(i), \quad (2.23)$$

gdje su i moguće diskretne vrijednosti intenziteta. Zajedničku entropiju slika I_1 i I_2 formalno definiramo kao:

$$H_{I_1, I_2} = - \sum_{i_1=0}^{255} \sum_{i_2=0}^{255} P_{I_1, I_2}(i_1, i_2) \log P_{I_1, I_2}(i_1, i_2) \quad (2.24)$$

Kako definicija zajedničke entropije zahtijeva unaprijed određenu funkciju dispariteta, entropija se računa na domeni piksela. Računanje zajedničke entropije uključuje konvoluciju dvodimenzionalnog Gaussova filtra sa vjerojatnosnom distribucijom intenziteta piksela. Jednadžbe zajedničke entropije na domeni piksela p sada imaju sljedeći oblik:

$$H_{I_1, I_2} = \sum_p h_{I_1, I_2}(I_{1p}, I_{2p}) \quad (2.25)$$

$$h_{I_1, I_2} = - \frac{1}{n} \log(P_{I_1, I_2}(i, k) \otimes \text{gauss}(i, k) \otimes \text{gauss}(i, k)),$$

pri čemu je n broj piksela u prozoru koji uspoređujemo. Na sličan način možemo računati i pojedinačnu entropiju piksela. Definiramo ju kao:

$$h_I(i) = -\frac{1}{n} \log(P_I(i) \otimes \text{gauss}(i) \otimes \text{gauss}(i)) \quad (2.26)$$

Konačno, definiramo srednji uzajamni sadržaj informacije dvaju piksela kao:

$$mi_{I_1, I_2}(i, k) = h_{I_1}(i_1) + h_{I_2}(i_2) - h_{I_1, I_2}(i_1, i_2) \quad (2.27)$$

Funkciju cijene podudaranja definiramo za neki piksel $I_l(x_l, y_l)$ i disparitetnu funkciju d kao:

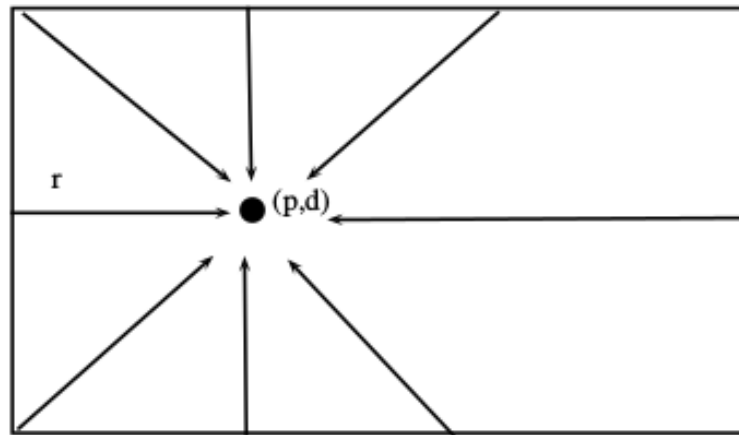
$$C(I_l(x_l, y_l), d) = -mi_{I_l, I_r}(I_l(x_l, y_l), I_r(x_l + d, y_l)) \quad (2.28)$$

U algoritmu poluglobalnog podudaranja, energija piksela se računa kao prosječna vrijednost energija piksela u bloku. Ukoliko funkciju cijene podudaranja (za piksel p sa susjedima iz skupa N_4 uz funkciju dispariteta d) označimo sa $C(p, d)$, energija disparitetne funkcije d računa se kao:

$$E(d) = \sum_p \left(C(p, d) + \sum_{q \in N_4} P_1 T[|d(p) - d(q)| = 1] + \sum_{q \in N_4} P_2 T[|d(p) - d(q)| > 1] \right) \quad (2.29)$$

Prvi član se odnosi na E_{data} definiran jednadžbom (2.21) i daje ukupnu cijenu podudaranja, dok drugi i treći član osiguravaju glatke prijelaze između dispariteta susjednih slikovnih elemenata. P_1 je konstantan element penalizacije za disparitete susjednih elemenata koji se razlikuju za vrijednost 1, dok se P_2 mijenja ovisno o razlici u intenzitetu susjednih piksela. Pri tome mora vrijediti $P_2 \geq P_1$. Funkcija T vraća 1 ukoliko je jednadžba ili nejednadžba unutar zagrada istinita, a 0 inače. Minimizacija funkcije energije zahtijeva minimizaciju funkcije dispariteta na cijeloj slici, što može zahtijevati veće računalne resurse. Stoga se proces minimizacije energije radi duž jednog reda slike, tj. na jednoj dimenziji slike. Agregacija cijena radi se u nekoliko smjerova slike. Cijena u smjeru r za piksel p računa se kao zbroj cijene podudaranja u pikselu p i minimalne vrijednosti između cijena susjednih piksela u smjeru r . Ukoliko definiramo osam smjerova kao na slici 2.9, za svaki smjer možemo računati minimalnu cijenu za put r , L_r . Definicija puta na smjeru r sa minimalnom cijenom je kako slijedi:

$$L_r(p, d) = C(p, d) + \min \left[\begin{array}{l} L_r(p - r, d) \\ L_r(p - r, d - 1) + P_1 \\ L_r(p - r, d + 1) + P_1 \\ \min_i L_r(p - r, i) + P_2 \end{array} \right] - \min_k L_r(p - r, k) \quad (2.30)$$



Slika 2.9: Smjerovi agregacije cijene kod poluglobalnog podudaranja

Konačno, agregirana cijena računa se kao zbroj cijena po smjerovima:

$$S(p, d) = \sum_r L_r(p, d) \quad (2.31)$$

Treći korak je izračun disparitetne funkcije. Funkcija $d(x, y)$ mora minimizirati cijenu $S(p, d)$, dakle za svaki piksel p disparitet d je određen kao $\min_d S(p, d)$. Za svaki piksel na lijevoj slici određujemo koji od piksela na pripadnom epipolarnom pravcu minimizira cijenu $S(p, d)$. Četvrti korak, ugađivanje dispariteta se vrši eliminacijom ekstremnih i vanpopulacijskih vrijednosti ili usporedbom s promjenama intenziteta. Rezultat rada algoritma na sceni prometnice prikazan je na slici 2.10.



Slika 2.10: Mapa dispariteta izgrađena algoritmom poluglobalnog podudaranja

2.2. Predstavljanje scene kartom visina

Ideja grupiranja točaka koje se nalaze unutar oblaka točaka za potrebe modela 3D scene prezentirana je u (Elfes, 1989). Pojam rešetka zauzetosti (engl. occupancy grid) analiziran je za problem modeliranja prometnice u članku (Badino et al., 2007). Takav

oblik reprezentacije oblaka točaka ima mnoštvo primjena u prostornom modeliranju okoline, specijalno u navigaciji robota.

Jedan oblik grupiranja točaka u ćelije rešetke je digitalna karta visina (DEM). Nakon izgradnje guste 3D stereo slike, vršimo projekciju u kartu visina, tako da se nekoliko točaka iz oblaka točaka projicira u jednu ćeliju karte. Svako grupiranje omogućava brži rad na detekciji objekata i značajki na slici. Različite diskretizacije rešetke, daju različite rezolucije na većoj dubini (udaljenosti od kamere u z-smjeru), što ima za posljednicu različitu točnost reprezentacije za udaljene objekte. Rad (Badino et al., 2007) navodi 3 načina predstavljanja oblaka točaka rešetkom:

- diskretizacija po kartezijevim koordinatama
- diskretizacija po stupcu slike i disparitetu
- diskretizacija po polarnim koordinatama

U (Badino et al., 2007) su također navedeni zahtjevi za ovakvu tehniku rada sa 3D podacima:

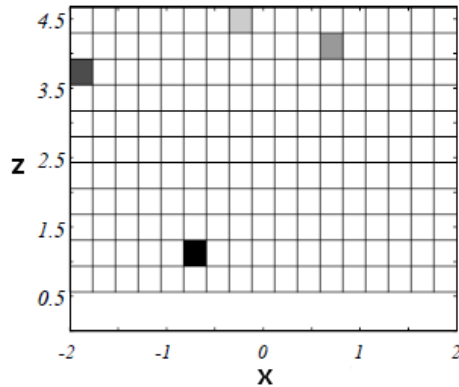
- reprezentacija mora biti kompaktna, smanjuje volumen podataka
- kompletnost - informacija ostaje očuvana
- stabilnost - male razlike u ulaznim podacima ne utječu jače na reprezentaciju podataka
- robusnost - vanpopulacijske vrijednosti ne smiju veći utjecaj na reprezentaciju podataka

2.2.1. Reprezentacije oblaka točaka

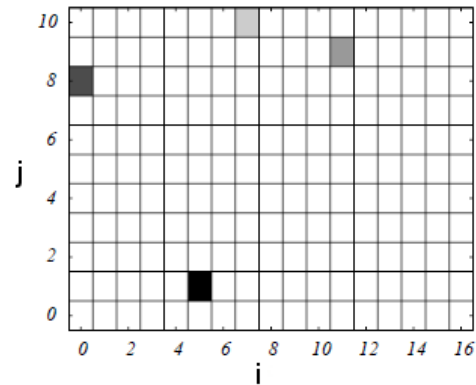
Karta visina je dvodimenzionalno polje, rešetka koja sprema vrijednosti visine za svaku ćeliju. Trodimenzionalni model prostora projicira se u rešetku sa ćelijama sa koordinatama (i, j) . Presjek ćelija ne postoji, svaka od točaka trodimenzionalnog oblaka se projicira u samo jednu ćeliju.

Najjednostavniji oblik reprezentacije oblaka točaka rešetkom je karta visina čije su ćelije grupirane po kartezijevim koordinatama. Za svaku točku u oblaku odredi se projekcija u ćeliju na osnovu x i z koordinate točke. Svako ćeliji dodijeli se jednaka širina i dubina (Δx i Δz). Ako uzmemo u obzir da se na većim udaljenostima od optičkog centra kamere sve manje piksela preslikava u neku ćeliju, slijedi da je teško odrediti visinu ćelije. Manji broj točaka slijedi iz manjeg raspona dispariteta koji se preslikavaju u danu ćeliju. Rješenja se mogu naći u promjeni dimenzija ćelija na većim udaljenostima, no to veže i veće zahtjeve za računalnim resursima. Primjer karte visina

s diskretizacijom po xz ravnini dan je na slici 2.11. Lijeva slika prikazuje područje u prostoru koje diskretiziramo (xz ravninu), dok desna slika prikazuje pripadnu kartu visina.



(a) xz ravnina prostora



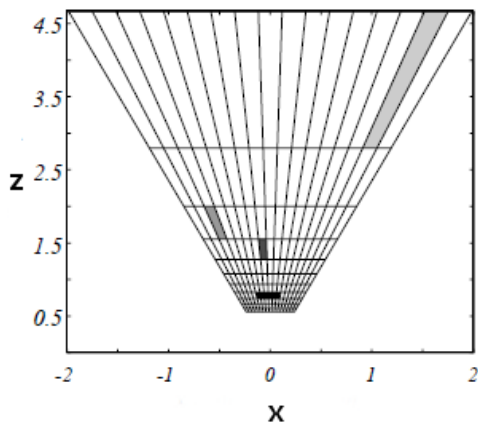
(b) Karta visina s ćelijama s koordinatama i (raspon x koordinate) i j (raspon z koordinate)

Slika 2.11: Karta visina sa ćelijama grupiranim po koordinatnim osima

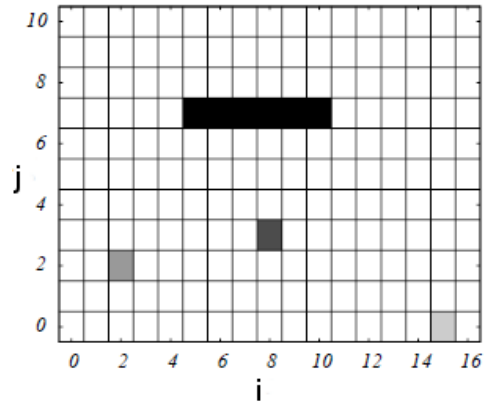
Sljedeći oblik karte visina nastaje diskretizacijom po stupcu slike te disparitetu. Svaka ćelija zauzima jedinstveni raspon stupaca i i dispariteta. Poboljšanje u odnosu na podjelu po koordinatnim osima je činjenica da ćelije udaljene od optičkog centra kamere u smjeru z -osi sada u koordinatnom sustavu slike zauzimaju više slikovnih elemenata i moguće im je preciznije odrediti prosječnu visinu. Primjer xz ravnine i karte visina izgrađene diskretizacijom po stupcima i disparitetu vidljivi su na slici 2.12. Lijeva slika prikazuje xz ravninu prostora, a desna pripadnu kartu visina. Jedno polje ravnine na lijevoj slici preslikava se u jednu ćeliju karte visina. Kako je dubinska koordinata obrnuto proporcionalna disparitetu (definirano jednačinom (2.16)), dijelovi ravnine sa većom dubinskom koordinatom prikazani su u ćelijama sa malim vrijednostima koordinate j (manje vrijednosti dispariteta).

Konačno, kartu visina je moguće izgraditi na osnovu polarnih koordinata. Ćelije zauzimaju jedinstveni raspon stupaca slike i i z koordinata koordinatnog sustava. Kao što je već navedeno u prošlom ulomku, dubina varira obrnuto proporcionalno disparitetu. Rezolucija dispariteta se smanjuje sa većom dubinom piksela na slici. Stoga polarne koordinate pružaju veću rezoluciju elemenata u odnosu na prethodno opisane reprezentacije oblaka točaka. Primjer je dan na slici 2.13. Slika 2.13a prikazuje xz ravninu prostora podijeljenu u polja po stupcima slike i z koordinati, a slika 2.13b pripadnu kartu visina).

U okviru rada testirane su podjela s obzirom na koordinatne osi i podjela na stupce

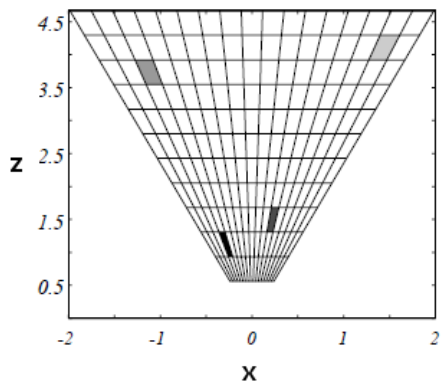


(a) xz ravnina prostora

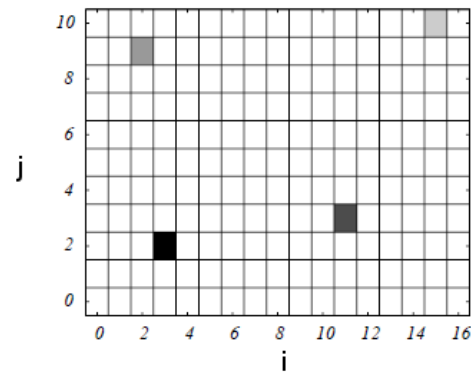


(b) Karta visina sa ćelijama s koordinatama i (raspon stupaca slike) i j (raspon vrijednosti dispariteta)

Slika 2.12: Karta visina sa ćelijama grupiranim po stupcima i disparitetu



(a) xz ravnina prostora



(b) Karta visina sa ćelijama s koordinatama i (raspon stupaca slike) i j (raspon z koordinate)

Slika 2.13: Karta visina sa ćelijama točkaka grupiranih po stupcima i dubinskoj koordinati

i disparitete. Za svaku ćeliju karte visina izračunava se visina. Kako se traži rubnik uz cestu, promatraju se samo ćelije čija prosječna visina nema veća odstupanja od visine ceste. Od visina svih točkaka iz oblaka koji se projiciraju u određenu ćeliju računa se histogram visina. Histogram bilježi sve visine u određenom rasponu. Vršni se izjednačavanje histograma, čime se postiže smanjenje osjetljivosti histograma na vanpopulacijske vrijednosti (ignoriraju se točke čija visina jače odstupaju od prosječne vrijednosti). Na izjednačenom histogramu pronađe se medijan, te se postavi kao visina ćelije. Karta visina sa diskretizacijom po stupcima i disparitetu, izgrađena na slici iz

KITTI skupa dana je na slici 2.14.



Slika 2.14: Karta visina sa slike iz KITTI ispitnog skupa

2.3. Rekonstrukcija rubnika analizom gradijenata karte visina

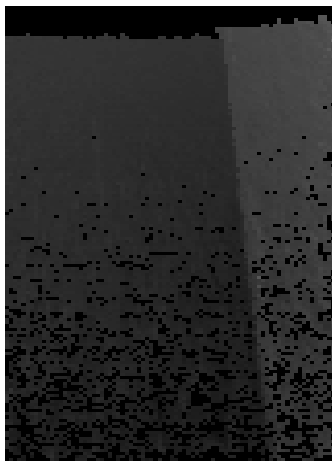
Ova metoda rekonstrukcije rubnika zahtijeva definiranje ispravnih ćelija na karti visina. Područje ispravnih ćelija uključuje ćelije u koje se preslikava barem jedna točka iz oblaka točaka. Pri analizi karte visina, nastoji se isključiti neispravne ćelije. Karta visina tretira se kao matrica slična slici. Detekcija gradijenata je bitan korak algoritma i zahtijeva da na karti visina nema većeg broja praznih (neispravnih) ćelija. Stoga se karta podvrgava morfološkoj operaciji dilatacije. Operacijom se vrši konvolucija ćelija karte sa strukturnim elementom koji osigurava proširenje oblika na karti visina. Ukoliko neispravna ćelija ima u susjedstvu više ispravnih ćelija, ona isto postaje ispravna i dobiva vrijednost visinske koordinate na osnovu vrijednosti visinskih koordinata susjednih ćelija. Rezultat je širenje granica regija ispravnih ćelija. Na tako obrađenoj matrici vrši se detekcija gradijenta Cannyjevim detektorom. Rezultat je binarna karta na kojoj su označene velike magnitude gradijenata. Među tim gradijentima Houghovom transformacijom izdvajaju se povezani nizovi gradijenata koji predstavljaju pravce, što kao rezultat može dati rubnike estimirane pravcem. Koraci algoritma vidljivi su na slici 2.15.



(a) Canny detektor rubova



(b) Pronađeni rubovi



(c) Karta visina



(d) Pravci pronadjeni među rubovima

Slika 2.15: Koraci u detekciji rubova

2.3.1. Cannyjev detektor rubova

Detekcija rubova je alat u procesiranju slike koja za rezultat ima pronalazak diskontinuiteta u intenzitetu svjetlosti slike. Intenziteti u konkretnom primjeru su projicirane visine ćelija, dok je obrađena karta visina ulazna slika. Jedan od najrobustnijih detektora je Cannyjev operator. Prvi je korak algoritma Cannyjeva detektora zamućivanje slike. Zamućivanje se vrši konvolucijom slike sa Gaussovom filtrom i time dobijemo redukciju šuma na slici. Rub je definiran kao diskontinuitet, konkretno ekstremne vrijednosti prve derivacije u točkama slike. Za pronalazak tih derivacija, tj. gradijenta intenziteta koristi se neki od filtara kao što su Sobel ili Roberts operator. Sobel operator primjerice, definiran je za traženje gradijenta u smjeru x i y osi koordinatnog

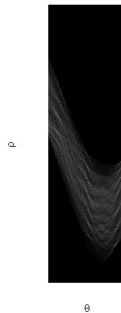
sustava kao:

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad Sobel_y = \begin{bmatrix} -1 & -2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Sljedeći je korak algoritma stanjivanje rubova (engl. non-maximum suppression). Tim se korakom određuje magnituda gradijenta u točki te se izdvajaju magnitude s najvećim vrijednostima. Konačno, uporabom segmentacije dvostrukom pragom nalaze se samo rubovi koji prelaze viši prag ili rubovi koji prelaze niži prag i susjedni su rubovima sa višim pragom. Rubovi kao značajke koriste se u brojnim algoritmima računalnog vida.

2.3.2. Houghova transformacija

Houghova transformacija također je bitna metoda ekstrakcije značajki u računalnom vidu. Metoda se temelji na dvodimenzionalnom polju akumulatora. Akumulacijsko polje u parametarskom prostoru, uz intenzitet piksela kao vrijednost ćelije akumulacijskog polja prikazano je na slici 2.16.



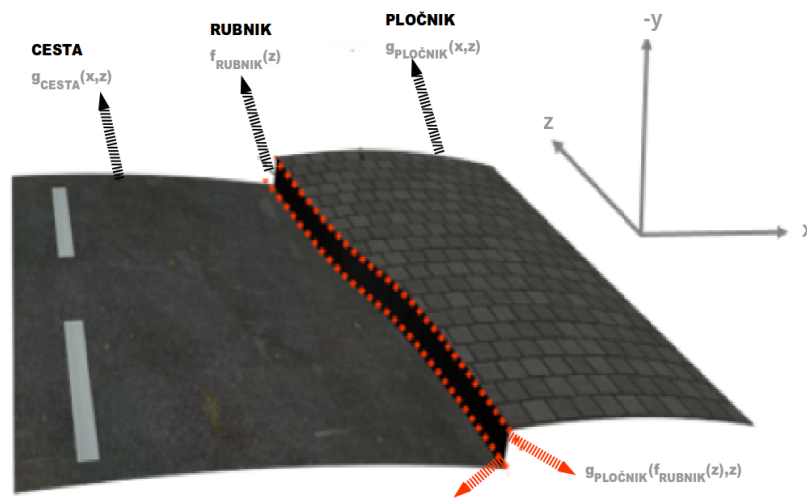
Slika 2.16: Akumulacijsko polje Houghove transformacije

Akumulator prati detekcije pravaca definiranih jednačbom $r = x \cos \theta + y \sin \theta$. Za svaki slikovni element na poziciji (x, y) , te njegove susjedne elemente, Houghova transformacija traži crte koje prolaze kroz piksel. Ukoliko pravac definiran parametrima r i θ prolazi kroz analizirani piksel, ćelija akumulacijskog polja na toj poziciji uvećava svoju vrijednost. Kako bismo pronašli pravce među svim ćelijama u parametarskom prostoru, potrebno je izdvojiti maksimume akumulacijskog polja. Maksimumi se traže jednostavnim određivanjem praga ili drugim statističkim metodama. Konačno, kao rezultat imamo pravce među ranije pronađenim rubovima, te ih možemo projicirati natrag u koordinatni sustav slike.

2.4. Rekonstrukcija rubnika korištenjem uvjetnog slučajnog polja

2.4.1. Estimiranje parametara modela prometnice

Druga metoda pronalaska uključuje stvaranje modela scene. Modeliramo cestu, pločnik i krivulju koja ih razdvaja, tj. krivulju rubnika. Model je temeljen na onome definiranom u (Siegemund et al., 2010) i prikazan je na slici 2.17. Kako je vidljivo



Slika 2.17: Model prometne scene

iz slike 2.17 potrebno je definirati funkcije $g_{pločnik}$, g_{cesta} i f_{rubnik} . Funkcije $g_{pločnik}$ i g_{cesta} preslikavaju x i z koordinatu u y (visinsku) koordinatu. Te funkcije modeliraju površine pločnika i ceste. Funkcija f_{rubnik} preslikava z koordinatu u x koordinatu i modelira krivulju rubnika.

Funkcije $g_{pločnik}$ i g_{cesta} definiraju površine drugog reda. Ako označimo površinu ceste sa \mathcal{C} , možemo ju definirati kao:

$$\mathcal{C} = \left\{ [x, y, z]^T \mid y = g_{cesta}(x, z) \right\} \quad (2.32)$$

Slično radimo i za površinu pločnika koju označavamo sa \mathcal{P} :

$$\mathcal{P} = \left\{ [x, y, z]^T \mid y = g_{pločnik}(x, z) \right\} \quad (2.33)$$

Kako $g_{plocnik}$ i g_{cesta} definiraju površine drugog reda, moguće ih je odrediti sa 4 parametra. Definiramo pomoćni vektor \mathbf{q} kao:

$$\mathbf{q} = \begin{bmatrix} x \cdot z \\ x \\ z \\ 1 \end{bmatrix} \quad (2.34)$$

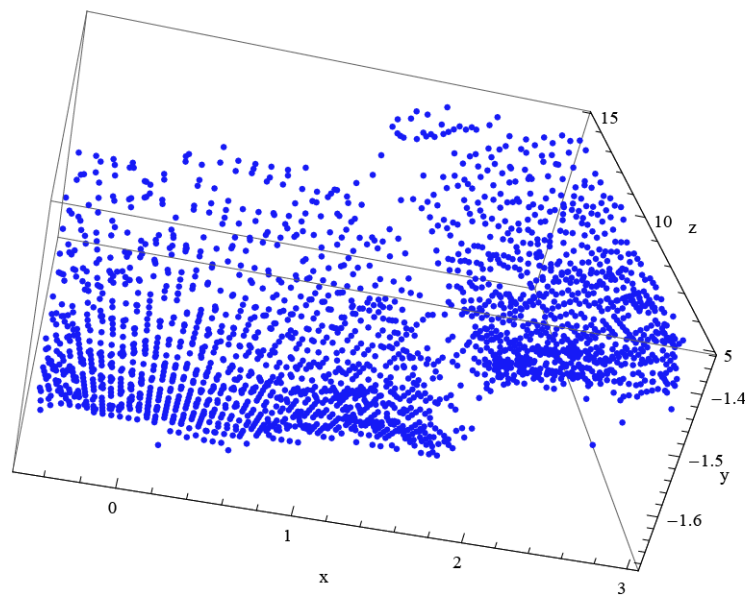
Sada je moguće definirati funkcije kao:

$$g_{cesta}(x, z) = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix} \cdot \mathbf{q} = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix} \cdot \begin{bmatrix} x \cdot z \\ x \\ z \\ 1 \end{bmatrix} \quad (2.35)$$

Sličnu definiciju dajemo i za $g_{plocnik}$:

$$g_{plocnik}(x, z) = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \end{bmatrix} \cdot \mathbf{q} = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \end{bmatrix} \cdot \begin{bmatrix} x \cdot z \\ x \\ z \\ 1 \end{bmatrix} \quad (2.36)$$

Na slici 2.18 možemo vidjeti oblak točaka s modeliranim površinama pločnika i ceste, \mathcal{P} i \mathcal{C} . Razgraničenje između površine \mathcal{C} i \mathcal{P} predstavljamo funkcijom f_{rubnik} , defini-

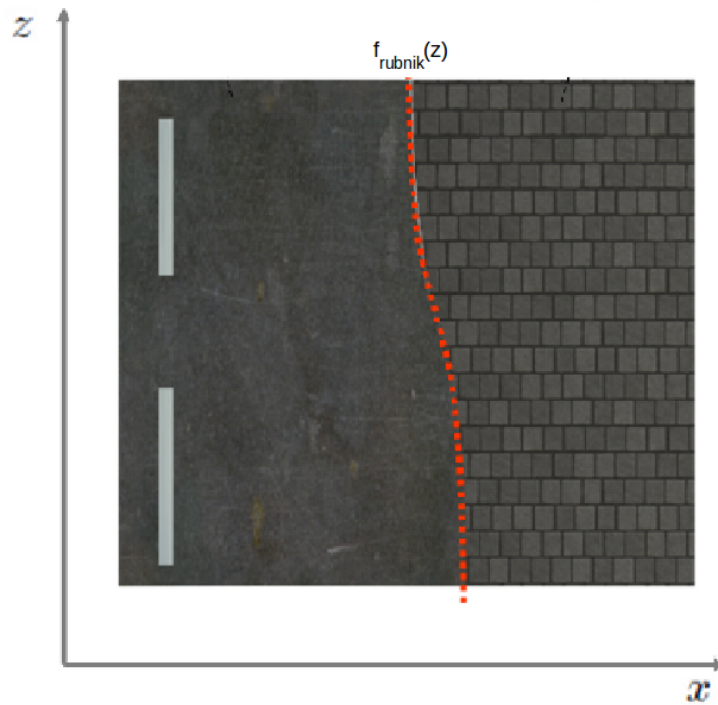


Slika 2.18: Oblak točaka u modelu prometne scene

ranom kao polinom trećeg stupnja u xz ravnini s obzirom na z koordinatu. Formula (2.37) definira funkciju.

$$f_{rubnik}(z) = z^3 r_0 + z^2 r_1 + z r_2 + r_3 \quad (2.37)$$

Funkciju, dakle, možemo definirati sa četiri parametra. Graf funkcije na modelu iz ptičje perspektive možemo vidjeti na slici 2.19. Sada možemo definirati donju i gornju



Slika 2.19: Prikaz funkcije f_{rubnik}

krivulju rubnika kao:

$$D(z) = g_{cesta}(f_{rubnik}(z), z) \quad (2.38)$$

$$G(z) = g_{plocnik}(f_{rubnik}(z), z) \quad (2.39)$$

Zatim definiramo i površinu rubnika s oznakom \mathcal{R} :

$$\mathcal{R} = \left\{ [x, y, z] \mid x = f_{rubnik}(z), D(z) < y < G(z) \right\} \quad (2.40)$$

Estimacija površina ostvaruje se optimizacijom parametara površine. Želimo pronaći parametre koji što bolje procjenjuju sve točke iz karte visine koje su označene oznakom pripadne površine. Čelije mogu imati oznake ceste ili pločnika u prvome koraku optimizacije parametara. Potrebno je pridijeliti inicijalne oznake ćelijama karte

visina. Kako inicijalno nemamo informaciju o tome koje ćelije pripadaju kojoj površini, vršimo pretraživanje po xz ravnini kako bismo pronašli ćelije za najvećim visinskim razlikama u odnosu na susjedne ćelije. Iz pronađenih ćelija sa najvećim visinskim razlikama estimiramo pravac. Pretražimo li xz ravninu, dobit ćemo niz ćelija sa svojim x i z koordinatama u prostoru. Nastojimo estimirati pravac $x = a + bz$, gdje su a i b skalari. Ukoliko imamo n ćelija, koeficijente estimiramo prema sljedećim jednadžbama:

$$b = \frac{\sum_{i=1}^n z_i x_i - \frac{1}{n} \sum_{i=1}^n z_i \sum_{j=1}^n x_j}{\sum_{i=1}^n z_i^2 - \frac{1}{n} (\sum_{i=1}^n z_i)^2} \quad (2.41)$$

$$a = \frac{1}{n} \sum_{i=1}^n x_i - b \frac{1}{n} \sum_{i=1}^n z_i$$

Nakon što estimiramo koeficijente pravca, sve ćelije i karte visina za koje vrijedi $x_i < a + bz_i$ označimo kao ćelije ceste, a sve ćelije i za koje vrijedi $x_i > a + bz_i$ označimo kao ćelije pločnika. U kasnijim koracima koriste se oznake ćelija pribavljene sljedećim korakom algoritma (algoritam propagacije vjerojatnosti).

Za estimaciju površine pločnika \mathcal{P} nastojimo estimirati vektor koeficijenata $p = [p_0, p_1, p_2, p_3]$. Sa DEM_p označene su sve ćelije karte visina koje imaju oznaku pločnika. Visina h_i neka predstavlja visinu ćelije i , a σ_i^2 varijancu visine vezanu uz ćeliju i . Vektor q ponovno je definiran kao u (2.34). Funkciju pogreške koju koristimo pri optimizaciju parametara definiramo kao ponderiranu pogrešku najmanjih kvadrata, kao što je navedeno u jednadžbi (2.42)

$$p = \arg \min_p \left(\sum_{i \in DEM_p} \left(\frac{(h_i - p * q)^2}{\sigma_i^2} \right) \right) \quad (2.42)$$

Na isti način vektor parametara $c = [c_0, c_1, c_2, c_3]$ koji definira površinu ceste optimiramo istom funkcijom pogreške. Ćelije sa oznakom ceste označavamo kao DEM_c .

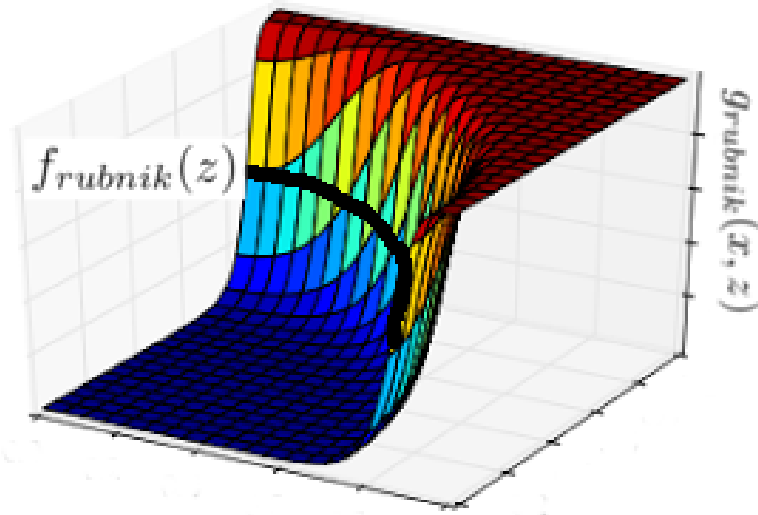
$$c = \arg \min_c \left(\sum_{i \in DEM_c} \left(\frac{(h_i - c * q)^2}{\sigma_i^2} \right) \right) \quad (2.43)$$

Kako bismo procijenili funkciju krivulje rubnika, tj. funkciju $f_{rubnik}(z)$, uvest ćemo novu funkciju, $g_{rubnik}(x, z)$. Definiramo funkciju u jednadžbi (2.44)

$$g_{rubnik}(x, z) = \frac{2}{1 + \exp(b(f_{rubnik}(z) - x))} - 1 \quad (2.44)$$

Ova funkcija definira površinu drugog reda. Površina uključuje ćelije s oznakom pločnik i ćelije s oznakom ceste. Zahvaljujući ovoj funkciji možemo izvršiti razdvajanje

ćelija u dvije skupine za koje vrijedi ili $g_{rubnik}(x, z) > 0$ ili $g_{rubnik}(x, z) < 0$, pri čemu je granica na krivulji $f_{rubnik}(z)$. Koeficijent b pri tome određuje koliko će oštra biti granica između dvije skupine ćelija. Primjer površine modelirane funkcijom $g_{rubnik}(x, z)$ dan je na slici 2.20. Kako bismo optimizirali koeficijente funkcije f_{rubnik} , definiramo



Slika 2.20: Prikaz funkcije g_{rubnik}

funkciju greške kao u jednadžbi (2.45). Optimiziramo vektor $\mathbf{r} = [r_0, r_1, r_2, r_3]$

$$r = \arg \min_r \left(\sum_{i \in DEM_c} \left(-1 - g_{rubnik}(x_i, z_i) \right)^2 + \sum_{i \in DEM_p} \left(1 - g_{rubnik}(x_i, z_i) \right)^2 \right) \quad (2.45)$$

U samoj implementaciji ograničavamo kubični parametar r_0 krivulje rubnika na male vrijednosti oko nule, kao i parametre c_0 i p_0 zbog velikog prostora pretraživanja kod optimizacije.

2.4.2. Uvjetno slučajno polje kao grafički model

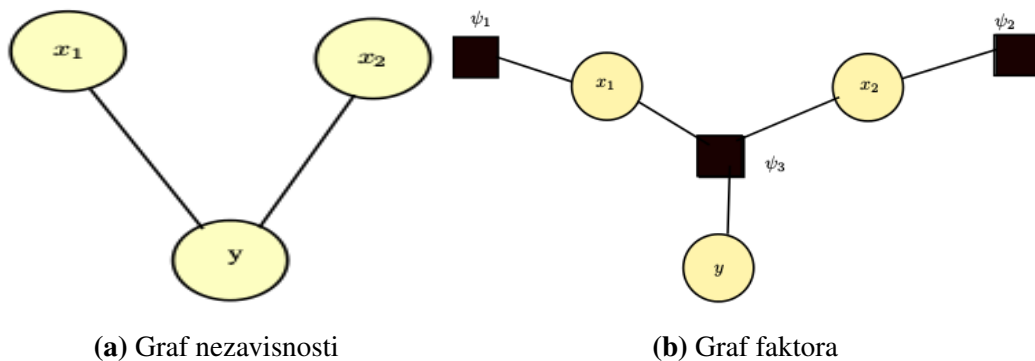
Vjerojatnosni modeli imaju velik značaj u računalnom vidu, posebno zbog činjenice da imaju grafičku reprezentaciju. Grafičkim vjerojatnosnim modelom dijagramski prikazujemo vjerojatnosnu distribuciju. Pri modeliranju grafom, svaki vrh grafa predstavlja slučajnu varijablu. Ukoliko dva vrha nisu povezana bridom, to znači da su uvjetno nezavisna. Uvjetnu nezavisnost formalno definiramo na sljedeći način - dvije slučajne varijable x i y su uvjetno nezavisne uz danu treću varijablu z ukoliko vrijedi:

$$p(x, y|z) = p(x|y)p(y|z) \quad (2.46)$$

Definirajmo graf nezavisnosti $G = (V, E)$. Vrijedi $V = X \cup Y$ gdje su X i Y skupovi slučajnih varijabli. Modelirajmo graf na primjeru distribucije $p(x_1, x_2, y)$. Neka su varijable x_1 i x_2 uvjetno neovisne uz dani y . Graf nezavisnosti prikazan je na slici 2.21a Izgradnjom grafa nezavisnosti, moguće je odrediti uvjetnu vjerojatnosne parametre distribucije modela. Bitna implikacija korištena uvjetne neovisnosti je u tome što se vjerojatnosna distribucija može rastaviti na produkt faktora. Pri tome svaki faktor uključuje podskup svih slučajnih varijabli. Općenito, ako je \mathcal{S} skup svih slučajnih varijabli, a $p(\mathbf{v})$ vjerojatnosna distribucija modela, možemo definirati faktorizaciju kao:

$$p(\mathbf{v}) = \prod_{s \in \mathcal{S}} \psi_s(\mathbf{v}_s) \quad (2.47)$$

Definirajmo primjer faktorizacije kao $p(x_1, x_2, y) = p(x_1)p(x_2)p(y|x_1, x_2)$. Pri tome označavamo faktore kao $\psi_1 = p(x_1)$, $\psi_2 = p(x_2)$ i $\psi_3 = p(y|x_1, x_2)$. Graf faktora crtamo kao na slici 2.21b Grafičke modele dijelimo na usmjerene i neusmjerene.



Slika 2.21: Grafički model

Usmjereni grafički modeli predstavljaju Bayesovu mrežu. Promotrimo primjer naivnog Bayesova klasifikatora. Za uvjetnu vjerojatnosti u navedenom klasifikatoru vrijedi

Bayesovo pravilo:

$$p(y|\mathbf{x}) = \frac{p(y, \mathbf{x})}{p(\mathbf{x})} \quad (2.48)$$

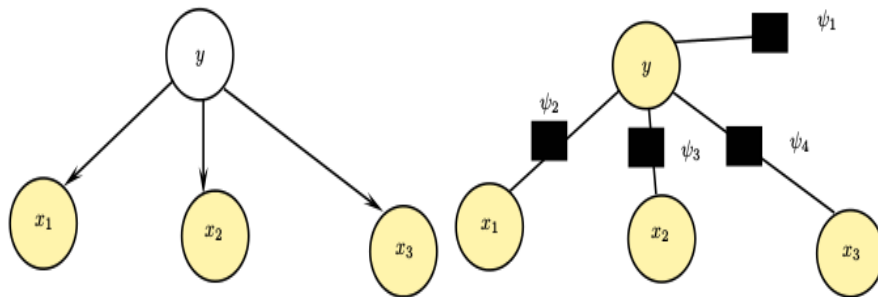
Pri tome vrijedi $\mathbf{x} = (x_1, x_2, \dots, x_n)$, dok je y klasa ulaznog vektora \mathbf{x} . Zajednička vjerojatnost $p(y, \mathbf{x})$ modelirana je kao $p(y) \prod_{i=2}^n p(x_i|x_{i-1}, \dots, x_1, y)$. Naivni Bayesov klasifikator pretpostavlja uvjetnu neovisnost varijabli x_i . Stoga vrijedi:

$$p(y|\mathbf{x}) \propto p(y) \prod_{i=1}^n p(x_i|y) \quad (2.49)$$

Ako su varijable opservacije označene sa x_i , a y predstavlja varijablu klase, možemo faktorizirati naivni Bayes klasifikator za tri promatrane varijable kao $p(y, x_1, x_2, x_3) = p(y)p(x_1|y)p(x_2|y)p(x_3|y)$. U usmjerenim grafičkim modelima definiramo vrhove "roditelje" v_i^{roditelj} o kojima ovise neki od vrhova v_i . Formalno pišemo distribuciju vjerojatnosti kao:

$$p(\mathbf{v}) = \prod_{i=1}^I p(v_i|v_i^{\text{roditelj}}) \quad (2.50)$$

Pri crtanju usmjerenog grafa, strelice se povlače od vrha roditelja. Primjer naivnog Bayesova klasifikatora prikazan je na slici 2.22a. Graf faktora za isti primjer nalazi se na slici 2.22b. Kod neusmjerenih grafičkih modela, uvjetno neovisni vrhovi ne



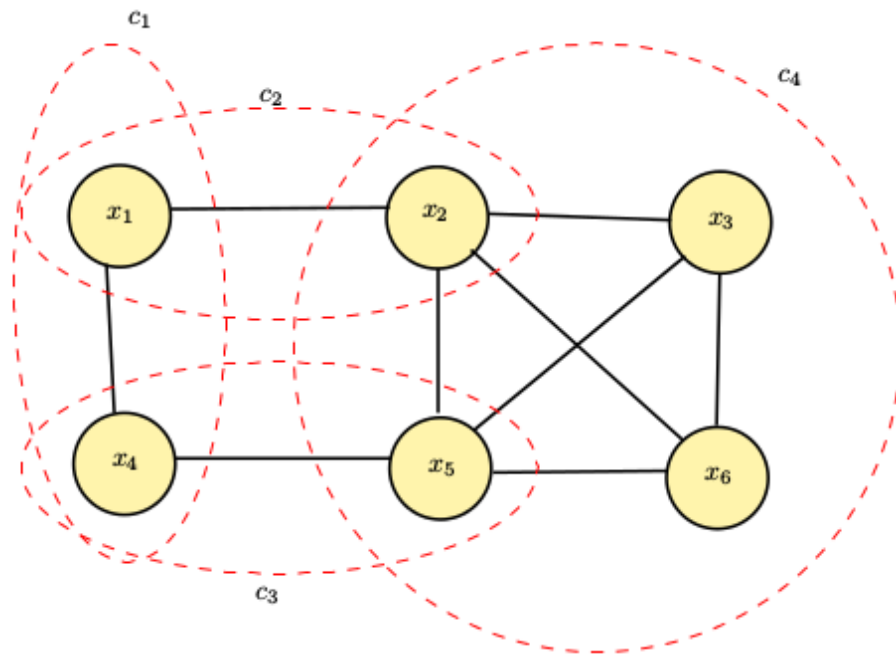
(a) Graf nezavisnosti naivnog Bayesova klasifikatora (b) Graf faktora naivnog Bayesova klasifikatora

Slika 2.22: Usmjereni grafički model

svrstavaju se u isti faktor, dakle pripadaju različitim klikama u grafu. Klik je pritom podskup grafa u kojem su svaka 2 vrha podskupa međusobno povezana bridom. Skup klika u grafu neka je označen sa \mathcal{C} , tada za distribuciju vjerojatnosti na vektoru varijabli \mathbf{v} vrijedi:

$$p(\mathbf{v}) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(\mathbf{v}_c) \quad (2.51)$$

Sa ψ_c smo označili potencijal klike, funkciju koja ovisi o vrhovima klike c , \mathbf{v}_c . Primjer neusmjerenog grafa sa označenim klikama je na slici 2.23.



Slika 2.23: Neusmjereni grafički model

Struktura slučajnog polja

Nakon definiranja parametara modela prometnice, moguće je sve ćelije smjestiti u strukturu slučajnog polja. Kako bismo mogli iscrtati rubnik kao granicu između ceste i pločnika, površine koje predstavljaju cestu i pločnik moraju biti precizno određene i za svaku se ćeliju mora znati kojoj od tih površina pripada. Praktično rješenje za modeliranje i kasnije označavanje ćelija je uvjetno slučajno polje (eng. Conditional Random Field, CRF). Uvjetna slučajna polja su statistički model s brojnim primjenama u strojnom učenju. Polja, prvi puta opisana u Lafferty et al. (2001), bitnu primjenu imaju u segmentaciji i označavanju podataka. Predstavljani kao nadogradnja na skrivene Markovljeve modele (Hidden Markov model, HMM), CRF ne sadrže pretpostavku neovisnosti sekvence podataka koji se procjenjuju i označavaju.

Slučajna varijabla \mathbf{X} neka predstavlja promatranu sekvencu. U ovom slučaju to su ćelije karte visina. Promatrana varijabla sekvence je visina ćelije izračunata pri izgradnji karte visina. Sa \mathbf{Y} možemo označiti slučajnu varijablu koja predstavlja klasu koju dodjeljujemo svakoj od ćelija. U konkretnom slučaju to će biti neka od oznaka: cesta, pločnik ili nepoznata oznaka. Slučajne varijable imaju zajedničku razdiobu, a model uvjetnih slučajnih polja nam omogućava definirati $p(\mathbf{Y}|\mathbf{X})$. Definiramo graf $G = (V, E)$, pri čemu u V postoji vrh v koji uključuje svaki Y_v iz \mathbf{Y} . Dakle vrh v uključuje sve elemente polja sa oznakom Y_v . Varijable Y_v moraju posjedovati Markov-

ljevo svojstvo na grafu G uz dani \mathbf{X} . Definiramo li $v \sim w$ kao relaciju susjedstva u grafu G , Markovljevo svojstvo za CRF glasi:

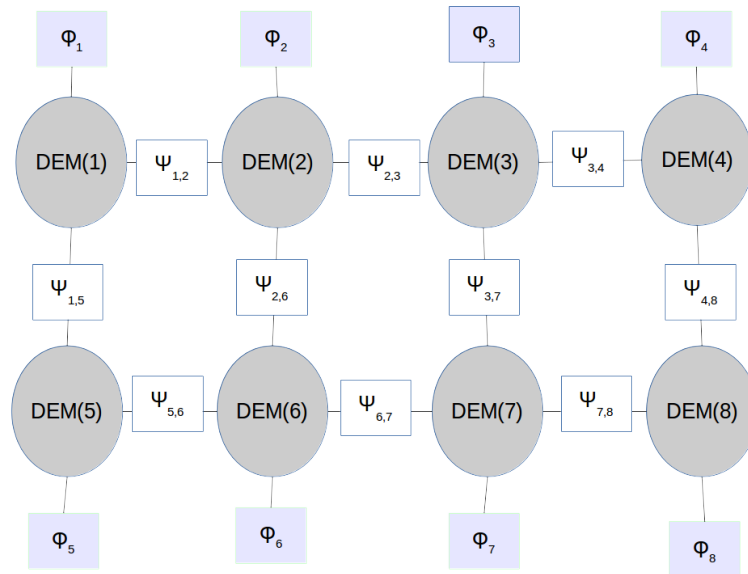
$$p(\mathbf{Y}_v | \mathbf{X}, \mathbf{Y}_w, w \neq v) = p(\mathbf{Y}_v | \mathbf{X}, \mathbf{Y}_w, w \sim v)$$

Navedenom jednadžbom osiguravamo da uz poznate varijable opservacije na grafu G varijabla \mathbf{Y}_v ne ovisi o susjednim varijablama \mathbf{Y}_w .

Jednostavan primjer modela uvjetnog slučajnog polja koji se koristi na karti visina je sljedeći:

- svaka ćelija karte visina je vrh u grafu G
- bridovi grafa G su susjedne ćelije s obzirom na relaciju 4-susjedstva

Razlog spajanja ćelija u 4-susjedstvu je činjenica da je veća vjerojatnost da susjedne ćelije imaju iste oznake. Iz navedenog slijedi da ćemo za određivanje zajedničke distribucije sekvence \mathbf{Y} uz poznati \mathbf{X} morati modelirati unarne potencijale (potencijale vrhova), vezane samo uz distribuciju jedne ćelije, te binarne (bridne) potencijale vezane uz ćelije brida. Konačna struktura uvjetnog slučajnog polja slična je onoj na slici 2.24. Potencijali vrhova označeni su kao φ_i , dok je potencijal brida koji spaja vr-



Slika 2.24: Struktura uvjetnog slučajnog polja na slici

hove (susjedne ćelije) i i j označen kao $\psi_{i,j}$. Konačno, uz parametre Θ možemo pisati:

$$p(\mathbf{y} | \mathbf{x}, \Theta) \propto \prod_{i \in V} \varphi_i(y_i | x_i, \Theta) \prod_{(i,j) \in E} \psi_{i,j}(y_i, y_j | x_i, x_j, \Theta) \quad (2.52)$$

Za skup parametara Θ , uvjetna ovisnost sekvence oznaka \mathbf{y} proporcionalna je umnošku potencijala vrhova iz skupa V , te potencijala bridova iz skupa E . Unarni potencijali (potencijali vrhova) mogu se rastaviti Bayesovim pravilom na sljedeće faktore:

$$\varphi_i(y_i|x_i, \Theta) \propto p(x_i|y_i, \Theta)p(y_i|\Theta)$$

Potencijal vrha, dakle ovisi o a priori vjerojatnosti $p(y_i|\Theta)$, te o izglednosti sekvence \mathbf{x} , $p(x_i|y_i, \Theta)$. Nakon što je struktura CRF potpuno definirana, na polju se može primijeniti neki algoritam zaključivanja koji će odrediti sekvencu oznaka s najvećom vjerojatnošću. Vjerojatnosti i potencijali opisani u ovom koraku za konkretan primjer će biti definirane u sljedećem potpoglavlju.

Uvjetno slučajno polje na karti visina

Na konkretnom polju imamo posebno definirane potencijale unarne potencijale ψ i binarne potencijale φ . Potencijali moraju osigurati da najveću vjerojatnost pridaju oznaci kojoj ćelija DEM-a stvarno pripada. Unarni potencijali modeliraju se za samo jednu varijablu, jednu ćeliju karte visina, dok binarni potencijali ovise i o susjednim ćelijama. Sekvencu oznaka ćemo označiti sa \mathcal{L} . Sekvenca oznaka predstavlja sekvencu klasa koju pridijeljujemo ćelijama, u prošlom pododjeljku označenu sa \mathbf{Y} . Za jedan korak određivanja najvećih vjerojatnosti, definiramo parametre označene sa Θ . Sekvencu visina pridijeljenih ćelijama karte visina označit ćemo sa \mathbf{h} , i ta sekvencu predstavlja opservacijsku varijablu u prošlom pododjeljku označenu sa \mathbf{X} . Klasifikacija svih ćelija definirana je kao:

$$\mathcal{L} = \arg \max_{\mathcal{L}} (p(\mathcal{L}|\mathbf{h}, \Theta))$$

Sada možemo definirati vjerojatnost u (2.52):

$$p(\mathcal{L}|\mathbf{h}, \Theta) \propto \prod_{i \in \text{DEM}} \varphi_i(l_i|h_i, \Theta) \prod_{(i,j) \in E} \psi_{i,j}(l_i, l_j|h_i, h_j, \Theta) \quad (2.53)$$

Pri tome uzimamo l_i kao oznaku ćelije i unutar karte visina, (i, j) kao brid u grafu (ćelije iz 4-susjedstva čine bridove), te h_i kao izmjerenu visinu ćelije i . Unarne potencijale u gornjoj formuli možemo faktorizirati kao:

$$\varphi_i(l_i|h_i, \Theta) \propto p(h_i|l_i, \Theta)p(l_i|\Theta) \quad (2.54)$$

Prvi faktor u (2.54) definira kolika je vjerojatnost da ćelija i ima izmjerenu visinu h_i uz oznaku l_i te parametre Θ . Drugi faktor predstavlja a priori vjerojatnost oznake l_i . A priori vjerojatnost definiramo jednadžbom (2.57). Faktori vjerojatnosti izmjerene

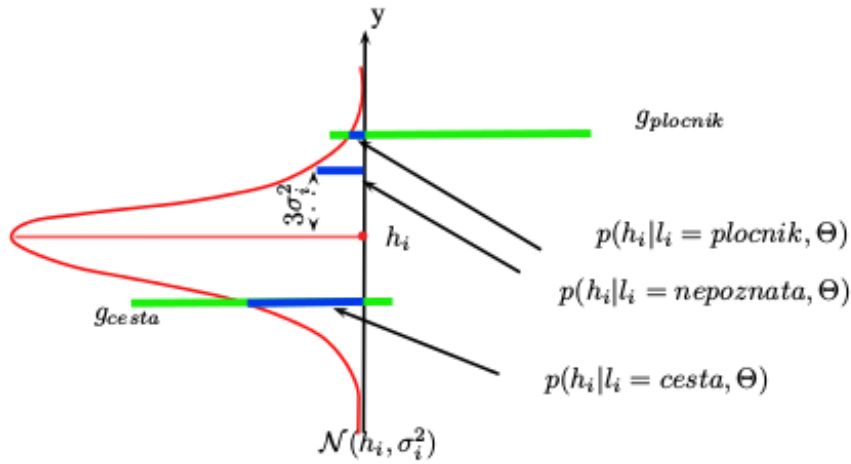
visine h_i uz danu oznaku l_i i parametre su, slično (Siegemund et al., 2010), definirani kao:

$$p(h_i|l_i, \Theta) = \begin{cases} f((g_{plocnik}(x_i, z_i)), h_i, \sigma_i^2) & \text{if } l_i = plocnik \\ f(g_{cesta}(x_i, z_i), h_i, \sigma_i^2) & \text{if } l_i = cesta \\ f(h_i + 3\sigma_i, h_i, \sigma_i^2) & \text{if } l_i = nepoznata \end{cases} \quad (2.55)$$

gdje je $f(\cdot)$ normalna (Gaussova distribucija) uz izmjerenu visinu h_i kao očekivanje, te varijancu σ_i^2 . Prema (2.55), vjerojatnost da je izmjerena visina h_i uz oznaku pločnika za ćeliju i računa se kao:

$$f((g_{plocnik}(x_i, z_i)), h_i, \sigma_i^2) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(g_{plocnik}(x_i, z_i) - h_i)^2}{2\sigma_i^2}\right) \quad (2.56)$$

Funkcije $g_{plocnik}$ i g_{cesta} određene su u koraku opisanom u odjeljku posvećenom estimiranju parametara modela prometnice. Pri tome nastojimo pridati najveću vjerojatnost oznaci čija funkcija površine najbolje opisuje izmjerenu visinu h_i . Ćelija ima najveću vjerojatnost da će biti označena kao nepoznata ukoliko je udaljenost od obje funkcije površine veća od 3σ . U ovisnosti o kvaliteti podataka u ćelijama, moguće je povećati i smanjiti faktor koji se množi sa devijacijom σ . Primjer vjerojatnosti klasifikacije prikazan je na slici 2.25.

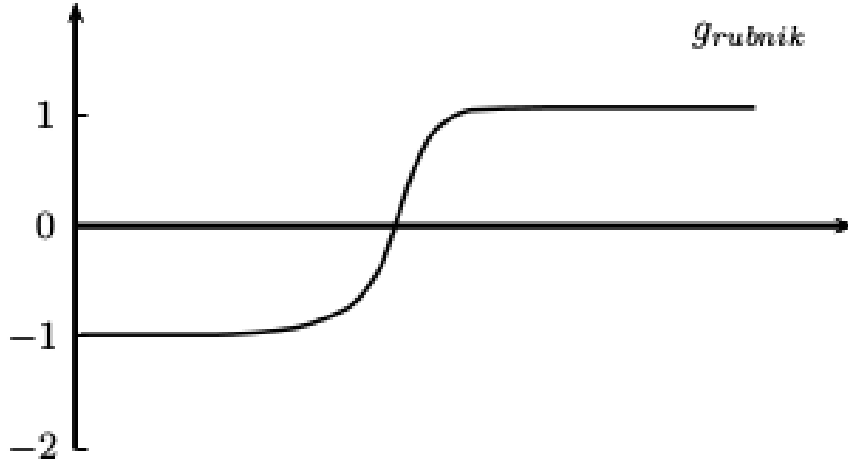


Slika 2.25: Vjerojatnost izmjerene visine h_i uz danu oznaku l_i i parametre Θ

Drugi faktor definiran u (2.54) predstavlja a priori vjerojatnosti oznaka l_i i računamo ga na sljedeći način:

$$p(l_i|\Theta) = \begin{cases} \frac{1}{\eta} |1 - g_{rubnik}(x_i, z_i)| & \text{ako } l_i = plocnik \\ \frac{1}{\eta} |1 - g_{rubnik}(x_i, z_i)| & \text{ako } l_i = cesta \\ \frac{1}{\eta} & \end{cases} \quad (2.57)$$

Pri tome je η normalizacijski faktor koji osigurava da zbroj a priori vjerojatnosti bude 1. Funkcija g_{rubnik} definirana je u koraku prezentiranom u odjeljku o estimiranju parametara modela prometnice. Slika navedene funkcije na 2.26 jasnije pokazuje odabir faktora $p(l_i|\Theta)$.



Slika 2.26: Funkcija g_{rubnik}

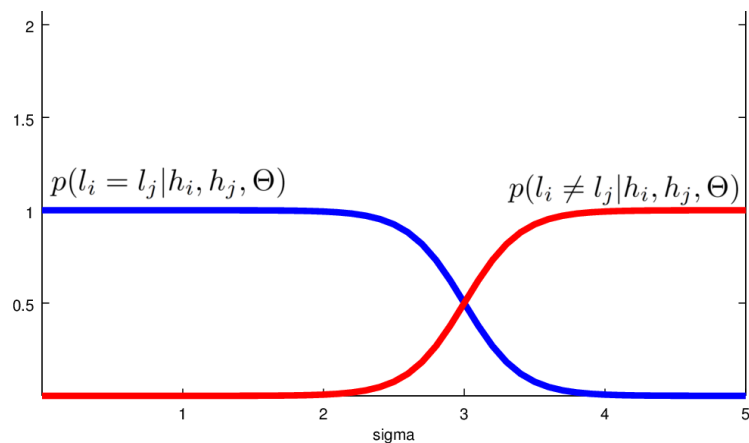
Također, model uvjetnih slučajnih polja na primjeru uključuje potencijale bridova. Funkcija vjerojatnosti binarnih potencijala mora osigurati da se susjedne ćelije dobiju različite oznake samo ako njihova razlika u visini prelazi određeni prag. Definiramo li izmjerenu visinu ćelije i sa h_i , te visinu ćelije j sa h_j , njihovu razliku računamo kao

$$d_{i,j} = |h_i - h_j|. \quad (2.58)$$

Prag razlike visine među ćelijama definiramo kao:

$$3\sigma_{i,j} = 3\sqrt{\sigma_i^2 + \sigma_j^2}$$

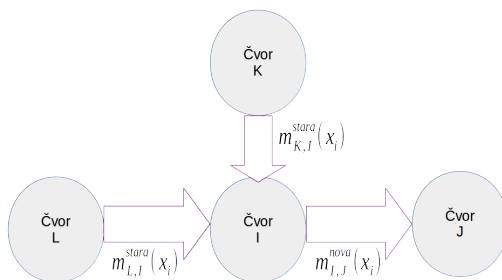
Vjerojatnost $\psi_{i,j}(y_i, y_j|x_i, x_j, \Theta)$ definirana je kao sigmoidalna funkcija s $d_{i,j}$ i $\sigma_{i,j}$ kao parametrima. Primjer možemo vidjeti na slici 2.27. Vjerojatnost da dvije susjedne ćelije i i j dobiju različitu oznaku uz izmjerene visine h_i i h_j označavamo sa $p(l_i \neq l_j|h_i, h_j, \Theta)$, dok je sa $p(l_i = l_j|h_i, h_j, \Theta)$ označena vjerojatnost istih oznaka susjednih ćelija. Crvenom je crtom označena vjerojatnost $p(l_i \neq l_j|h_i, h_j, \Theta)$, dok je plavom crtom označena vjerojatnost $p(l_i = l_j|h_i, h_j, \Theta)$. Vjerojatnosti su iscrtane u ovisnosti o razlici visina $d_{i,j}$. Kako bismo pronašli najvjerojatniju sekvencu oznaka, koristimo neki algoritam zaključivanja (engl. inference algorithm). U radu je korištena varijacija algoritma propagacije vjerojatnosti (engl. belief propagation), tzv. loopy belief propagation algoritam (algoritam propagacije vjerojatnosti za cikluse).



Slika 2.27: Modeliranje potencijala bridova, vjerojatnost različitih oznaka (crvena crta) i istih oznaka (plava crta) susjednih ćelija u ovisnosti o razlici njihove visine

Algoritam propagacije vjerojatnosti

Algoritam belief propagation predstavljen je u (Pearl, 1982). Algoritam propagacije također je poznat i kao sum-product message passing algoritam. Algoritam koristimo kao metodu inferencije na grafičkom modelu. Grafički model u radu je graf binarnih i unarnih faktora definiranih u prošlom pododjelku. Struktura grafa prikazana je na slici 2.24.



Slika 2.28: Ažuriranje poruka u algoritmu propagacije vjerojatnosti

Algoritam koristi poruke koje šalje među susjednim elementima uvjetnog slučajnog polja. Poruku od elementa (čvora) i do čvora j označit ćemo kao $m_{i,j}(x_j)$. Vrijednost x_j proporcionalna je vjerovanju čvora i o vrijednosti marginalne vjerojatnosti $P(x_j)$. U svakom koraku algoritma, poruke se ažuriraju na nove vrijednosti, s tim da nova poruka od čvora i do čvora j ovisi o svim porukama oblike $m_{k,i}(x_i)$ osim o poruci

$m_{j,i}(x_i)$. Način ažuriranja nove poruke predstavljen je jednadžbom:

$$m_{i,j}^{nova}(x_j) = \sum_{x_i} \psi_{i,j} \varphi_i \prod_{k \in E(i) \setminus j} m_{k,i}^{stara}(x_i) \quad (2.59)$$

Ažuriranje poruke može se vidjeti na slici 2.28. Dakle, koriste se sve poruke iz susjedstva čvora i osim poruke od čvora j . Kod ažuriranja poruka na strukturama sličnim slici, obično se koristi algoritam ažuriranja od gore prema dolje, s lijeva na desno. Vrijednosti poruka se obično normaliziraju, kako bi se omogućila sumacija vrijednosti marginalne vjerojatnosti. Kada vrijednosti poruka konvergiraju prema određenim uvjetima, možemo izračunati vjerovanja za svaku varijablu (ćeliju karte visina). Vjerovanje označujemo kao $b(x_i)$, te ga izračunavamo prema:

$$b(x_i) \propto \varphi_i \prod_{k \in E(i)} m_{k,i}(x_i) \quad (2.60)$$

Konačan rezultat je vjerojatnost oznaka u svakoj ćeliji karte visina.

3. Programska implementacija i ispitni skupovi

3.1. Ispitni skupovi

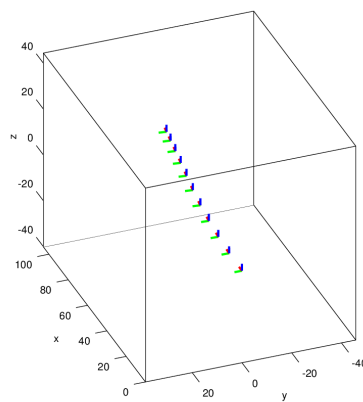
Za ispitivanje ispravnosti i točnosti programske implementacije korišteno je nekoliko video-sekvenci iz skupa KITTI, dijela projekta kojeg je izradio Karlsruhe Institute of Technology. Skup se sastoji od niza video snimki vožnji u različitim prometnim okruženjima. Zadaci sustava KITTI koncentrirani su na stereoskopski računalni vid, optički tok, odometriju te detekciju i praćenje 3D objekata. Skup podataka se pribavlja nizom senzora postavljenih na automobil. Snimke stvaraju dvije kamere koje su postavljene i kalibrirane za stvaranje stereo snimki. Postoje dvije kamere koje snimaju slike s razinama sive, te dvije kamere koje snimaju slike u boji visoke rezolucije. Osim toga podatke u bazu pribavlja i Velodyne laserski senzor te GPS sustav za lociranje. Platforma za snimanje prikazana je na slici 3.1.



Slika 3.1: Projekcija točke u prostoru na stereo slike

Slike su rektificirane i vremenski usklađene kako bi bile pogodne za stvaranje tro-

dimenzionalne scene. Dimenzije slika su 1242x375 piksela. Pri rekonstrukciji trodimenzionalne scene, uz slike stereo kamera, moguće je koristiti i niz 3D točaka koje detektira laserski skener. Takva bi metoda bila robusnija zbog veće točnosti i preciznosti dobivenih podataka. Podaci o GPS lokaciji vozila stvaraju se za svaki okvir snimke. Uz skup koordinata zemljopisne širine i dužine pozicije vozila dostupni su i podaci o ubrzanju i brzini vozila u više smjerova 3D prostora. Na osnovu tih podataka, moguće je odrediti pomake vozila u 3D koordinatnom sustavu, te izračunati matricu projekcije točaka između dviju okvira video snimke. Takvi podaci mogu se koristiti za rekonstrukciju putanje, kao što je prikazano na slici 3.2.



Slika 3.2: Putanja koordinatnog sustava automobila

Problem ove baze snimki koji je uočen pri izradi sustava detekcije rubnika nalazi se u nejednolikom osvjetljenju na slikama pribavljenim u vožnji. Dodatni korak korekcije osvjetljenja omogućio bi bolje rezultate pri izradi mape dispariteta.

3.2. Vanjske biblioteke

3.2.1. Biblioteka OpenCV

Programska implementacija ovog rada oslanja se među ostalim i na biblioteku OpenCV (engl. Open Source Computer Vision). Ova biblioteka otvorenog koda pruža niz metoda iz područja računalnog vida i obrade slika. Biblioteka je napisana u programskom jeziku C++, ali sadrži i sučelja za Python, C i Javu. Podržana je na većini bitnih operacijskih sustava. U radu se koriste brojne funkcije implementirane u sklopu OpenCV-a.

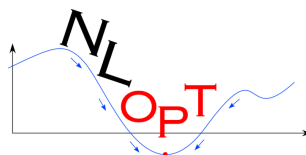


Slika 3.3: OpenCV logo

Koriste se metode za učitavanje i manipulaciju slikama u matričnom obliku. Pri analizi gradijenata koriste se funkcije biblioteke OpenCV vezane uz morfološke operacije, Canny detektor rubova i Houghovu transformaciju. Funkcije iz biblioteke se također koriste za sve potrebe vizualizacije rezultata, kao i za potrebe ručnog označavanja rubnika. OpenCV je korišten i za potrebe pronalaska podudarnih slikovnih elemenata funkcijom za poluglobalno podudaranje. Moduli za optimizaciju i paralelizaciju algoritama su u razvoju i mogli bi donijeti bitna poboljšanja u implementaciji rješenja.

3.2.2. Biblioteke optimizacije parametara površina

U svrhu optimizacije koeficijenata površina implementirana su rješenja sa dvije različite biblioteke. NLOpt je biblioteka otvorenog koda za nelinearnu optimizaciju. Pruža sučelje prema nizu programskih jezika. Implementira brojne algoritme optimizacije i pruža mogućnost optimizacije uz veliki broj parametara i podataka. U okviru funkcija ove biblioteke moguće je postaviti ograničenja na parametre u vidu jednadžbi ili nejednadžbi, te je moguće koristiti algoritme globalne i lokalne optimizacije. U implementaciji se koristi metoda Controlled Random Search s lokalnim mutacijama koja pokazuje sličnost metode evolucijskom algoritmu. Metoda je definirana u (Price, 1983) i temelji se na definiranju početne populacije parametarskih vektora te pronalasku najboljeg vektora parametara nakon nekoliko koraka izmjene populacije. Najbolji vektor parametara daje najmanju vrijednost funkcije greške definirane u odjeljku 2.4.1.



Slika 3.4: NLOpt logo

Druga biblioteka je biblioteka otvorenog koda Ceres Solver. Implementacije su napisane u programskom jeziku C++. Sadrži rješenja za modeliranje i rješavanje većih, kompleksnijih problema optimizacije. Modeliranje problema je implementirano sa jednostavnim sučeljem, te omogućuje uporabu gradijenata u optimizaciji. Moguće je modelirati specifične funkcije gubitka, robusnije od obične metode najmanjih kvadrata. Implementiran je niz algoritama za optimizaciju: Levenberg-Marquardt, QR faktorizacija, Cholesky faktorizacija, Schurov komplement... Dijelovi koda su paralelizirani i optimizirani. Moguć je ispis iscrpnih izvješća o koracima i napredovanju optimizacijskog algoritma.

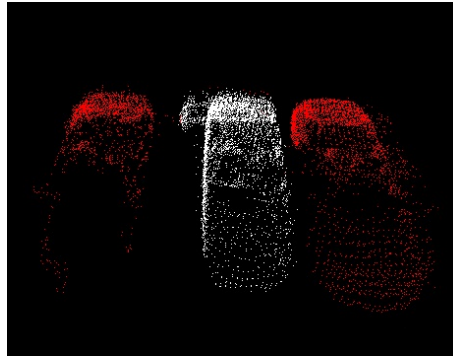
3.2.3. Biblioteka za grafičke modele

Za implementaciju grafičkog modeliranja problema, te algoritma propagacije vjerojatnosti koristi se biblioteka OpenGM, biblioteka otvorenog koda napisana u programskom jeziku C++. Fokus ove biblioteke su diskretni modeli grafa faktora te operacije na istima. Grafovi faktora su opisani u pododjelku 2.4.2. Implementiran je niz metoda zaključivanja (engl. inference method), uključivo s algoritmima propagacije poruka. Grafovi faktora modelirani ovom bibliotekom mogu imati proizvoljan broj faktora, varijabli i oznaka. Struktura grafa nije ograničena, a postoji i izbor između raznih metoda zaključivanja. Svaki faktor moguće je modelirati kao eksplicitnu funkciju (izlaz definiran od strane programera) ili korištenjem implementiranih funkcija. Iz ove je biblioteke korišten algoritam propagacije vjerojatnosti za cikluse (engl. loopy belief propagation algorithm), te je modeliran graf s unarnim i binarnim potencijalima.

3.2.4. Biblioteka PCL

Biblioteka PCL (engl. Point Cloud Library) služi za modeliranje i manipulaciju oblakom točaka. Implementira metode za obradu dvodimenzionalnih i trodimenzionalnih slika i oblaka točaka. Podržana je na nekoliko operacijskih sustava, a moguće je i posediti biblioteku za izračune na grafičkoj jedinici računala. Sadrži funkcije za vizualni prikaz izgrađenih oblaka točaka. Moduli biblioteke implementiraju razna rješenja za

segmentaciju, filtriranja oblaka točaka, detekciju i praćenje objekata...



Slika 3.5: Oblak točaka

Funkcije implementirane u biblioteci PCL su korištene za izgradnju oblaka točaka, te za izgradnju karte visina. Moguće je definirati posebni tip točaka za oblak. Moguće je definirati koje koordinate i varijable vezane uz točku želimo bilježiti. Primjer oblaka točaka izrađenog uz pomoć PCL biblioteke dan je na slici 3.5

3.3. Programska implementacija

Programsko rješenje zadatka rada implementirano je u C++ programskom jeziku. Koristi se nekoliko vanjskih biblioteka, za potrebe lakše implementacije složenih struktura koje su potrebne za detekciju.

Program je podijeljen u nekoliko modula. Postoji dio programa za izgradnju mape dispariteta. Programski kod tog dijela se temelji na implementaciji algoritama u vanjskoj biblioteci OpenCV i preuzet je iz implementacije (Braut, 2014).

Jedan modul programa bavi se detekcijom rubnika metodom detekcije ruba. U tom dijelu programa karta visina se izgrađuje podjelom oblaka točaka po x i z koordinati. Nakon toga vrši se dilatacija, detekcija rubova i Houghova transformacija. Houghova transformacija spaja manje pravce sličnog smjera sa malim razlikama u poziciji. Među pronađenim pravcima izabire se najdulji, te iscrtava na slici.

Modul programa za detekciju rubnika modeliranjem prometnice i klasifikacijom karte visina podijeljen je u nekoliko datoteka. U prvom koraku se izgradi karta visina iz oblaka točaka podjelom po stupcima slike i disparitetu. Nakon toga na određenim z koordinatama se traže točke sa najvišom razlikom u visini u odnosu na točku lijevo od navedene. Zabilježavaju se x i z koordinate tih točaka i zatim se procjenjuje pravac koja spaja točke. Procjenjuju se koeficijenti a i b pravca. Za pravac mora vrijediti

$z = a + bx$. Procjenjujemo koeficijente jednadžbama (2.41). Tim se pravcem ocjenjuje inicijalna granica između pločnika i ceste. Pravac služi za inicijalizaciju oznaka pločnika i ceste. Točke koje su inicijalno označene kao dio ceste služe za estimaciju parametara površine ceste, a isto vrijedi i za pločnik. Vršiti se projekcija točaka u uvjetno slučajno polje i propagacija vjerojatnosti. Zabilježava se nova podjela ćelija karte visina. Ukoliko jedna od površina zauzima puno veći dio ćelija od druge površina, ponavlja se estimacija parametara krivulje i površina. Ponavlja se i stvaranje uvjetnog slučajnog polja i algoritam zaključivanja. Konačno, ako razlike u visini između točaka gornje i donje krivulje rubnika jako variraju, rubnik se ne iscrtava.

Dodan je i modul za testiranje. Modul sadrži program za ručno označavanje rubnika, te program za izračun i zapisivanje greške. Također, moguće je programom izabrati okvire snimke koje ćemo analizirati u testiranju. Uz sve navedeno, dodane su i upute za korištenje.

4. Eksperimentalni rezultati

Preciznost i točnost rješenja testirana je na video snimkama iz KITTI ispitnog skupa. Snimke se sastoje od niza okvira pribavljenih stereo parom kamera. Izabrano je po nekoliko okvira iz snimke za analizu. Za svaki analizirani okvir gornja i donja krivulja rubnika se ručno označavaju. Testni program prikazuje okvir snimke, te traži označavanje nekoliko točaka. Više točaka znači točniju aproksimaciju krivulja. Nakon što se nekoliko točaka označi za svaki okvir, koordinate točaka se zapisuju u datoteku. Nakon toga krivulje se interpoliraju kubičnim splineom. U drugu datoteku su zapisane točke rubnika određene nekom od metoda implementiranih u okviru rada.

Kod rekonstrukcije ruba analizom gradjenata karte visina za svaki okvir se zapisuju koordinate početne i krajnje točke najdulje linije pronađene Houghovom transformacijom. Iz ručno označene gornje i donje krivulje, interpolira se krivulja samog rubnika, kao sredina između gornje i donje krivulje. Razmatraju se početne i krajnje točke programom označenog rubnika. Za retke (y koordinate) tih točaka se interpolacijom srednja krivulja rubnika određuju stupci (x koordinate) krivulje rubnika. Stupci početne i krajnje točke određene programom se uspoređuju sa stupcima interpolirane ručno označene krivulje. Za svaki okvir se računa razlika obiju točaka u x koordinati.

Kod rekonstrukcije ruba korištenjem uvjetnog slučajnog polja ocijenjuju se točke gornje i donje krivulje ocijenjene aplikacijom. Za svaku od y koordinata učitanih točaka (y koordinata je stupac slike) splineom se interpolira x koordinata (redak slike) ručno označene krivulje. Nakon toga se računa razlika x koordinate ručno označene krivulje i točke procijenjene aplikacijom.

Rezultati pokazuju zadovoljavajuću preciznost na jednoliko osvijetljenim slikama sa jasno vidljivim rubnikom. Problemi se pojavljuju kada je snimljena slabo osvijetljena okolina ili se pojavljuju prepreke i šum na slici. Tada se može dogoditi da dobijemo nedovoljno preciznu mapu dispariteta algoritmom podudaranja, te nije moguće točno odrediti razgraničenje između pločnika i ceste. Primjer je na slici 4.1.

Dana je i vremenska analiza izvođenja programa bez rekonstrukcije trodimenzionalne scene. Kako algoritmi nisu pripremljeni za rad na paralelnim procesnim je-



Slika 4.1: Problem nedovoljne osvijetljenosti okoline

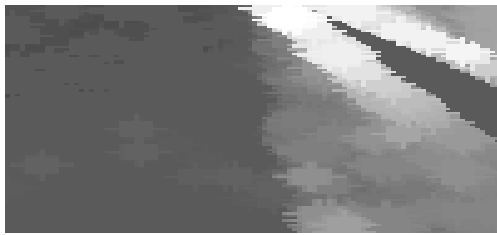
dicama, programi su testirani na jednoj jezgri Intel® Core™ i7-3610QM procesora sa frekvencijom 2.3GHz. Za metodu rekonstrukcije analizom gradjenata dana su prosječna vremena izvođenja za konstrukciju karte visina i cjelokupnu rekonstrukciju rubnika iz oblaka točaka. Za metodu rekonstrukcije korištenjem uvjetnog slučajnog polja dana su prosječna vremena konstrukcije karte visina, ostalih koraka algoritma te cjelokupne rekonstrukcije rubnika iz oblaka točaka. Detekcija rubnika u realnom vremenu trenutno nije moguća. Analiza rezultata pokazuje da kod obje metode najviše vremena oduzima izračun karte visina. Moguća rješenja tog problema su predložena u zaključku rada.

4.1. Rezultati rekonstrukcije rubnika analizom gradjenata karte visina

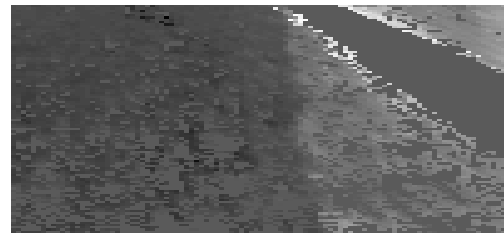
Implementirana je aplikacija koja redom analizira okvire sekvence slika, te na njima iscrtava pronađeni rubnik. Najprije se vrši stvaranje karte visina, zatim se karta visina podvrgava obradi medijan filtrom i dilatacijom. Na takvoj slici pronađemo gradijenti Cannyjevim detektorom, te najduži pravac pronađemo Houghovom transformacijom. Na slici 4.2 vidljivi su koraci rada algoritma.

Rezultati pokazuju da je ova metoda osjetljiva na problem lažnih rubnika. Analiza provedena na ispitnom skupu KITTI pokazuje nekoliko lažno pozitivnih klasifikacija. Problem nastaje kada na slici ne postoji jasno vidljiv rubnik. Tada se u fazi analize gradjenata mogu naći lažni rubnici zbog grešaka u rekonstrukciji trodimenzionalne scene i šuma na slici. Lažno pozitivna detekcija vidljivo je iz slike 4.3. Također se pojavljuje problem kod slika na kojima pločnik graniči sa nekom površinom veće prosječne visine. Tada se kao rubnik može rekonstruirati granica između pločnika i njemu susjedne površine, a ne granica ceste i pločnika. Primjer takvog problema prikazan je na slici 4.4.

Problemi se pojavljuju i kada postoje prepreke između vozila i pločnika. Ipak,



(a) Karta visina



(b) Karta visina bez dilatacije



(c) Pronađeni gradijenti



(d) Pronađeni gradijenti na filtriranoj slici



(e) Rekonstruirani rubnik

Slika 4.2: Koraci u rekonstrukciji rubnika analizom gradijenta karte visina



Slika 4.3: Lažno pozitivne rekonstrukcije



Slika 4.4: Primjer problema površine susjedne pločniku

postoje primjeri kada je prepreka detektirana i označena kao rubnik i time ograničeno pronađeno prohodno područje, kao na slici 4.5

Uspješno rekonstruirani rubnici prikazani su na slikama 4.6 i 4.7.

U tablici 4.1 su pokazana odstupanja rubnika pronađenih implementacijom od



Slika 4.5: Prepreke označene kao rubnik



Slika 4.6: Uspješna detekcija



Slika 4.7: Uspješna detekcija

ručno označenih rubnika. Greške su izražene u prosječnom odstupanju u pikselima, kao i odstupanju u pikselima za obje točke, kako je definirano na početku ovog odjeljka rada. Ukupna greška za snimke se računaju kao zbroj prosječnih grešaka po okvirima snimke, a prosječna greška je ukupna greška podijeljena sa brojem analiziranih okvira. U obzir su uzeti samo okviri na kojima postoje vidljivi rubnici.

	Prosječna greška u pikselima	Ukupna greška u pikselima (zbroj prosječnih grešaka)
Snimka 1	11.139	233.919
Snimka 2	17.5484	175.48451
Ukupno	13.207	409.403

Tablica 4.1: Preciznost rekonstrukcije rubnika analizom gradijenata karte visina, odstupanja u pikselima

Tablica 4.2 pokazuje prosječno vremensko izvođenje algoritma na računalu. Algoritam je za vremensko testiranje pokrenut na 18 različitih okvira snimke iz ispitnog skupa.

Prosječno vrijeme konstrukcije karte visina	Prosječno vrijeme rekonstrukcije rubnika
0.3534	0.35448

Tablica 4.2: Vremenska analiza rekonstrukcije rubnika analizom gradijenata karte visina, odstupanja u pikselima

4.2. Rezultati rekonstrukcije rubnika korištenjem uvjetnog slučajnog polja

Program je testiran na nekoliko video snimaka. Za svaku od snimki nasumično je odabran dio okvira, te je pronađen rubnik opisanom metodom. Primjer pronađenog rubnika vidljiv je na slici 4.8. Za svaki okvir slike određuju se gornja i donja krivulja rubnika. Između pronađenih točaka krivulje funkcijom biblioteke OpenCV se iscrtava njihova spojnica.



Slika 4.8: Primjer označenog rubnika

U početnoj fazi se granica između oznaka ceste i oznaka pločnika definira kao x koordinata najveće visinske razlike na nekoliko točaka po z koordinati. Nakon toga, prema tim točkama s koordinatama (x, z) interpolira se pravac, kao prema slici 4.9. Odabrane su dvije sekvence. Ukupno je analizirano 63 okvira snimke, 43 u jednoj sek-



Slika 4.9: Početno označavanje granice interpolacijom pravca

venci, te 20 u drugoj sekvenci ispitnog skupa. U prvoj snimci lažno negativnih detekcija ima 5, dok u drugoj postoji 1 lažno negativna detekcija. Uspješnost rekonstrukcije

ocijenjena je kroz nekoliko mjera. Točnost (engl. Accuracy) definira se kao:

$$Tocnost = \frac{Okviri\ sa\ uspjesnom\ rekonstrukcijom\ rubnika}{Svi\ okviri} = \frac{57}{63} = 90.48\%$$

Odziv (engl. recall) ocijenjen je:

$$Odziv = \frac{Pronadeni\ postojeći\ rubnici}{Postojeći\ rubnici} = \frac{56}{62} = 90.32\%$$

Postoji niz problema na analiziranim okvirima snimke. Glavni problem je u nedovoljnoj preciznosti pri rekonstrukciji trodimenzionalne scene. Ukoliko se svaka od površina ocjenjuje sa 4 koeficijenta, nastaje problem da algoritam optimizacije ne može dovoljno dobro ocijeniti površine koje imaju značajnije varijacije u visini. Stoga algoritam zaključivanja na uvjetnom slučajnom polju ne može pronaći jasnu granicu između ceste i pločnika. Veći broj koeficijenata za estimaciju površine obično dovodi do toga da je optimizacija istih vremenski zahtjevnija, te koeficijenti površine previše ovise o vanpopulacijskim točkama. Kako se mora naći kompromis između većeg broja parametara i dobre procjene parametara, funkcija krivulje rubnika može detektirati samo rubnike predstavljene pravcem ili krivuljom čiji kvadratni i kubni članovi nemaju veća odstupanja od nule. Primjer precizno detektiranih krivulja koji nemaju veće kubne i kvadratne parametre dan je na slikama 4.10 i 4.11



Slika 4.10: Krivulje sa manjim kubnim i kvadratnim članovima



Slika 4.11: Krivulje sa manjim kubnim i kvadratnim članovima

Zbog nedovoljno precizno određenih parametara, postoji i problem kada visina pločnika varira na jednom okviru. Dobro rekonstruiran rubnik uz primjer navedenog

problema dan je na slici 4.12. Na nekim okvirima postoje veća odstupanja zbog male



Slika 4.12: Dobro rekonstruiran rubnik uz problem nejednolike visine pločnika

razlike u visini između pločnika i prometnice, kao na slici 4.13.



Slika 4.13: Loše označen rubnik zbog male visinske razlike pločnika i ceste

Okviri sa najmanjim odstupanjima od ručno označenih primjera dani su na slikama 4.14 i 4.15.



Slika 4.14: Precizna rekonstrukcija rubnika



Slika 4.15: Precizna rekonstrukcija rubnika

Na slici 4.17 vidi se konačni raspored oznaka ćelija karte visina na okviru. Slika 4.16 pokazuje iscrtane procijenjene površine ceste i pločnika.



Slika 4.16: Estimirane površine



Slika 4.17: Oznake ćelija karte visina, bijelo za pločnik, crno za cestu

Tablica 4.3 prikazuje greške po krivuljama, okvirima, snimkama i ukupna odstupanja u pikselima između ručno označenih i programski određenih rubnika. Način izračuna greške opširnije je objašnjen u početku ovog odjeljka rada. Ukupna greška za snimke računa se kao zbroj prosječnih grešaka na obje krivulje. Analizom tablica 4.1 i 4.3 vidljivo je da rekonstrukcija rubnika korištenjem uvjetnog slučajnog polja daje kvalitetnije i preciznije rezultate.



(a) Ručno označen rubnik



(b) Programski rekonstruiran rubnik

Slika 4.18: Razlika ručno i programski rekonstruiranog rubnika

	Prosječna greška u pikselima	Ukupna greška u pikselima (zbroj prosječnih grešaka)
Obje krivulje	4.0015	224.08
Gornja krivulja	3.9279	219.96
Donja krivulja	4.0751	228.21
Snimka 1	3.7635	139.25
Snimka 2	4.4649	84.834

Tablica 4.3: Preciznost rekonstrukcije rubnika korištenjem uvjetnog slučajnog polja, odstupanja u pikselima

Tablica 4.4 pokazuje prosječno vremensko izvođenje algoritma na računalu. Algoritam je za vremensko testiranje pokrenut na 22 različita okvira snimke iz ispitnog skupa. Usporedbom sa tablicom 4.2 vidimo da je veće prosječno vrijeme konstrukcije karte visina kod prve metode. Takav rezultat je posljedica veće rezolucije ćelija karte visina u metodi rekonstrukcije rubnika analizom gradijenata. U obje metode, najveći dio vremena oduzima upravo konstrukcija karte visina.

Prosječno vrijeme konstrukcije karte visina	Prosječno vrijeme estimiranja parametara modela i propagacije vjerojatnosti	Prosječno vrijeme rekonstrukcije rubnika
0.22965	0.0865	0.316175

Tablica 4.4: Vremenska analiza rekonstrukcije rubnika analizom gradijenata karte visina, odstupanja u pikselima

5. Zaključak

Cilj ovog rada bilo je implementirati robusne i precizne metode detekcije rubnika na sekvenci slika. Metode bi bile dio šireg sustava za pomoć u vožnji. Implementirane su dvije metode. Nakon početnih koraka rekonstrukcije trodimenzionalne scene i stvaranja karte visina, prvom metodom analiziramo gradijente karte visina te među gradijentima pronalazimo pravce kao model rubnika. U drugoj metodi nakon početnih koraka estimiramo parametre modela prometnice te rekonstruiramo rubnik korištenjem uvjetnog slučajnog polja. Prva metoda pokazuje manju preciznost u rekonstrukciji rubnika. Ta je metoda većim dijelom temeljena na algoritmu predloženom u radu (Oniga et al., 2007), dok je rješenje drugom metodom temeljeno na članku (Siegemund et al., 2010). Implementirana rješenja iscrstavaju rubnik kao granicu između ceste i pločnika, nakon čega je moguće rezultate usporediti sa ručno označenim rubnicima. Rezultati pokazuju nekoliko mana u predloženim rješenjima, ali pružaju zadovoljavajuću točnost na primjerima koji nisu prekompleksni.

Problem nastaje kod nedovoljno guste i precizne mape dispariteta. Kako bi program detektirao promjenu u visini između ceste i pločnika, potrebno je imati precizno rekonstruiranu trodimenzionalnu scenu. U slučaju nedovoljno točno rekonstruirane scene pojavljuju se lažne detekcije rubnika i lažno negativni rezultati.

Jedan od problema svakako je brzina izvođenja programa. Kao što je već pojašnjeno u odjeljku 4, detekcija nije prilagođena radu u stvarnom vremenu. Velik dio vremena se troši na podudaranje slikovnih elemenata i rekonstrukciju trodimenzionalne scene. Jedno od rješenja je iskoristiti sklopovsku podršku za ubrzanje navedenog dijela algoritma. Tehnika predložena u (Gehrig et al., 2009) spada u popularna i kvalitetna rješenja. U tome se članku jasno prezentira izgradnja sklopovske podrške za zadatak poluglobalnog podudaranja sa brzim vremenom izvođenja. Brzina izvođenja može se poboljšati i paralelizacijom algoritma. U prethodnom odjeljku je zaključeno da najveći dio vremena nakon rekonstrukcije scene uzima stvaranje karte visina. Budući radovi na ovom problemu mogli bi se posvetiti problemu paralelizacije tog dijela algoritma. Stvaranje grafa faktora također je lako prilagoditi paralelnom izvođenju.

Detekcija rubnika implementiranim metodama nije dala zadovoljavajuće rezultate na okvirima snimke koji imaju problem nejednolike osvjetljenosti ili većeg šuma na slici, kao što je prikazano u prethodnom odjelku. Rješenje koje bi omogućilo brže izvođenje metode rekonstrukcije korištenjem uvjetnog slučajnog polja, kao i točniju rekonstrukciju rubnika, predloženo je u (Siegemund et al., 2011). U tome se članku predlaže uvođenje ovisnosti između okvira snimke, čime vjerojatnost oznaka za ćelije karte visine ovisi i o prethodnom okviru. Članak navodi da takvo vremensko filtriranje omogućuje precizniju rekonstrukciju u kraćem vremenu izvođenja. Implementacija vremenskog filtriranja je započeta, ali nije uključena u konačno programsko rješenje zadatka. Također, jedno od rješenja za rekonstrukciju drugom metodom je robusniji algoritam zaključivanja na uvjetnom slučajnom polju. Iako metode predložene u (Maye et al., 2012) i (Siegemund et al., 2010) jamče konvergenciju algoritma propagacije vjerojatnosti u malom broju iteracija, rezultati algoritma uvelike ovise o početnim označavanju ćelija opisanom u pododjeljku 2.4.1. Dodatnim testiranjem uvidjeli smo mogućnost pronalaska inicijalnog označavanja algoritmom k-srednjih vrijednosti, u slučaju kad ne postoji informacija o lokaciji površina koje se estimiraju.

Metoda detekcije pravaca među gradijentima mogla bi se poboljšati detekcijom jednostavnijih krivulja. Iako druga metoda obrađena u radu ima velikog potencijala, potrebna su poboljšanja u vidu vremenskog filtriranja i robusnijeg algoritma zaključivanja kako bi metoda bila uspješna na širem rasponu problema i različitim ispitnim skupovima.

LITERATURA

- Romuald Aufrere, Christoph Mertz, i Charles Thorpe. Multiple sensor fusion for detecting location of curbs, walls, and barriers. U *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, stranice 126–131. IEEE, 2003.
- Hernán Badino, Uwe Franke, i Rudolf Mester. Free space computation using stochastic occupancy grids and dynamic programming. U *Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil*, svezak 20, 2007.
- Viktor Braut. Detekcija objekata u gustoj stereoskopskoj rekonstrukciji. Magistarski rad, Fakultet elektrotehnike i računarstva, Zagreb, Hrvatska, 2014.
- Alberto Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.
- Stefan K Gehrig, Felix Eberli, i Thomas Meyer. A real-time low-power stereo vision engine using semi-global matching. U *Computer Vision Systems*, stranice 134–143. Springer, 2009.
- Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. U *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, svezak 2, stranice 807–814. IEEE, 2005.
- John Lafferty, Andrew McCallum, i Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.
- Jerome Maye, Ralf Kaestner, i Roland Siegwart. Curb detection for a pedestrian robot in urban environments. U *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, stranice 367–373. IEEE, 2012.
- Joel C McCall i Mohan M Trivedi. Video-based lane estimation and tracking for driver

assistance: survey, system, and evaluation. *Intelligent Transportation Systems, IEEE Transactions on*, 7(1):20–37, 2006.

Florin Oniga, Sergiu Nedevschi, i Marc Michael Meinecke. Curb detection based on elevation maps from dense stereo. U *Intelligent Computer Communication and Processing, 2007 IEEE International Conference on*, stranice 119–125. IEEE, 2007.

Florin Oniga, Sergiu Nedevschi, i Marc Michael Meinecke. Curb detection based on a multi-frame persistence map for urban driving scenarios. U *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, stranice 67–72. IEEE, 2008.

Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. U *AAAI*, stranice 133–136, 1982.

WL Price. Global optimization by controlled random search. *Journal of Optimization Theory and Applications*, 40(3):333–348, 1983.

Daniel Scharstein i Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

Jan Siegemund, David Pfeiffer, Uwe Franke, i W Forstner. Curb reconstruction using conditional random fields. U *Intelligent Vehicles Symposium (IV), 2010 IEEE*, stranice 203–210. IEEE, 2010.

Jan Siegemund, Uwe Franke, i Wolfgang Forstner. A temporal filter approach for detection and reconstruction of curbs and road surfaces based on conditional random fields. U *Intelligent Vehicles Symposium (IV), 2011 IEEE*, stranice 637–642. IEEE, 2011.

Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st izdanju, 2010. ISBN 1848829345, 9781848829343.

Riccardo Turchetto i Roberto Manduchi. Visual curb localization for autonomous navigation. U *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, svezak 2, stranice 1336–1342. IEEE, 2003.

Gangqiang Zhao i Junsong Yuan. Curb detection and tracking using 3d-lidar scanner.
U *Image Processing (ICIP), 2012 19th IEEE International Conference on*, stranice
437–440. IEEE, 2012.

Pronalaženje prohodnog tla stereoskopskim računalnim vidom

Sažetak

Rad se bavi tehnikama detekcije rubnika kao dijelom sustava za pomoć u vožnji. Ulazni podaci u implementaciji su slike pribavljene stereo parom kamera. Implementirane su metode rekonstrukcije trodimenzionalne scene iz slika u strukturu oblaka točaka. Ta se struktura projicira u kartu visina. Nakon što je konstruirana karta visina, moguće je koristiti dvije metode detekcije. Jednostavna metoda detektira rubove na karti visina, te među rubovima traži pravce koje iscrtava kao rubnike. Druga metoda uključuje estimiranje parametara modela prometnice, tj. definiciju površina ceste i pločnika i krivulje rubnika kao granice među njima. Rezultate dobivamo nakon projekcije karte visina u uvjetno slučajno polje i klasifikacije. Rubnik se iscrtava na originalnim slikama. Prezentirani su rezultati eksperimenata na ispitnim skupovima.

Ključne riječi: stereo vid, detekcija rubnika, oblak točaka, karta visina, rekonstrukcija 3D scene, uvjetno slučajno polje

Detecting traversable terrain by stereoscopic computer vision

Abstract

This thesis is concerned with techniques of curb detection, as a part of a advanced driver assistance system. Input to the algorithm implementation are pictures acquired by a stereo rig on a vehicle. Thesis includes implementation of methods for extracting the three-dimensional information from the pictures into a point cloud structure. Point cloud is then projected into a digital elevation map. After building the elevation map, it is possible to make use of any of two methods for curb detection. The simpler method detects edges on the digital elevation map and locates lines in edge map. Those lines are projected back to the image and represent curbs. The other method includes parameter estimation for the environment model, i.e. defining the surfaces of the road and sidewalk and defining the curb curve as an edge between the two surfaces. Results are acquired after projecting the elevation map into a conditional random field and using an inference algorithm for classification. The drawings of the curb are then added to the input images. Results of testing the algorithms are also presented.

Keywords: stereo vision, curb detection, point cloud, digital elevation map, 3D reconstruction, conditional random field