

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 693

Detekcija objekata u gustoj stereoskopskoj rekonstrukciji

Viktor Braut

Zagreb, lipanj 2014.

Umjesto ove stranice umetnite izvornik Vašeg rada.

Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik.

*Zahvaljujem svima koji su svojim savjetima i podrškom pridonijeli izradi ovoga
diplomskog rada, posebice mentoru prof. dr. sc. Siniši Šegviću*

SADRŽAJ

Popis slika	vi
Popis tablica	viii
1. Uvod	1
2. Algoritmi i matematičke metode	2
2.1. Stereo vid	2
2.1.1. Model kamere	3
2.1.2. 3D rekonstrukcija	6
2.1.3. Određivanje strukture iz stereo slika	7
2.1.4. Mjere sličnosti	13
2.1.5. Lokalne metode podudaranja	14
2.1.6. Globalne metode podudaranja	15
2.2. Konsenzus slučajnih uzoraka	18
2.2.1. Određivanje vrijednosti parametara algoritma	20
3. Ispitni skupovi	22
3.1. Ispitni skup KITTI	22
3.2. Ispitni skup EISATS	22
4. Programska izvedba i vanjske biblioteke	24
4.1. Biblioteka OpenCV	24
4.2. Biblioteka PCL	25
4.3. Formiranje oblaka točaka	26
4.4. Estimiranje ravnine ceste	26
4.5. Rešetkasta podjela oblaka točaka	27
4.6. Grupiranje blokova u objekte	30
4.7. Prikaz objekata na slici	30

5. Eksperimentalni rezultati	32
6. Zaključak	38
Literatura	39

POPIS SLIKA

2.1.	Model kamere s točkastim otvorom	3
2.2.	Shema kamere i koordinatnog sustava slike	4
2.3.	Odnos koordinatnih sustava prostora, kamera i slika	4
2.4.	Odnos koordinatnih sustava prostora, kamera i slika	4
2.5.	3D rekonstrukcija iz 2D slika	6
2.6.	Tipiči sustav stereo kamera	7
2.7.	3D rekonstrukcija triangulacijom	9
2.8.	Epipolarna ravnina	10
2.9.	Rektificirana triangulacija	13
2.10.	Prikaz disparitetne mape izračunate BM metodom	15
2.11.	minimizacija energije duž putanja r	17
2.12.	Prikaz disparitetne mape izračunate SGBM metodom	18
2.13.	Prikaz estimacije pravca RANSAC algoritmom	18
3.1.	Lijeva i desna slika ispitnog skupa KITTI	23
3.2.	Lijeva i desna slika ispitnog skupa EISATS	23
4.1.	logo	24
4.2.	PCL logo	25
4.3.	Prikaz disparitetne mape	26
4.4.	Prikaz oblaka točaka	27
4.5.	Skup točaka za estimaciju ravnine ceste	28
4.6.	Sferni koordinatni sustav	29
4.7.	Blok u oblaku točaka	29
4.8.	Objekt i sumovi	30
4.9.	Izvorna slika s označenim objektima	31
5.1.	Izvorna slika s označenim objektima	33
5.2.	Scene vangradske vožnje	34

5.3.	Scene gradske vožnje	35
5.4.	1. primjer pogrešne detekcije objekata	36
5.5.	2. primjer pogrešne detekcije objekata	36
5.6.	3. primjer pogrešne detekcije objekata	37

POPIS TABLICA

5.1. Usporedba performansi BM i SGBM	33
--	----

1. Uvod

U današnje vrijeme automatsko prepoznavanje objekata na cesti dobiva sve više na značenju. Sustavi za prepoznavanje objekata se mogu koristiti za povećavanje sigurnosti cesta, bolje održavanje cesta, pronalaženje oštećenih znakova ili mjesta na kojima su prometni znakovi odstranjeni. Također, ta rješenja se mogu koristiti za povećanje sigurnosti vozila. Vozač u automobilu može previdjeti prometni znak ili čak cijelo vozilo zbog umora ili nekog drugog razloga. Stoga je cilj ugraditi u vozila sustave koji bi mogli detektirati i raspoznati objekte u prihvatljivom vremenskom intervalu. Takvi sustavi bi se mogli koristiti kao potpora vozaču pružajući mu informacije kojih možda nije bio svjestan. Rezultat korištenja takvih sustava bio bi povećanje sigurnosti u prometu.

Cilj ovog rada je napraviti sustav za detekciju objekata u stvarnom vremenu u gusto stereoskopskoj rekonstrukciji s naglaskom na scene snimane u prometu. Budući da je cilj ostvariti sustav u stvarnom vremenu, prilikom formiranja oblaka točaka težište je stavljeno na lokalne metode podudaranja koje su brže od globalnih metoda što naravno donosi i prepreke poput određivanja odgovarajućeg disariteta na rubovima objekata i točkama površine jednolike boje. Kao korak prema detekciji objekata, oblak točaka je podjeljen u blokove. Susjedni blokovi koji sadrže više od zadanog praga točaka se zatim grupiraju u objekte.

U 2. poglavlju dan je opis algoritama i matematičkih metoda korištenih u ovom radu. U 3. poglavlju predstavljeni su korišteni ispitni skupovi. U 4. poglavlju predstavljena je programska izvedba i korištene vanjske biblioteke dok su u 5. poglavlju opisani eksperimentalni rezultati.

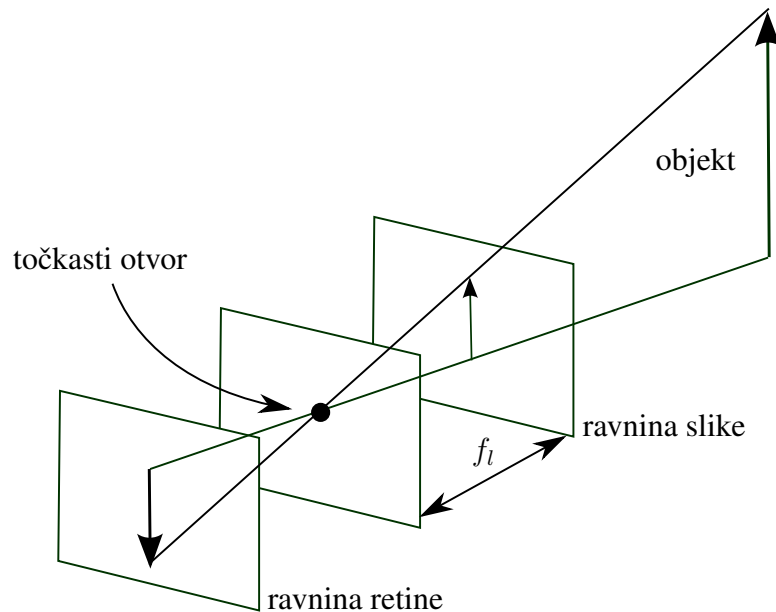
2. Algoritmi i matematičke metode

2.1. Stereo vid

Stereo vid je instrument pomoću kojeg se dobivaju 3D informacije o sceni promatrane pomoću dvije ili više slika snimljenih u istom trenutku s različitih pozicija. Stereo rekonstrukcija se temelji na sposobnosti pronalaženja projekcija istog objekta scene na svakoj slici. Informacija o dubini objekta u prostoru dobiva se pomoću dispariteta (razlike u projekcijama objekta na slikama) i relativne pozicije kamera. Dakle, stereo rekonstrukcija se sastoji od dva problema. Prvi je izračunavanje dispariteta, što se svodi na nalaženje odgovarajućih parova piksela u slikama (problem podudaranja). Drugi problem je napraviti inverz geometrijske projekcije, tj. rekonstrukciju 3D prostora pomoću poznavanje dispariteta i relativne pozicije kamera. Radom na projektivnoj geometriji Faugeras (1993) je postavio osnovu rješavanja rekonstrukcije 3D prostora. Što se tiče problema podudaranja, ne postoji metoda dovoljno pouzdana, robusna i efektivna koja bi omogućila jednostavno korištenje stereo vida kao senzora za mjerenje dubine.

Preliminarna faza kalibracije je potrebna kako bi se procijenili različiti parametri stereo opreme: parametri projekcijskog modela svake kamere i prostorni odnosi između kamera. To znanje omogućava izračunavanje 3D koordinata točke iz njezinih projekcija na slikama pomoću triangulacije.

Problem podudaranja je tema koja se proučava već više od pola stoljeća Julesz (1962). Cilj je pronaći značajke u lijevoj i desnoj slici koje su projekcije istog entiteta u 3D sceni. Općenito, problem podudaranja možemo gledati kao minimizacijski problem. Lokalni pristupi pokušavaju minimizirati mnogo odvojenih funkcija energije, pretpostavljajući da su entiteti koje predstavljaju funkcije energije međusobno nezavisni. Globalni pristup pokušava minimizirati jedinstvenu funkciju energije uzimajući u obzir lokalnu cijenu podudaranja i također cijenu sukladnosti koja izražava koliko je dos-



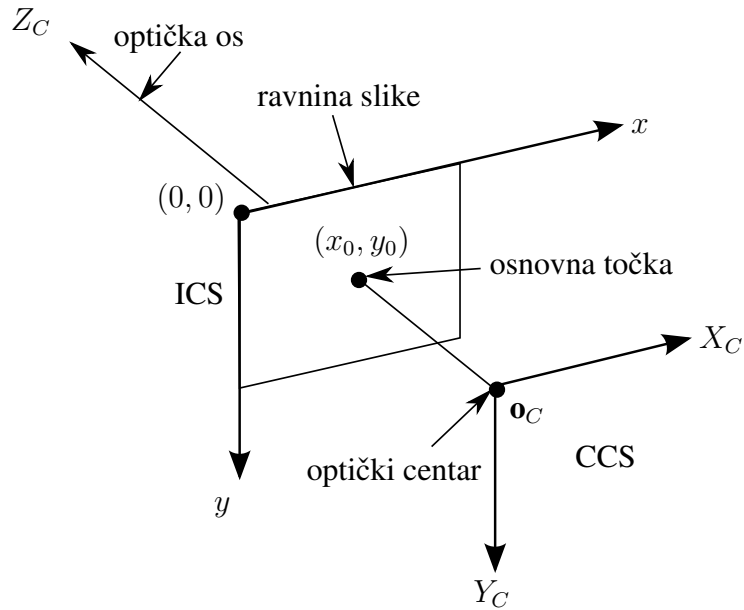
Slika 2.1: Model kamere s točkastim otvorom

Ijedno sparivanje izračunato na cijeloj slici. Detaljna taksonomija algoritama stereo podudaranja predložena je u Scharstein i Szeliski (2002). Autori klasificiraju metode s obzirom na četiri kriterija: (i) cijena lokalnog podudaranja, (ii) agregacija područja za vrijeme računanja lokalne cijene, (iii) optimizacijska metoda, (iv) metoda poboljšavanja rezultata.

2.1.1. Model kamere

Kamera s točkastim otvorom (engl. *pinhole camera*) je jednostavni geometrijski model kojim se prikazuje proces formiranja slike u kameri. Kao što je prikazano na slici 2.1, otvor na kameri (promjer leće) je predstavljena kao mala rupica kroz koju svjetlost prolazi. Intenziteti svjetlosti objekta se projiciraju na ravninu slike pomoću perspektivne projekcije. Kako bi se odredio postupak kojim se trodimenzionalni objekti iz svijeta geometrijski preslikavaju na dvodimenzionalnu sliku kamere potrebno je definirati tri različita koordinatna sustava: *koordinatni sustav svijeta* (engl. *the world coordinate system* - *WCS*), *koordinatni sustav kamere* (engl. *the camera coordinate system* - *CCS*) i *koordinatni sustav slike* (engl. *the image coordinate system* - *ICS*).

WCS je fiksni, trodimenzionalni referentni sustav za prikaz trodimenzionalnih objekata i scena u prostoru. Definiran je ortogonalnim osima (X_W, Y_W, Z_W) i ishodištem $\mathbf{o}_W$. CCS je također trodimenzionalni koordinatni sustav koji odgovara po-

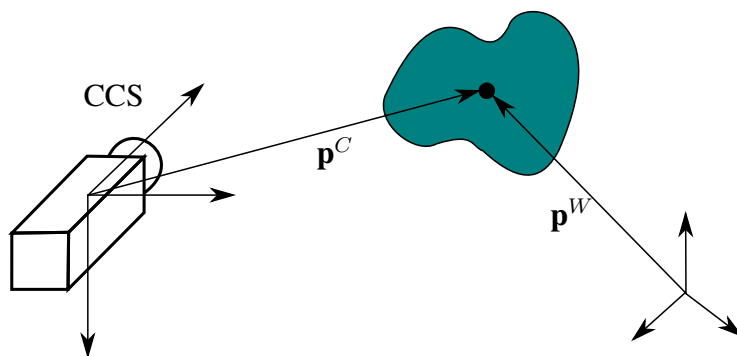


Slika 2.2: Shema kamere i koordinatnog sustava slike



Slika 2.3: Odnos koordinatnih sustava prostora, kamera i slika

ložaju i orijentaciji kamere. Kao što je prikazano na slici 2.2, CCS je definiran osima (X_C, Y_C, Z_C) i ishodištem $\mathbf{o}_C$ koje odgovara *optičkom centru kamere* (točkastom otvoru). Z_C os se naziva *optička os* i ortogonalna je na ravninu slike. ICS je dvodimenzionalni sustav u kojem su smješteni pikseli slike kamere. Definiran je (x, y) osima i ishodištem u gornjem lijevom kutu slike. Sjecište optičke osi kamere s ravninom slike naziva se *osnovna točka* (x_o, y_o) (engl. *principal point*), čije se koordinate označavaju u ICS.



Slika 2.4: Odnos koordinatnih sustava prostora, kamera i slika

Gore navedeni koordinatni sustavi su povezani međusobno pomoću dva seta parametara, intrinzičnih i ekstrinzičnih, kao što je prikazano na slici 2.3. Ekstrinzični parametri sastoje se od 3×3 ortogonalne rotacijske matrice A_W i translacijskog vektora $\mathbf{b}_W$, koji opisuju poziciju i orijentaciju CCS u odnosu na WCS. Slika 2.4 prikazuje odnos između ta dva koordinatna sustava.

Neka je $\mathbf{p}^W = [X_W \ Y_W \ Z_W]^T$ trodimenzionalna točka u WCS, tada njezine koordinate u CCS iznose

$$\mathbf{p}^C = [X_C \ Y_C \ Z_C]^T = A_W \mathbf{p}^W + \mathbf{b}_W.$$

Intrinzični parametri, određeni optičkim i digitalnim svojstvima senzora kamere, opisuju perspektivnu projekciju tridimenzionalne točke na dvodimenzionalnu sliku. Neka je $\mathbf{m} = [x \ y]^T$ koordinata piksela u ICS. Njezine koordinate u odnosu na koordinate u CCS se izračunavaju pomoću formula

$$\begin{aligned} x &= \frac{f_l X_C}{a_x Z_C} + x_o, \\ y &= \frac{f_l Y_C}{a_y Z_C} + y_o, \end{aligned} \tag{2.1}$$

gdje je

- f_l je fokalna duljina leća kamere,
- a_x i a_y su širina i visina slike kamere,
- (x_o, y_o) su koordinate osnovne točke kamere.

Alternativno, ako definiramo projektivnu matricu

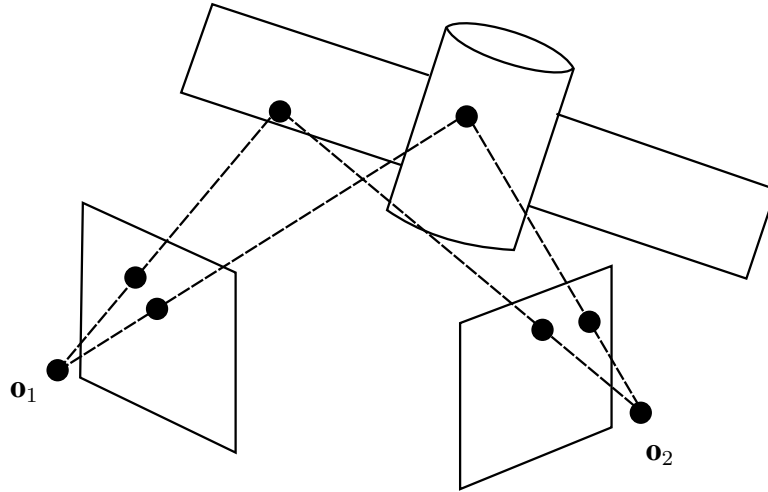
$$P = \begin{bmatrix} \frac{f_l}{a_x} & 0 & X_o \\ 0 & \frac{f_l}{a_y} & y_o \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

i

$$\begin{bmatrix} x^* \\ y^* \\ w^* \end{bmatrix} = P \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix},$$

tada je

$$x = \frac{x^*}{w^*}, \quad y = \frac{y^*}{w^*}.$$



Slika 2.5: 3D rekonstrukcija iz 2D slika

Kao što se vidi iz (2.1), kamera reducira tridimenzionalnu reprezentaciju svijeta na dvije dimenzije, pri čemu se gubi informacija o dubini. Nadalje, perspektivna projekcija je nelinearan proces, što komplicira razvoj algoritama pomoću kojih bi se izračunala izgubljena dimenzija.

2.1.2. 3D rekonstrukcija

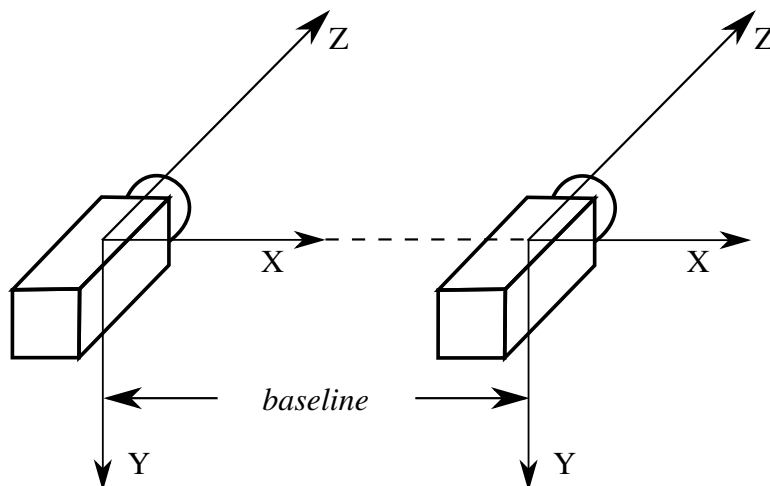
3D rekonstrukcija je postupak izračunavanja izgubljenih informacija o dubini iz slika intenziteta. Uobičajeni pristup 3D rekonstrukcije koristi dvije ili više slika. Bazira se na činjenici da fizička točka u prostoru se projicira na različite lokacije na slikama slikanim s različitih gledišta, kao što je prikazano na slici 2.4. Razlika u projiciranim lokacijama se koristi za izračun informacije o dubini.

Formalnije, neka je kruto tijelo predstavljeno skupom od N 3D točaka $\{\mathbf{p}_i\}$. Svaka točka $\mathbf{p}_i$ je projicirana na sliku $\mathbf{I}_s$, koja je snimljena s gledišta s . Te točke označimo $\mathbf{m}_i^s$ u ICS.

$$(\mathbf{m})_i^s = \begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{h}(s, \mathbf{p}_i). \quad (2.3)$$

$\mathbf{h}$ je pojednostavljenje perspektivnih jednadžbi (2.1) za određenu kameru.

Za dan skup slika snimljenih s različitih gledišta s moguće je rekonstruirati točke $\{\mathbf{p}_i\}$. Detaljnije kako je moguće izvesti objašnjeno je u idućim poglavljima.



Slika 2.6: Tipiči sustav stereo kamera

Postoje dva podproblema vezana uz 3D rekonstrukciju iz dviju ili više slika:

1. korespodencija značajki
2. određivanje strukture

Prvi problem je najlakše objasniti na primjeru. Na primjer, fizička 3D točka je projicirana na sliku A kao točka 1 i na sliku B kao točka 2. Takve točke su korespondentne. Stoga, korespodencija značajki ili problem sparivanja značajki je pronalaženje lokacije točke 2 na slici B ako je poznata lokacija točke 1 na slici A. Ljudski vid je odličan u rješavanju ovog problema, ali automatizacija pomoću računala je poprilično teška. U suštini zahtijeva potragu na cijeloj slici B. Primjenom odgovarajućih ograničenja moguće je suziti pretragu, dok bez ograničenja problem ima previše mogućih rješenja.

Drugi problem, određivanje strukture, je računanje skupa točaka $\{p_i\}$ nakon što je problem sparivanja značajki riješen. Težina ovog podproblema ovisi o količini *a priori* dostupnih informacija. Ako su intrinzični i ekstrinzični parametri kamera poznati, tada točna rekonstrukcija je moguća. Međutim, točnost rekonstruirane strukture je osjetljiva na točnost tih parametara. Štoviše, svaka greška nastala prilikom računanja korespodencija među slikama također utječe na točnost rekonstrukcije.

2.1.3. Određivanje strukture iz stereo slika

Korištenje vizualnih podražaja oba oka za stvaranje percepcije dubine dobro je poznat fenomen ljudskog vida. Računalni vid primjenjuje isti princip za računanje dubine

iz slika dobivenih s različitih gledišta. Tipični sustav stereo kamera prikazan je na slici 2.6. Montirane su dvije kamere s koplanarnim, paralelnim optičkim osima (Z-osima). Razmak između optičkih centara lijeve i desne kamere (engl. *baseline*) nastaje translacijom optičkog centra kamere po zajedničkoj X osi.

Lijeva i desna kamera u stereo sustavu snimaju par slika, $\{I_L, I_R\}$, istovremeno ili odvojeno pod uvjetom da se ne događaju promjene u sceni između snimanja dviju slika. U stereo vidu, razlika u poziciji projicirane točke na lijevu i desnu sliku se naziva *disparitet* (engl. *disparity*), a skup disparitetnih vrijednosti za cijelu sliku se naziva *disparitetna mapa* (engl. *disparity map*).

Stereo geometrija

Jedna od prednosti rekonstrukcije 3D scene iz stereo slika je činjenica da su geometrijski odnosi između slika I_L i I_R poznati zbog fiksne konfiguracije većine stereo sustava. Ako su i intrinzični i ekstrinzični parametri kamere određeni *kalibracijom kamere*, onda se problem određivanja strukture može riješiti koristeći jednostavnu proceduru zvanu *triangulacija*. Slova L i R će se koristiti konzistentno kroz ovaj rad kao indexi koji označavaju lijevu, odnosno desnu kameru.

Neka su $\mathbf{p}^L$ i $\mathbf{p}^R$ koordinate kamere iste točke u prostoru $\mathbf{p}^W$, $(A_{WL}, \mathbf{b}_{WL})$ i $(A_{WR}, \mathbf{b}_{WR})$ ekstrinzični parametri lijeve, odnosno desne kamere takvi da vrijedi

$$\begin{aligned}\mathbf{p}^L &= A_{WL}\mathbf{p}^W + \mathbf{b}_{WL} \\ \mathbf{p}^R &= A_{WR}\mathbf{p}^W + \mathbf{b}_{WR}\end{aligned}$$

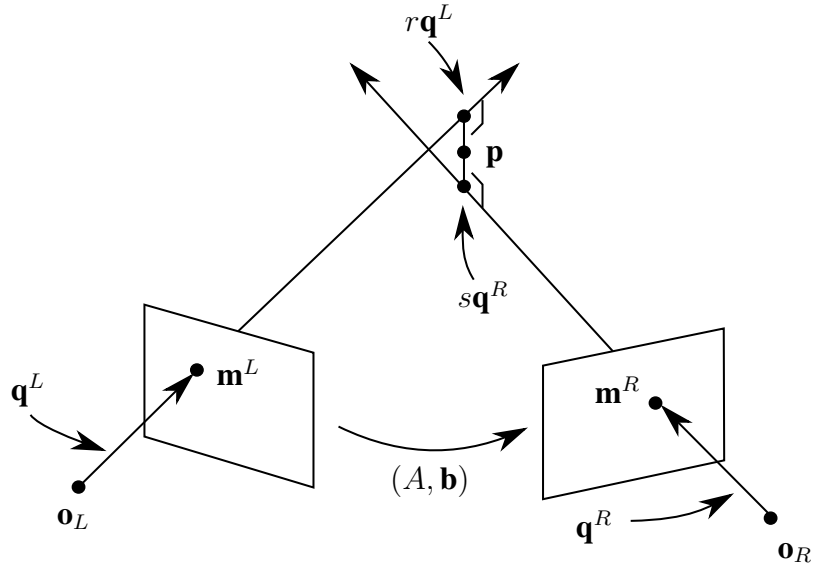
Tada $\mathbf{p}^R$ se može iskazati preko $\mathbf{p}^L$:

$$\mathbf{p}^R = A\mathbf{p}^L + \mathbf{b} \tag{2.4}$$

gdje je

$$A = A_{WR}A_{WL}^{-1}, B = -A_{WR}A_{WL}^{-1}\mathbf{b}_{WL} + \mathbf{b}_{WR}. \tag{2.5}$$

Koristeći (2.4) koordinate kamere lijevog CCS mogu se transformirati u koordinate kamere desnog CCS i obrnuto.



Slika 2.7: 3D rekonstrukcija triangulacijom

Rekonstrukcija pomoću triangulacije

Za rekonstrukciju iz stereo slika, problem pronalaženja odgovarajućih značajki jednak je traženju skupa odgovarajućih projekcija $\{\mathbf{m}^L, \mathbf{m}^R\}$. Pretpostavimo da je taj problem već riješen, tj. imamo $\{\hat{\mathbf{m}}^L, \hat{\mathbf{m}}^R\}$, procjenjene projekcije točke $\mathbf{p}$.

Neka su $\mathbf{q}^L$ i $\mathbf{q}^R$ trodimenzionalni vektori koji označavaju smjer $\mathbf{m}^L$ i $\mathbf{m}^R$ u odnosu na optičke centre kamera. Kao što je prikazano na slici 2.7, svrha triangulacije je pronaći sjecište dvaju vektora ekstrapoliranih iz $\mathbf{q}^L$ i $\mathbf{q}^R$.

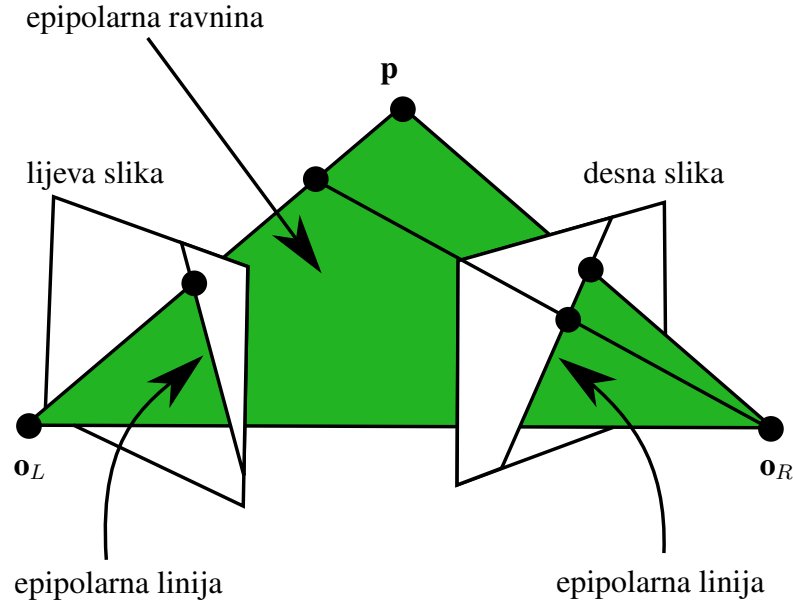
Neka su P_L i P_R projektivne matrice definirane u (2.2). Primjenom inverza projekcije na homogene koordinate od $\hat{\mathbf{m}}_i^L$ i $\hat{\mathbf{m}}_i^R$ dobivamo:

$$\mathbf{q}^L = P_L^{-1} \begin{bmatrix} \hat{\mathbf{m}}^L \\ 1 \end{bmatrix},$$

$$\mathbf{q}^R = P_R^{-1} \begin{bmatrix} \hat{\mathbf{m}}^R \\ 1 \end{bmatrix}.$$

Zbog grešaka prilikom kalibracije kamera i izlučivanja značajki, moguće je da se ekstrapolirani vektori ne sijeku točno. Jednostavna metoda kojom se rješava ovaj problem jest procijeniti $\mathbf{p}$ kao središnju točku segmenta ortogonalnog na $\mathbf{q}^L$ i $\mathbf{q}^R$.

Neka su r, s, t skalarne varijable. Uz (2.4), odnosi prikazani na slici 2.7 mogu se prikazati u lijevom CCS-u:



Slika 2.8: Epipolarna ravnina

$$r\mathbf{q}^L - A^T (s\mathbf{q}^R - \mathbf{b}) = t [\mathbf{q}^L \times A^T (\mathbf{q}^R - \mathbf{b})], \quad (2.6)$$

pojednostavljanjem (2.6) dobiva se

$$\begin{aligned} r\mathbf{q}^L - sA^T\mathbf{q}^R - t [\mathbf{q}^L \times A^T (\mathbf{q}^R - \mathbf{b})] &= -A^T\mathbf{b}, \\ \begin{bmatrix} \mathbf{q}^L & -A^T\mathbf{q}^R & \mathbf{q}^L \times A^T (\mathbf{q}^R - \mathbf{b}) \end{bmatrix} \begin{bmatrix} r \\ s \\ t \end{bmatrix} &= -A^T\mathbf{b}. \end{aligned} \quad (2.7)$$

(r, s) je određen rješavanjem linearnog sustava zadanog u (2.7). $\hat{\mathbf{p}}$, procjena za $\mathbf{p}$, je srednja točka od $r\mathbf{q}^L$ i $A^T(s\mathbf{q}^R - \mathbf{b})$:

$$\hat{\mathbf{p}} = \frac{r\mathbf{q}^L + A^T(s\mathbf{q}^R - \mathbf{b})}{2}$$

U ovoj diskusiji pretpostavljeno je da odgovarajuće projekcije točke $\mathbf{p}$ su dostupne. Međutim, kao što je spomenuto ranije, podudaranje značajki je zahtjevan problem koji treba riješiti prije rekonstrukcije. O njemu će biti raspravljeno u nastavku.

Epipolarno ograničenje

Nakon provedbe izdvajanja značajki dobivena su dva skupa značajki, jedan za svaku sliku I_L i I_R ; problem sparivanja značajki je pronaći odgovarajuće značajke na desnoj slici za detektiranu značajku u lijevoj slici ili obrnuto. Teoretski svaka značajka u desnoj slici je potencijalni odgovarajući par za svaku značajku u lijevoj slici, što znači da podudaranje značajki je dvodimenzionalan problem pretraživanja. Kako bi se moglo efikasnije i brže riješiti taj problem, treba ga ograničiti.

Na slici 2.8 vidljivo je da točka u prostoru i optički centri lijeve i desne kamere čine ravninu zvanu *epipolarna ravnina*. Linije kojom epipolarna ravnina presjeca ravnine dviju slika zovu se *epipolarne linije*. *Epipolarno ograničenje* označava da projekcije prostorne točke moraju ležati na epipolarnim linijama što reducira problem pretraživanja na jednu dimenziju.

Neka je $EP[\mathbf{m}^L]$ normala epipolarne linije značajke $\mathbf{m}^L$ lijeve slike

$$EP[\mathbf{m}^L] = \begin{bmatrix} u \\ v \\ w \end{bmatrix}.$$

Epipolarno ograničenje implicira da odgovarajuća točka od $\mathbf{m}^L$ u desnoj slici leži na liniji određenoj s $EP[\mathbf{m}^L]$. Matematički zapisano

$$\begin{bmatrix} (\mathbf{m}^R)^T & 1 \end{bmatrix} EP[\mathbf{m}^L] = 0, \quad (2.8)$$

Pozicija epipolarne linije na lijevoj i desnoj slici ovisi o geometriji stereo sustava i može biti određena koristeći intrinzične i ekstrinzične parametre kamera.

Koristeći P_L, P_R, A i $\mathbf{b}$ prije definirane, neka B bude antisimetrična matrica za koju vrijedi $B\mathbf{x} = \mathbf{b} \times \mathbf{x}$ za sve 3D vektore $\mathbf{x}$, tj.

$$B = \begin{bmatrix} 0 & -b_z & b_y \\ b_z & 0 & -b_x \\ -b_y & b_x & 0 \end{bmatrix}$$

Matrica 3×3 , poznata kao *fundamentalna matrica*, definirana je kao

$$F = P_R^{-T} B A P_L^{-1}$$

i zadovoljava relaciju

$$\begin{bmatrix} (\mathbf{m}^R)^T & 1 \end{bmatrix} F \begin{bmatrix} \mathbf{m}^L \\ 1 \end{bmatrix} = 0. \quad (2.9)$$

Tada uspoređujući epipolarna ograničenja (2.8) i (2.9) dobivamo

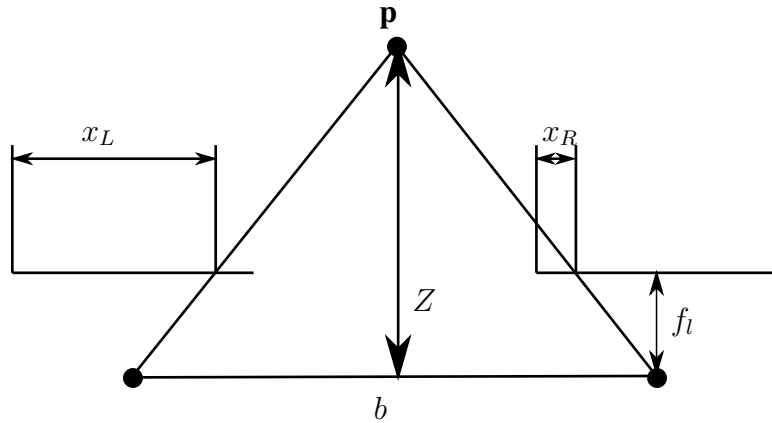
$$EP[\mathbf{m}^L] = F \begin{bmatrix} \mathbf{m}^L \\ 1 \end{bmatrix}.$$

U slučaju da je rijetka gustoća značajki na slici, epipolarno ograničenje može biti dovoljno za jedinstveno rješavanje problema sparivanja značajki uz pretpostavku da izvlačenje značajki pouzdano detektira odgovarajuće točke u obje slike. Međutim, kao što prikazuje slika 2.8, ako postoji više od jedne značajke u desnoj slici koja leži u blizini epipolarne linije, dubina točke koju promatramo ne može se jedinstveno odrediti. Stoga epipolarno ograničenje samo za sebe nije garancija jedinstvenog rješenja problema pronalaženja odgovarajućih značajki.

Čak i nakon uvođenja epipolarnog ograničenja, proizvoljna orijentacija tih linija nije prikladna za jednostavnu i brzu izvedbu algoritma. Kako bi se riješio taj problem, ulazne slike su *rektificirane* tako da su epipolarne linije transformirane u odgovarajuće horizontalne linije. Rektifikacije može biti izvedena prvo rotacijom kamera tako da im optičke osi budu okomite na liniju koja spaja centre kamera (odmak), nadalje y-os kamera je rotirana da bude okomita na odmak i ako je potrebno slike se skaliraju kako bi se kompenzirala razlika u fokalnim duljinama. U praksi, vrlo je teško namjestiti da optičke osi dviju kamera budu paralelne međusobno i okomite na odmak. Povrh toga, kamere su često nagnute kako bi adekvatno snimile regiju interesa. Stoga rektifikacija se češće izvodi softverski nego hardverski.

$$\begin{aligned} \frac{b}{Z} &= \frac{b + x_R - x_L}{Z - f_l}; & Z &= \frac{f_l \cdot b}{x_L - x_R} = \frac{f_l \cdot b}{d} \\ \frac{X}{Z} &= \frac{x_L}{f_l}; & X &= \frac{x_L}{f_l} \cdot \frac{f_l \cdot b}{d} = \frac{x_L \cdot b}{d} \\ \frac{Y}{Z} &= \frac{y_L}{f_l}; & Y &= \frac{y_L}{f_l} \cdot \frac{f_l \cdot b}{d} = \frac{y_L \cdot b}{d} \end{aligned} \quad (2.10)$$

Slika 2.9 prikazuje ilustraciju triangulacije kod rektificiranog sustava. Iz sličnosti trokuta dobivamo dubinu Z točke p u koordinatnom sustavu lijeve kamere. Na sličan način pomoću sličnosti drugih trokuta dobivamo i koordinate X i Y . U izrazu (2.10) uvedena je oznaka $d = x_L - x_R$ koja predstavlja disparitet.



Slika 2.9: Rektificirana triangulacija

2.1.4. Mjere sličnosti

Bez obzira na vrstu stereo algoritma koji se koristi, sredstvo za određivanje sličnosti između piksela u različitim slikama je ključ rješavanja problema jednoznačnosti potencijalnih podudaranja tj. pronalaženja korespondentnih točaka. Kako bi se za dani piksel $\mathbf{x} = [xy]^T$ na lijevoj slici I_L našao odgovarajući piksel u desnoj slici I_R koristiti se fiksni prozor uzorkovan na pozicijama oko piksela $\{\mathbf{x} = (x_i, y_i)\}$ koji mjeri stupanj sličnosti I_L s različitim disparitetima. Osnovna mjera je *suma kvadrata razlike* (engl. *sum of squared differences - SSD*) definirana formulom

$$C_{SSD}(\mathbf{x}, \mathbf{d}) = \sum_i [I_R(\mathbf{x}_i + \mathbf{d}) - I_L(\mathbf{x}_i)]^2.$$

Ova mjera sličnosti implicitno pretpostavlja da intenziteti piksela oko korespondentnih točaka u lijevoj i desnoj slici ostaju konstantni te za točan disparitet ima minimalnu vrijednost. Mjera sličnosti može biti robusnija na prisutnosti šuma u slici, ako raste raste sporije od kvadratne SSD mjere. Uobičajen primjer takve robusne mjere je *suma apsolutnih razlika* (engl. *sum of absolute differences - SAD*) definirana formulom

$$C_{SAD}(\mathbf{x}, \mathbf{d}) = \sum_i |I_R(\mathbf{x}_i + \mathbf{d}) - I_L(\mathbf{x}_i)| \quad (2.11)$$

U ovom slučaju, funkcija cilja linearno raste s pogreškom između prozora u slikama čime smanjuje utjecaj neusklađenih piksela. Također robusna mjera sličnosti je *normalizirana kros-korelacija* (engl. *normalized cross corelation - NCC*)

$$C_{NCC}(\mathbf{x}, \mathbf{d}) = \frac{\sum_i I_L(\mathbf{x}_i) * I_R(\mathbf{x}_i + \mathbf{d})}{\sqrt{\sum_i I_L^2(\mathbf{x}_i) * \sum_i I_R^2(\mathbf{x}_i + \mathbf{d})}}$$

2.1.5. Lokalne metode podudaranja

Lokalne metode podudaranja ocjenjuju u kojoj mjeri točka jedne slike odgovara točki druge slike zbrajajući cijenu podudaranja prozora piksela oko tih točki. Kako bi se postigla glatka i detaljna disparitetna mapa odabir odgovarajućeg prozora je kritičan. Optimalan prozor treba biti dovoljno velik da sadrži dovoljno različitih intenziteta za pouzdano podudaranje, pogotovo u zonama niske kvalitete teksture. S druge strane, treba biti dovoljno malen da minimizira varijaciju dispariteta na pregibima površina ili dubinskim diskontinuitetima. Koristeći preveliki prozor može dovesti do neželjenog zaglađivanja i zadebljavanja ili stanjivanja rubova kao posljedica razlike između teksture visoke rezolucije i niske rezolucije na granicama objekata. Dok u najjednostavnijem slučaju prozor pretraživanja može biti fiksni kvadratni prozor, postoje tehnike koje balansiraju između prednosti i mana različitih prozora. Generalno, te metode dinamički se adaptiraju ovisno o okolnim pikselima, te prozor na primjer može imati adaptivnu veličinu, oblik, itd.

Nakon računanja i zbrajanja funkcije cilja, disparitet dane točke se jednostavno računa koristeći optimizaciju *pobjednik uzima sve* (engl. *winner takes all*), tj. uzme se disparitetna vrijednost povezana s minimalnom cijenom podudaranja. Ponavljajući taj proces za svaku točku slike stvara se disparitetna mapa.

$$D(\mathbf{x}) = \operatorname{argmin} C_{SAD}(\mathbf{x}, \mathbf{d})$$

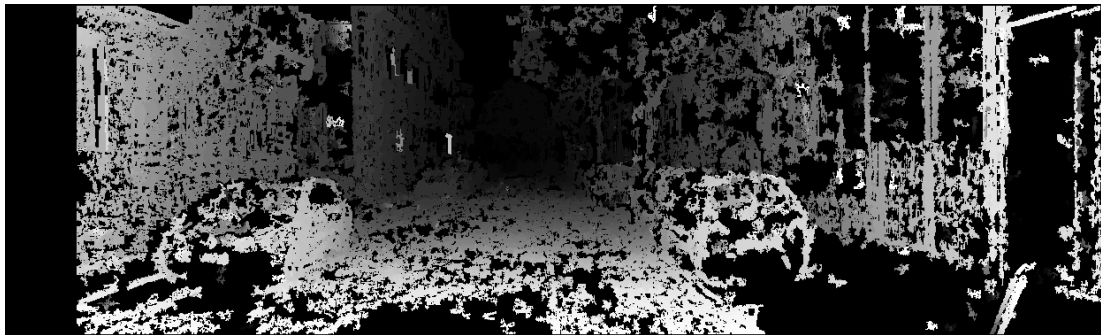
Ipak, u nekim primjenama tako stvorena disparitetna mapa nije dovoljno točna, tj. stvara nemoguće ili šumovite objekte. Stoga mnogi lokalni stereo algoritmi koriste dodatne korake zaglađivanja. Postoji nekoliko tehnika koje se mogu koristiti za procjenu prikladnijih dispariteta uz malu dodatnu cijenu izračuna. Na primjer, napraviti disparitetnu mapu između lijeve i desne slike, i obrnuto, između desne i lijeve i pomoću njih detektirati skrivena područja, ili na primjer primijeniti median filter kojim se eliminiraju šumovi u disparitetu.

Lokalne metode podudaranja su generalno brzi algoritmi zbog čega se mogu koristiti u aplikacijama za rad u stvarnom vremenu. Međutim, izložene su mnogim problemima,

posebno okluziji i varijaciji intenziteta između slika koji mogu proizvesti mnogo lažnih podudaranja. Osim toga, zbog nedostatka bilo kakvih ograničenja među podudaranjima, susjedni pikseli mogu imati vrlo različite disparitete što može biti značajno u scenama koje sadrže vertikalne linije. Globalne metode podudaranja pokušavaju prevladati ove probleme.

Block Matching (BM)

U ovom radu, kao izbor lokalne metode podudaranja uzet je StereoBM algoritam iz OpenCV-a. Kao mjeru sličnosti odabranih blokova piksela koristi sumu apsolutnih vrijednosti (2.11).



Slika 2.10: Prikaz disparitetne mape izračunate BM metodom

Na slici 2.10 je vidljivo kako za mnogo piksela nije moguće s dovoljnom sigurnošću odrediti pripadajući disparitet te su na slici označeni crnom bojom. Taj problem nastaje zbog sličnosti intenziteta okolnih piksela te za različite disparitete su izračunate jednake ocjene sličnosti.

2.1.6. Globalne metode podudaranja

Globalni pristupi pokušavaju definirati globalni model promatrane scene i minimizirati globalnu funkciju cilja. U globalnoj metodi, podudaranje piksela lijeve slike i piksela desne slike ne ovisi samo o njihovim susjedima, nego ovisi i o podudaranju njihovih susjeda. Utjecaj je modeliran reguliranjem ograničenja nad podudarajućim skupom. U grubo, mogu se podijeliti na metode kojima je cilj minimizirati globalnu energiju i metode bazirane na dinamičkom programiranju koje problem dvodimenzionalnog podudaranja smanjuju na jednodimenzionalni problem. Globalno reguliranje ima cilj

smanjiti probleme prilikom podudaranja u slučajevima okluzije, loše lokalne teksture ili promjene osvjetljenja. Ova poboljšanja imaju svoju cijenu, što znači da imaju povećanu algoritamsku složenost, i kao posljedicu, duže vrijeme izvođenja. Funkcija cilja se općenito definira

$$E(D) = \sum_{\mathbf{p}} \left(C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P[|D_{\mathbf{p}} - D_{\mathbf{q}}| \geq 1] \right) \quad (2.12)$$

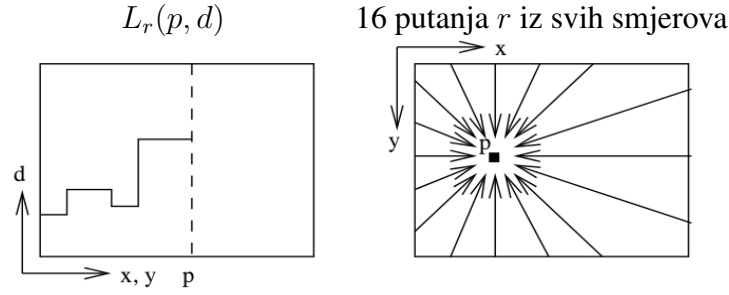
Prvi član funkcije sumira cijene podudaranja svih piksela slike, dok drugi član dodaje penale za sve piksele čiji susjedi imaju različiti disparitet od promatranog piksela. Ovakvom formulom diskontinuiteti su dozvoljeni ako je podudaranje piksela veće od penala, tj. ako tekstura označava diskontinuitet. Vrijedi primjetiti da drugi član indirektno povezuje sve piksele međusobno što čini funkciju globalnom. Nažalost traženje disparitetne slike D koja minimizira ovu funkciju je problem NP složenosti. Algoritam Semi-Global Block Matching uspješno kombinira koncepte globalnih i lokalnih metoda podudaranja i time postiže polinomijalnu složenost uz bolje rezultate od lokalnih metoda.

Semi-Global Block Matching (SGBM)

Semi global block matching je OpenCV implementacija građena na ideji predstavljenoj u Hirschmüller (2005). Energija piksela se izračunava kao srednja vrijednost bloka okolnih piksela pa je u naziv još dodano i *block*. Implementacija algoritma pomoću kojeg se izračunava cijena podudaranja blokova piksela opisana je u Birchfield i Tomasi (1998). SGBM koristi lagano izmijenjenu funkciju cilja u odnosu na definiranu u (2.12). Dodani su različiti penali P_1 i P_2 za male i velike razlike u disparitetima među susjednim pikselima.

$$E(D) = \sum_{\mathbf{p}} \left(C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1] \right) \quad (2.13)$$

Penal za manje promjene u disparitetu P_1 je konstanta dok penal za veće promjene P_2 se mijenja po formuli (2.14) kako bi se očuvali diskontinuiteti. $I_{L\mathbf{p}}$ i $I_{L\mathbf{q}}$ predstavljaju susjedne piksele u lijevoj slici, a P'_2 konstantu koja osigurava da vrijedi $P_1 \leq P_2$.



Slika 2.11: minimizacija energije duž putanja r

$$\frac{P'_2}{|I_{Lp} - I_{Lq}|} \quad (2.14)$$

Kao što je rečeno prije, minimizacija energije u 2D je NP-kompletna problem za razliku od minimizacije duž jednog reda slike, tj. 1D se može izračunati u polinomijalnom vremenu koristeći dinamičko programiranje. Da bi se pokrile bitnije informacije iz svih smjerova slike, uzete su u obzir 1D linije iz 'svih' smjerova duž kojih se traži disparitet kao što je prikazano na slici 2.11. Hirschmüller preporuča korištenje 8-16 linija iz svih smjerova, dok standardna OpenCV implementacija koristi 5 ili 8 smjerova zbog velikog troška memorije.

Cijena L_r duž linije u smjeru r za piksel p i disparitet d se rekurzivno računa kao

$$L_r(\mathbf{p}, d) = C(\mathbf{p}, d) + \min \left[L_r(\mathbf{p} - \mathbf{r}, d), \right. \\ L_r(\mathbf{p} - \mathbf{r}, d - 1) + P_1, \\ L_r(\mathbf{p} - \mathbf{r}, d + 1) + P_1, \\ \left. \min_i L_r(\mathbf{p} - \mathbf{r}, i) + P_2 \right] - \min_k L_r(\mathbf{p} - \mathbf{r}, k)$$

Skupljena cijena $S(\mathbf{p}, d)$ za piksel $\mathbf{p}$ i disparitet d računa se sumirajući cijene svih $L_r(\mathbf{p}, d)$.

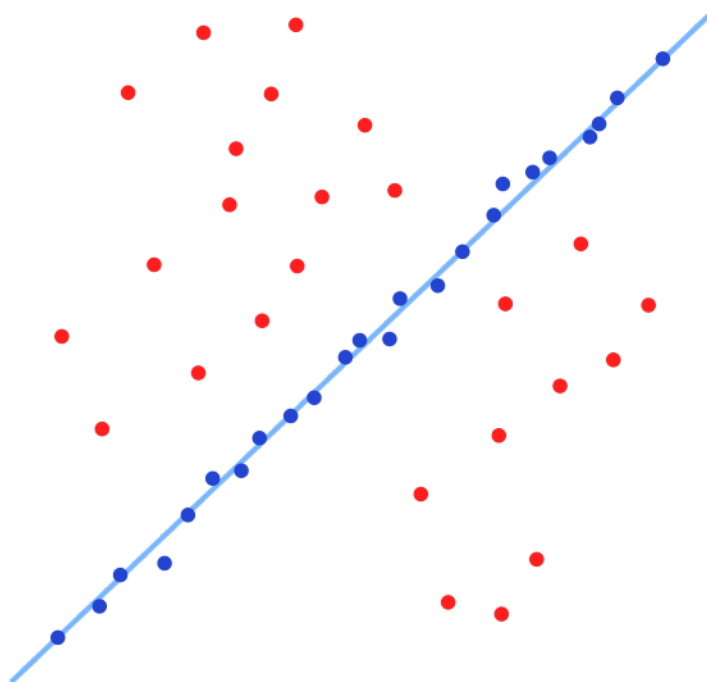
$$L_r(\mathbf{p}, d) = \sum_r L_r(\mathbf{p}, d)$$

Disparitetna mapa D se računa birajući disparitete s minimalnom skupljenom cijenom putanja za svaki piksel.



Slika 2.12: Prikaz disparitetne mape izračunate SGBM metodom

2.2. Konsenzus slučajnih uzoraka



Slika 2.13: Prikaz estimacije pravca RANSAC algoritmom

Konsenzus slučajnih uzoraka (RANSAC) je iterativna metoda za procjenu parametara matematičkog modela iz skupa podataka. Osnovna pretpostavka je da se podaci sastoje od populacijskih i vanpopulacijskih podataka. Populacijski podaci su podaci čija se distribucija može objasniti modelom, dok su vanpopulacijski podaci koji se ne mogu ukomponirati u model. Algoritam je prvi puta objavljen od strane Fischler i Bolles (1981).

Konvencionalne tehnike počinju sa što više podataka te pokušavaju eliminirati van-

populacijske podatke dok algoritam RANSAC s druge strane koristi najmanji mogući početni skup podataka i povećava ga podacima koji pripadaju modelu. Ideja je slučajnim odabirom minimalnog uzorka hipotetizirati traženi model. Hipoteze se zatim evaluiraju nad svim elementima ulaznog skupa podataka i algoritam vraća najbolju hipotezu.

Ako model M sadrži p parametara za čiju estimaciju je potrebno k podataka, algoritam slučajnim odabirom izabire podskup S_1 veličine k iz skupa S i na temelju njih definira model M . Koristeći estimirani model M određuje se podskup S_1^* skupa S čiji članovi odgovaraju modelu M s odustupanjem manjim od unaprijed zadanog praga. Skup S_1^* naziva se konsenzusnim skupom. Algoritam ponavlja navedeni korak unaprijed zadani broj puta i vraća parametre modela kome pripada najveći konsenzusni skup.

Algorithm 1: RANSAC

Input:

podaci - skup podataka koji sadrži populacijske i vanpopulacijske podatke

s - minimalni broj podataka potrebnih za estimaciju modela M

τ - prag koji definira slaže li se podatak d_i s modelom M

Output:

M^* - estimacija modela kojoj pripada najveća konsenzusna grupa

```

1 maksimalniKonsenzusSkup  $\leftarrow \emptyset$ 
2  $N \leftarrow \binom{n}{s}$ 
3 for  $i \leftarrow 1$  to  $N$  do
4   konsenzusSkupi  $\leftarrow \emptyset$ 
5   Slučajnim odabirom izabrati  $s$  elemenata i estimirati model  $M_i$ 
6   for  $j \leftarrow 1$  to kardinalitet(podaci) do
7     if Odgovara( $M, d_j, \tau$ ) then
8       konsenzusSkupi  $\leftarrow d_j$ 
9   if kardinalitet(maksimalniKonsenzusSkup) <
      kardinalitet(konsenzusSkupi) then
10    maksimalniKonsenzusSkup  $\leftarrow$  konsenzusSkupi
11 return  $M^*$  definiran maksimalniKonsenzusSkup-om
```

2.2.1. Određivanje vrijednosti parametara algoritma

RANSAC algoritam je upravljan s tri parametra: maksimalno odstupanje podataka od modela (τ), broj iteracija (N) i veličina podskupa pomoću kojeg se estimira model M (s). Potrebno je odrediti optimalne vrijednosti tih parametara kako bi se postigle što bolje performanse algoritma.

Ukoliko je poznata distribucija udaljenosti populacijskih podataka moguće je analitički odrediti vrijednost maksimalno dopuštenog odstupanja populacijskih podataka od modela, no u većini slučajeva distribucija je nepoznata te se vrijednost procjenjuje eksperimentalno.

Prilikom biranja podskupa pomoću kojeg se estimira model cilj je odabrati podskup podataka u kojem nema vanpopulacijskih podataka. Vjerojatnost da će biti izabran takav podskup veličine s iz skupa podataka veličine n , među kojima o su vanpopulacijski podaci je

$$\frac{\binom{n-o}{s}}{\binom{n}{s}} = \frac{(n-s)(n-s-1)\dots(n-s-o+1)}{n(n-1)\dots(n-o+1)} \quad (2.15)$$

Iz (2.15) je vidljivo kako vjerojatno raste smanjivanjem parametra s . Dakle treba izabrati minimalnu moguću vrijednost s , koja je jednaka minimalnom broju elemenata potrebnih za estimaciju modela.

I konačno potrebno je odrediti broj iteracija algoritma, tj. koliko različitih podskupova ćemo uzeti kao bazu za računanje parametara modela da budemo uz zadovoljavajuću vjerojatnost sigurni da barem jedan od tih podskupova ne sadrži vanpopulacijske elemente. Kada bi se provjeravali svi podskupovi kojih ima $\binom{n}{s}$, u većini slučajeva bi to bilo računalno neizvedivo, osim u slučaju kad je n mali broj ili $s \simeq n$.

Kako bi se smanjio teret računanja, zahtjev za podskupom bez vanpopulacijskih elemenata transformira se u vjerojatnosni zahtjev koji traži da podskup bez vanpopulacijskih elementata bude izabran uz vjerojatnost p . Neka je w vjerojatnost biranja populacijskog elementa a $\epsilon = 1 - w$ vjerojatnost biranja nepopulacijskog elementa. Tada je potrebno provesti najmanje N izbora podskupova, gdje je $(1 - w^s)^N = 1 - p$ što daje

$$N = \frac{\log(1-p)}{\log(1-w^s)} = \frac{\log(1-p)}{\log(1-(1-\epsilon)^s)} \quad (2.16)$$

Treba napomenuti da odabir podskupa bez vanpopulacijskih elemenata nije garancija da će se model moći estimirati. Za uspješnu estimaciju takav podskup je nuždan ali ne i dovoljan uvjet. Najčešće podskup mora zadovoljiti i neka dodatna ograničenja specifična za problem čiji model se estimira (npr. za određivanje ravnine, tri izabrane točke moraju biti nekolinearne).

3. Ispitni skupovi

3.1. Ispitni skup KITTI

Skup ispitnih snimki za evaluaciju stereo vida pod akronimom KITTI objavio je *Karlsruhe Institute of Technology*. Skup se sastoji od vožnji u različitim okruženjima snimljenih stereo kamerom montiranom na krov automobila. Osim stereo snimki ispitni skup također uz ostale senzore sadrži i GPS koordinate kamere za svaku sliku. Slike su rektificirane s danim parametrima kalibracije i rektifikacije. Rezolucija slika ovog skupa jednaka je 1241x376 piksela, a razmak između kamera jednak je 0.537 metara.

3.2. Ispitni skup EISATS

Skup ispitnih snimki EISATS (*ENPEDA Image Sequence Analysis Test Site*) sadrži skup sekvenci slika za evaluaciju stereo vida, optičkog toka, analize kretanja i ostalih tehnika kompjuterskog vida. Skup je nastao u sklopu projekta Sveučilišta u Aucklandu s *Daimler AG* iz Njemačke na kojem se razvijaju algoritmi za percepciju okoline i pomoći vozačima tijekom vožnje. Skup sadrži jedanaest sekvenci slika ceste u različitim vremenskim i prostornim uvjetima. Na slici 3.2 je prikazana jedna slika iz kompjuterski stvorene sekvence.



Slika 3.1: Lijeva i desna slika ispitnog skupa KITTI



Slika 3.2: Lijeva i desna slika ispitnig skupa EISATS

4. Programska izvedba i vanjske biblioteke

4.1. Biblioteka OpenCV

Biblioteka OpenCV (engl. Open Computer Vision library) je biblioteka otvorenog koda (BSD licenca) koja zasad uključuje implementacije nekoliko stotina algoritama vezanih uz računalni vid. Podržana je na svim značajnijim operacijskim sustavima: Linux, Windows, OSX, Android te iOS. Ima modularnu strukturu što znači da uključuje više dijeljenih i statičkih biblioteka. Neki od glavnih modula su:

- core - funkcionalnost jezgre, osnovne strukture i algoritmi
- imgproc - algoritmi procesiranja slika
- video - obrada videa, estimacija gibanja
- calib3d - algoritmi geometrije više pogleda, kalibracija (monokularna i stereo)
- features2d - detektori značajki
- highgui - korisničko grafičko sučelje za prikaz slika i rezultata
- ml - modul za strojno učenje
- gpu - sadrži neke od algoritama iz različitih modula implementirane za izvođenje na GPU
- ...

Ova dobro dokumentirana biblioteka je jedan od najčešće korištenih alata u razvoju novih algoritama računalnog vida. Jedan od razloga je dostupnost sučelja za brojne programske jezike i dovoljno snažna zajednica koja usavršava ovu biblioteku.



Slika 4.1: logo



Slika 4.2: PCL logo

4.2. Biblioteka PCL

Biblioteka PCL (engl. the Point Cloud Library) je biblioteka za procesiranje 2D slika, 3D slika i oblaka točaka. PCL biblioteka je biblioteka otvorenog koda objavljena pod BSD licencom. Besplatna je za komercijalne i istraživačke svrhe. PCL je među-platformna biblioteka podržana na Linux-u, MacOS-u, Windows-ima, Android-u i iOS-u. Za jednostavnije razvijanje, PCL je podijeljena u nekoliko manjih biblioteka koje se mogu kompajlirati odvojeno. PCL okruženje sadrži najsuvremenije implementacije algoritama za:

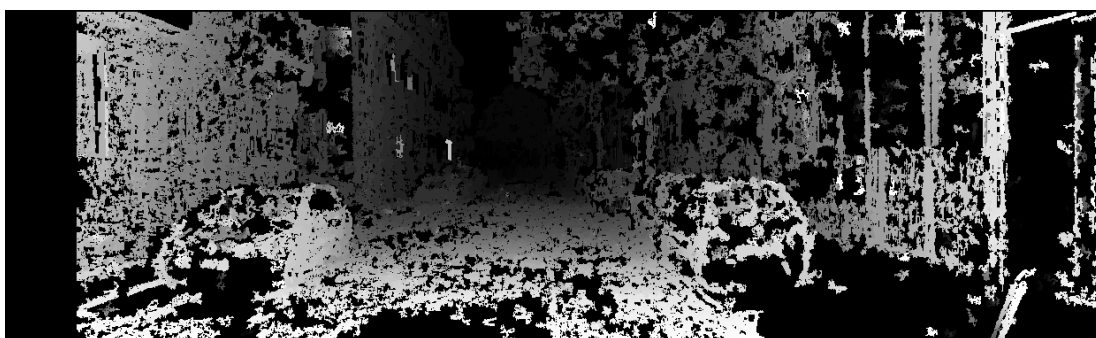
- filtriranje
- estimaciju značajki
- rekonstrukciju površine
- segmentaciju
- estimaciju modela
- ...

Oblak točaka je struktura podataka korištena za reprezentaciju skupa multidimenzionalnih točaka i obično se koristi za prikaz trodimenzionalnih podataka. U 3D oblaku točaka, točke su predstavljene s X, Y, Z geometrijskim koordinatama. Kada je informacija o boji prisutna, oblak točaka postaje 4D.

Oblak točaka je moguće formirati pomoću hardverskih senzora kao što su stereo kamere, 3D skeneri ili generirati pomoću kompjuterskih programa.

4.3. Formiranje oblaka točaka

Kao što je rečeno u opisu biblioteke PCL oblak točaka je struktura podataka korištena za reprezentaciju skupa trodimenzionalnih točaka. Oblak točaka je korišten za vizualizaciju 3D prostora dobivenog iz slika snimljenih stereo kamerom. Pomoću *Block Matchinga* ili *Semi-Global Block Matchinga* metode određena je disparitetna mapa u kojoj za svaki piksel lijeve slike je zapisana udaljenost odgovarajućeg piksela u desnoj slici ako ju je moguće odrediti uz zadovoljavajuću vjerojatnost pripadnosti. Budući da su ulazne slike prethodno rektificirane, za izračun koordinata točaka predstavljenih u oblaku točaka koriste se izrazi izvedeni u (2.10).

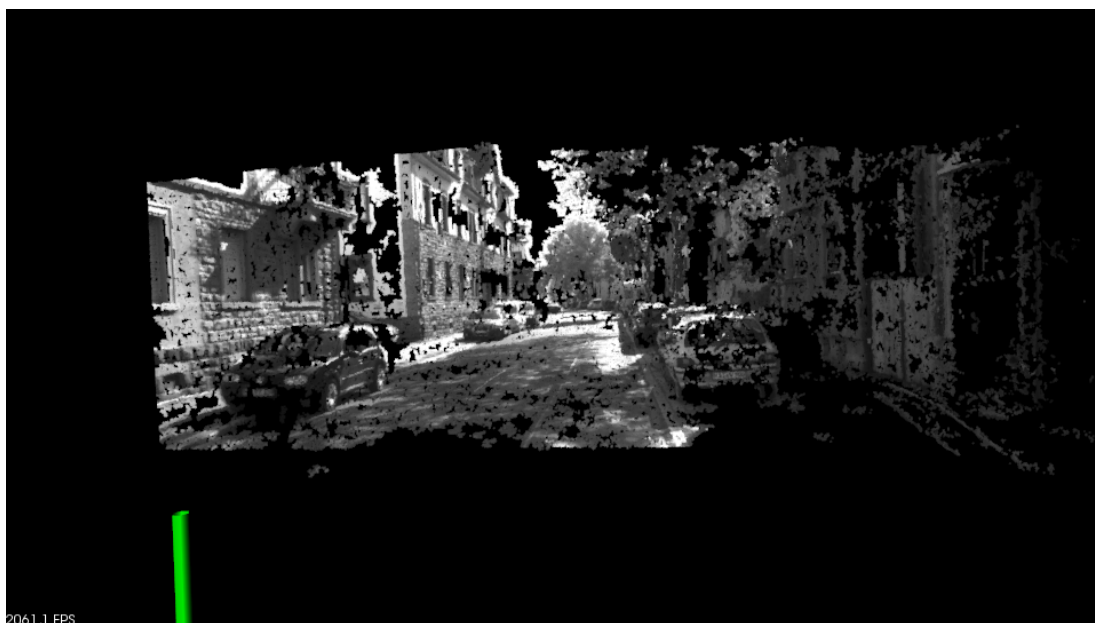


Slika 4.3: Prikaz disparitetne mape

Na slici 4.3 prikazana je disparitetna mapa dobivena *Block Matching* metodom podudaranja. Tamnije nijanse sive predstavljaju manji iznos dispariteta, što znači da je točka udaljenija od kamere. Na slici 4.4 je prikazan oblak točaka formiran iz disparitetne mape.

4.4. Estimiranje ravnine ceste

Kako bi se omogućila detekcija objekata potrebno je odrediti ravninu ceste. Bez ravnine ceste bilo bi gotovo nemoguće odvojiti objekte od pozadine jer u slučaju da optičke osi kamera nisu paralelne s cestom, cesta bi se mogla detektirati kao jedan veliki objekt. Kako bi se pojednostavio postupak traženja ravnine ceste pretpostavljeno je da se stereo sustav nalazi iznad ceste, tj. da je montiran na automobil koji se vozi cestom. Koristeći tu pretpostavku uzet je segment oblaka točaka koji se nalazi ispred pozicije kamere kao što je prikazano na slici 4.5. Zelenom bojom su označene točke nad kojima se vrši estimacija parametara ravnine ceste RANSAC metodom. Točke koje su



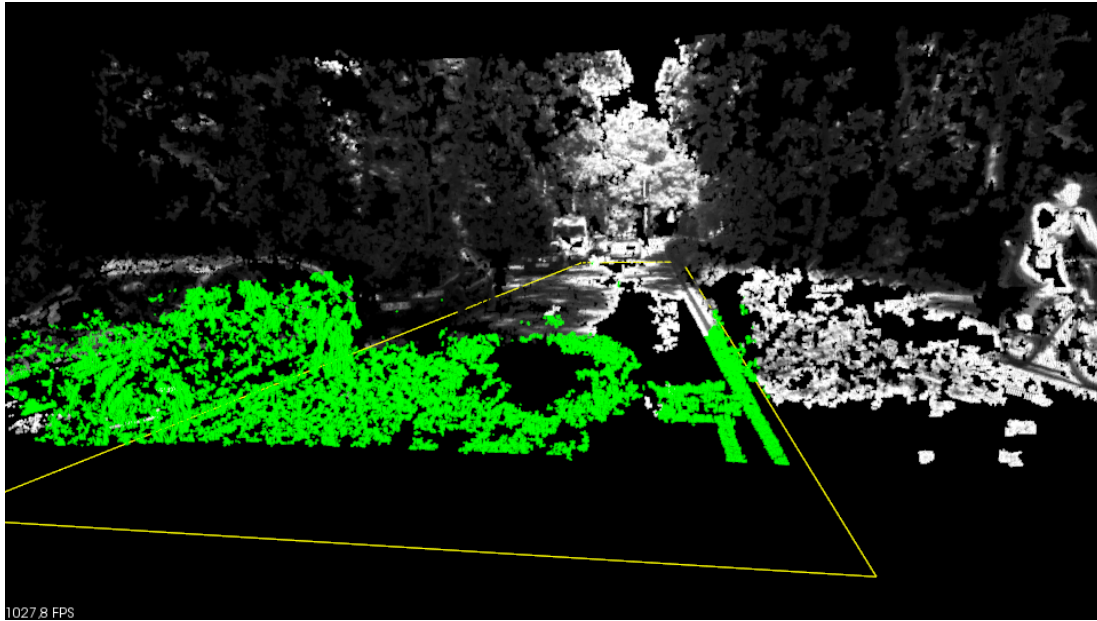
Slika 4.4: Prikaz oblaka točaka

uzete u obzir nalaze se 1.2 metara desno od centra kamera i 3 metra lijevo koristeći pretpostavku da automobil na kojeg su montirane kamere se nalazi u desnom voznom traku.

Nad odobranim skupom točaka pomoću RANSAC metode kao što je opisano u prethodnim poglavljima procjenjuju se parametri ravnine koja odgovara cesti. Ukoliko ravnina ceste ne leži u XZ-ravnini koordinatnog sustava oblaka točaka sve točke oblaka se rotiraju i transliraju kako bi ravnina ceste odgovarala XZ-ravnini. Na slici 4.5 žutom bojom je označena estimirana ravnina ceste. Problem prilikom estimiranja parametara ceste može nastati ukoliko se drugi automobil nalazi neposredno ispred kamere ili ako nedostaju točke koje predstavljaju cestu što je posljedica nemogućnosti određivanja kvalitetne disparitetne mape.

4.5. Rešetkasta podjela oblaka točaka

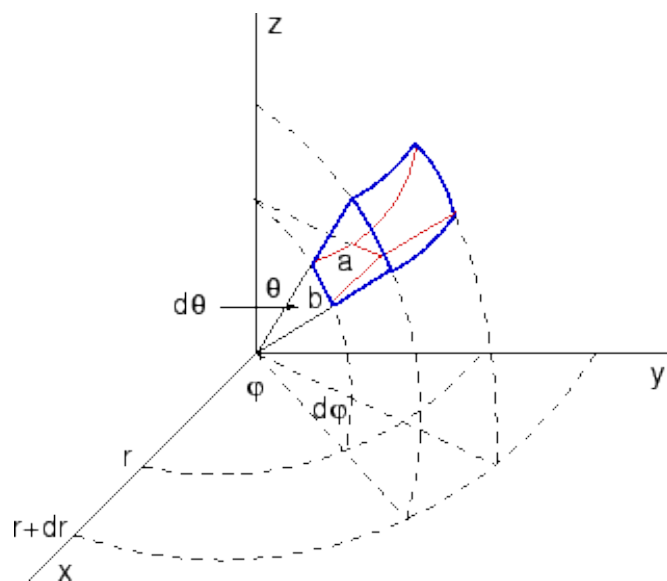
Podjelom prostora na blokove dobiva se prostor smanjene rezolucije pri čemu se smanjuje daljnja računaska složenost i eliminiraju manji šumovi. Točke oblaka su transformirane u sferni koordinatni sustav i pridružene blokovima koji odgovaraju njihovim sfernim koordinatama. Sferni koordinatni sustav ima ishodište jednako poziciji kamere. Kao posljedica korištenja sfernog koordinatnog sustava blokovi udaljeniji od kamere zauzimaju veći volumen što je prednost budući da udaljeniji objekti na slici



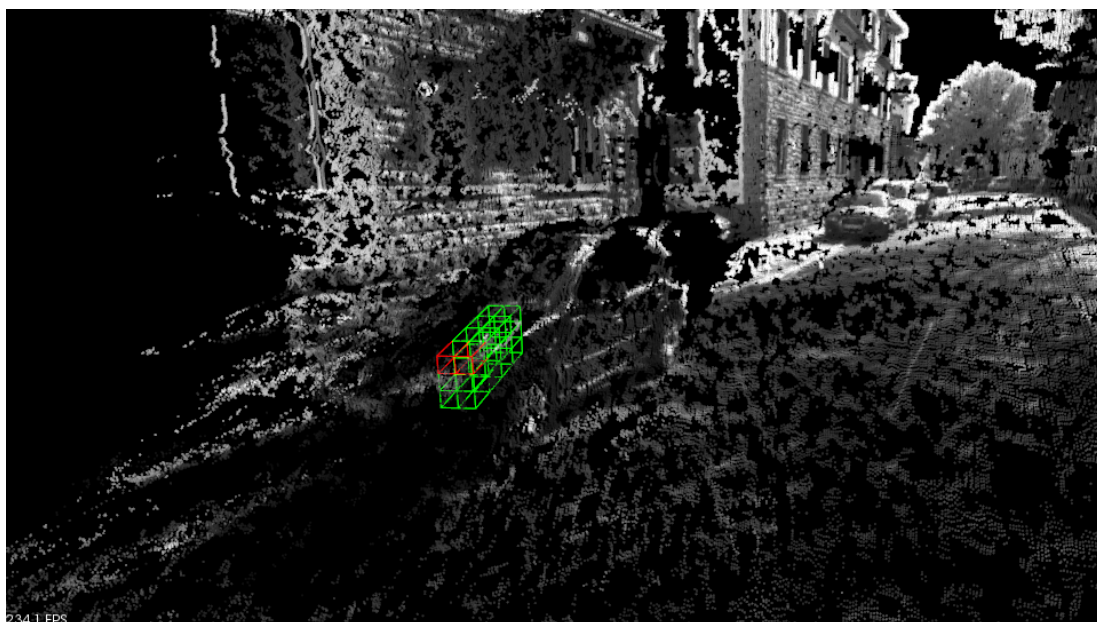
Slika 4.5: Skup točaka za estimaciju ravnine ceste

imaju manju rezoluciju, tj. o njima slika daje manju količinu podataka (točaka) nego o bližim objektima.

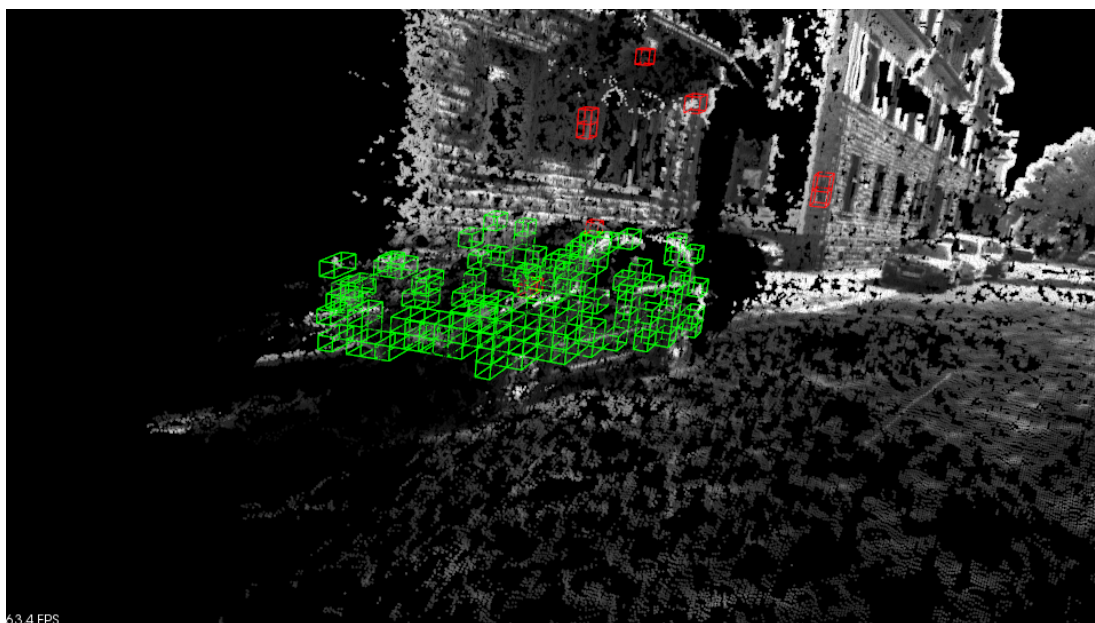
Na slici 4.7 je prikazana skupina susjednih blokova, gdje su zelenom bojom označeni *slobodni* blokovi a crvenom *okupirani* blokovi. Blok je označen kao okupiran ako sadrži više od unaprijed zadanog broja točaka. Tako označeni blokovi su korišteni u nastavku kao osnova za grupiranje blokova u objekte.



Slika 4.6: Sferni koordinatni sustav



Slika 4.7: Blok u oblaku točaka



Slika 4.8: Objekt i sumovi

4.6. Grupiranje blokova u objekte

U prethodnom odjeljku opisana je podjela prostora na blokove i pridjeljivanje točaka oblaka tim blokovima. Svaki blok koji sadrži više točaka od zadanog praga zatim se proglašava zauzetim. Zatim se za svaki zauzeti blok u susjedstvu traže drugi zauzeti blokovi te se proglašavaju zajedno jednim objektom. Postupak se ponavlja za svaki novopronađeni blok objekta. Promatranjem rezultata zaključeno je da objekti sastavljeni s manje od četiri bloka predstavljaju šum i odbacuju se.

Na slici 4.8 zelenom bojom označeni su blokovi koji pripadaju zajedničkom objektu, u ovom slučaju automobilu dok crvenom bojom su označeni blokovi koji su odbaceni kao šum. Kao što je vidljivo to su nakupine točaka koje se zbog netočnog izračuna dispariteta nalaze na krivim pozicijama u prostoru i predstavljaju šum.

4.7. Prikaz objekata na slici

Prikaz i rad s podacima u oblaku točaka je jednostavniji i lakše je uočiti greške u izračunatim disparitetima i razlike u dubini nego promatrajući disparitetnu mapu, ali kada je potrebno provjeriti rad algoritma na sekvenci slika onda je poželjno pripadnosti točaka objektima preslikati natrag na jednu od izvornih slika.



Slika 4.9: Izvorna slika s označenim objektima

Za svaku točku u oblaku ako pripada objektu, odgovarajuću točku na slici označi se rednim brojem tog objekta i svi pikseli koji pripadaju istom objektu uokvireni su pravokutnikom. Na slici 4.9 je prikazan jedan okvir iz testne videosekvence.

5. Eksperimentalni rezultati

Detekcija objekata u gustoj stereoskopskoj rekonstrukciji je testirana većinom na KITTI ispitnom skupu jer korištene scene iz EISATS ispitnog skupa su kompjuterski stvorene i vrlo jednostavne te samim time ne zanimljive.

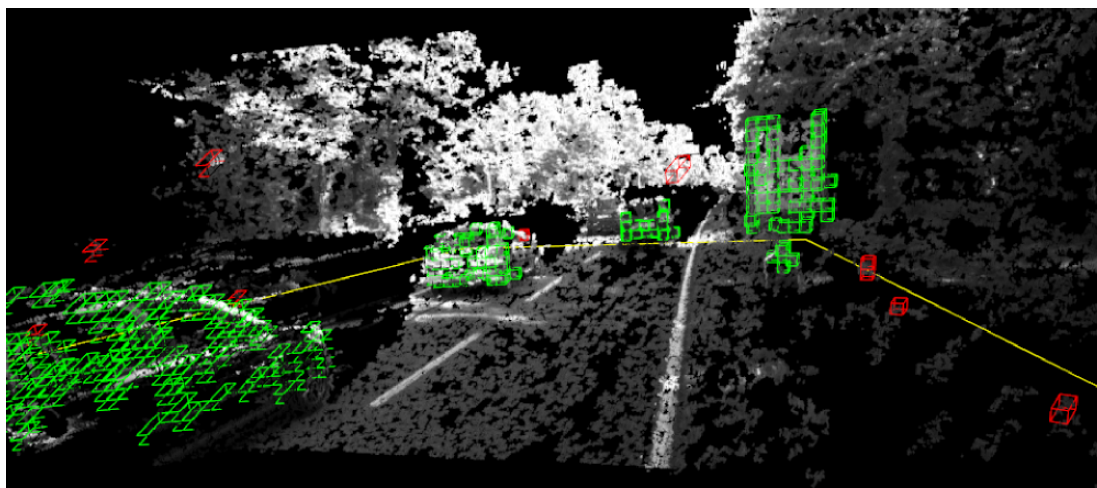
Glavni cilj ovog rada je bio u realnom vremenu što brže pronaći objekte u paru stereo slika kako bi se ti objekti mogli naprednijim tehnikama ispitati i adekvatno označiti ili npr. u slučaju znakova pročitati njihovo značenje. Kao što je prethodnim poglavljima opisano *Semi-global block matching* je vrlo pouzdana i robusna metoda podudaranja koja daje izvrsnu disparitetnu mapu, ali što se tiče performansi, svaki par slika iz KITTI ispitnog skupa se obrađuje 1.05 sec što je presporo za rad u stvarnom vremenu ako uzmemo u obzir da je to samo korak predprocesiranja. Stoga cilj je bio upogoniti i testirati rad lokalne *Block matching* metode te na njoj testirati detekciju objekata u gustoj stereoskopskoj rekonstrukciji.

Kao što je i očekivano *Block matching* metoda je brža od *Semi-global block matching* metode, dok joj je disparitetna mapa dosta lošije kvalitete. Kako bi se dobila predodžba kolike su razlike u performansama u tablici 5.1 su prikazani prosječni rezultati za prvu video sekvencu iz KITTI ispitnog skupa. Sekvenca sadrži 86 parova slika i svaka slika je dimenzija 1240 x 375 piksela. Iz tablice je vidljivo da lokalna metoda *Block Matching* ima 63% kraće vrijeme izvođenja ali također dobivena disparitetna mapa ima 55% manje točaka. *Semi-global block matching* nalazi 446420 disparitetnih parova od 465000 točaka u slici, ako oduzmemo lijevi rub slike za koji ne može izračunati disparitet (jer desna slika ne pokriva to područje) zaključujemo da SGBM pronalazi disparitetne parove za gotovo sve točke slike.

Prilikom pretraživanja prostora unaprijed su postavljene granice unutar kojih se pretražuje. Na gornjoj slici 5.1 je takav prostor pretraživanja označen žutim poligonom. Taj prostor pokriva 20 metara ispred automobila i 5 metara sa svake strane. Eksperimentalno su te granice postavljene nakon testiranja i uviđanja da prostor izvan tog

Algoritam	Vrijeme izvođenja	Broj tocaka
Block matching	0.39 sec	201 647
Semi global block matching	1.06 sec	446 420

Tablica 5.1: Usporedba performansi BM i SGBM



Slika 5.1: Izvorna slika s označenim objektima

poligona ne donosi bitne informacije nego uglavnom sadrži samo pozadinu ili objekte koji nisu od velikog značaja vozaču. Na donjoj slici 5.1 su prikazani objekti pronađeni i označen na oblaku iznad.

Kao što je navedno na početku poglavlja, testiranja su provedena na ispitnom skupu KITTI, tj. na sedam sekvenci slika koje predstavljaju različite situacije. Na slici 5.2 prikazani su uspješni primjeri obrade na scenama vangradske vožnje, dok na slici 5.3 scenama gradske vožnje.

Prva sekvenca je snimana na otvorenoj cesti u gustom prometu i sadrži 335 parova slika. Na gornjoj slici 5.2 je prikazan rezultat obrade jednog od parova prve sekvence. Druga sekvenca je snimana na otvorenoj cesti s vrlo rijetkim prometom i prikazana je



Slika 5.2: Scene vangradske vožnje

na srednjoj slici 5.2. Treća sekvenca prikazuje autoput i prikazana je na donjoj slici 5.2. Druga i treća sekvenca sadrže 429 parova slika odnosno 836 parova slika.

Rezultati obrade četiri različite sekvence gradskog prometa prikazane su na slici 5.3. Sadrže redom 227, 110, 227 i 87 parova slika.

Unutar gore navedenih sekvenci dolazi do tri tipa pogreške

- objekt nije pronađen kao cjelina nego je pronađeno više manjih objekata
- pronađen je objekt koji nismo tražili (false positive error)
- susjedni objekti su spojeni u jedan zajednički objekt (false negative error)

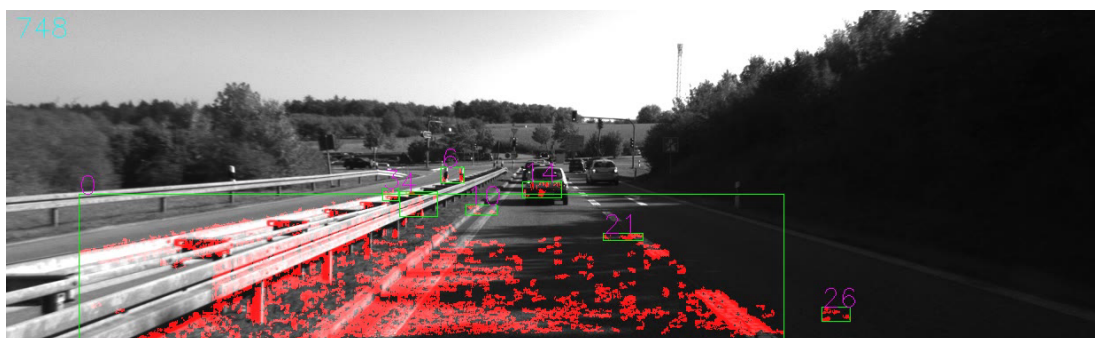
Na slici 5.4 je prikazan primjer prvog tipa pogreške u kojem automobil nije pronađen kao jedinstven objekt nego je sastavljen od nekoliko manjih. Na slici su objekti 0 i 1



Slika 5.3: Scene gradske vožnje



Slika 5.4: 1. primjer pogrešne detekcije objekata

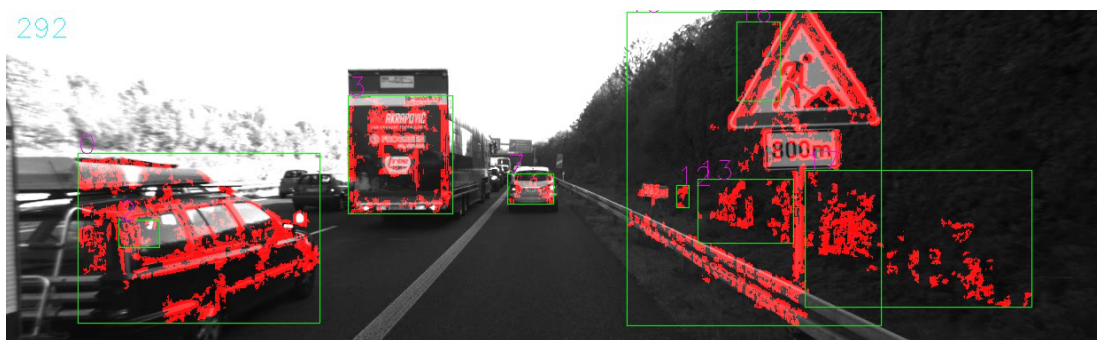


Slika 5.5: 2. primjer pogrešne detekcije objekata

označeni kao odvojeni dok bi zapravo trebali biti jedno. Jednako tako s desne strane ceste grmovi su označeni kao puno manjih objekata iz istog razloga. Ta greška nastaje zbog velikih rupa u disparitetnoj mapi, tj. mnogi pikseli nemaju dodjeljen disparitet zbog ne robusnosti metode podudaranja.

Drugi tip pogreške nastaje zbog jakog šuma prilikom računanja disparitetne mape ili pogrešno izračunate ravnine ceste. Prvi uzrok je jednak kao kod prvog tipa pogreške, dok pogrešno izračunati parametri ravnine ceste su posljedica loše odabranih točaka nad kojima se provodi RANSAC postupak estimacije parametara ravnine. Na slici 5.5 je prikazan slučaj kada zbog ograde koju uzima u obzir, odredi krivi parametre ravnine i kao posljedica toga na krivim mjestima traži objekte.

Treći tip pogreške u kojem se bliski objekti spajaju u jedan zajednički nastaje kao posljedica šuma zbog kojeg algoritam nalazi zajednički blok oba objekti ili zbog činjenice da objekti zaista dodiruju jedan drugoga. Takav primjer je prikazan na slici 5.6 u kojem znak se nastavlja na ogradu i algoritam ih prepoznaje kao jedan objekt.



Slika 5.6: 3. primjer pogrešne detekcije objekata

6. Zaključak

Cilj ovog rada je u stvarnom vremenu pronaći sve objekte u gustoj stereoskopskoj rekonstrukciji u blizini automobila koji bi mogli biti od interesa sustavu koji će služiti kao potpora vozaču pružajući mu informacije kojih možda nije svjestan. Zbog naglaske na brzini izvođenja u formiranju oblaka točaka je korištena OpenCV implementacija lokalne metode podudaranja *Block Matching* koja se pokazala 63% bržom od OpenCV implementacije globalne metode *Semi Global Block Matching*. Naravno to nosi posljedice u obliku puno rijedeg oblaka točaka uz povećani broj smetnji. Prilikom estimacije parametara ravnine ceste korištena je RANSAC metoda implementirana u PCL biblioteci. Oblak u kojem je poznata ravnina dijeli se na blokove i grupiraju se svi susjedni blokovi iznad razine ceste za koje je zaključeno da su zauzeti. Iz rezultata je vidljivo da zbog korištenja lokalne metode podudaranja zbog nedostatka točaka u oblaku, što je direktna posljedica rijetke disparitetne mape, dolazi do dijeljenja objekta u nekoliko manjih objekata.

U budućnosti je plan poboljšati trenutni sustav dodavanjem praćenja okupiranih blokova kroz više okvira sekvence koristeći 3D varijantu stohastičkog polja potpunosti predstavljenog u Badino et al. (2007). Zatim korištenjem usmjerenosti normala točaka, izračunatih na temelju okoline točke, odvojiti objekte različito usmjerenih normala. Naprimjer ograda i znak imaju različito usmjerene normale, tj. znak ima normale usmjerene prema kameri, dok ograda ima normale okomite na pravac kretanja automobila. Dodatno se postupak može optimirati i postići ubrzanje paralelizacijom metode podudaranja budući da se disparitet svakog piksel može računati neovisno o ostalim pikselima.

Primjena detekcije objekata u oblaku točaka ostvarene u ovom radu je u sustavima koji zahtijevaju brz pronalazak položaja objekata na koje će se onda fokusirati algoritmima veće složenosti računanja kao što je naprimjer klasifikacija objekata.

LITERATURA

Hernán Badino, Uwe Franke, i Rudolf Mester. Free space computation using stochastic occupancy grids and dynamic. U *Programming,” Proc. Int’l Conf. Computer Vision, Workshop Dynamical Vision*, 2007.

Stan Birchfield i Carlo Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:401–406, 1998.

Olivier Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT press, 1993.

M. A. Fischler i R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, stranice 381–395, 1981.

Badino Hernán, Franke Uwe, i Pfeiffer David. The stixel world - a compact medium level representation of the 3d-world. U *DAGM-Symposium*, svezak 5748 od *Lecture Notes in Computer Science*, stranice 51–60, 2009.

Heiko Hirschm. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. *IEEE Conference on Computer Vision and Pattern Recognition*, stranice 807–814, 2005.

B. Julesz. Towards the automation of binocular depth perception. *IFIP Congress*, stranice 439–444, 1962.

D. Scharstein i R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, stranice 439–444, 2002.

Detekcija objekata u gustoj stereoskopskoj rekonstrukciji

Sažetak

U sklopu ovog rada razmatra se postupak brze detekcije objekata u oblaku točaka. Specifično je podešen za detekciju objekata u rektificiranim sekvencama snimljenih parom kalibriranih kamera montiranih na vozilu. U okviru rada proučene su i uspoređene metode podudaranja pomoću kojih se izračunava disparitetna mapa iz stereo slike prema kojoj se formira oblak točaka. Zbog težnje za minimizacijom računske složenosti odabrana je lokalna *Block Matching* metoda. Formirani oblak točaka se dijeli na blokove koji se klasificiraju kao zauzeti ili slobodni ovisno o broju točaka koji im pripadaju. Susjedni zauzeti blokovi se zatim grupiraju s ciljem stvaranja objekata. Dobiveni rezultati su analizirani i prikazani u poglavlju s eksperimentima.

Ključne riječi: računalni vid, detekcija objekata, oblak točaka, stereo kamera, estimacija ravnine, 3D rekonstrukcija, triangulacija, epipolarna geometrija, biblioteka OpenCV, biblioteka PCL

Object detection in dense stereo reconstruction

Abstract

This master thesis considers the method of fast object detection in a point cloud. It's specifically tuned for the object detection in the rectified image sequences recorded by pair of calibrated cameras mounted on the vehicle. Stereo matching methods used for disparity map computation have been studied and compared in this master thesis. The local *Block Matching* method has been chosen for minimizing computational complexity. Obtained point cloud is divided into blocks that are classified as occupied or free, depending on the number of points that belong to them. Neighboring occupied blocks are then grouped in order to create objects. The obtained results are analyzed and presented in the chapter with experiments.

Keywords: computer vision, object detection, point cloud, stereo camera, plane estimation, 3D reconstruction, triangulation, epipolar geometry, OpenCV library, PCL library