

# Izrada geometrije 3D prostora na temelju sustava 2D kamera autonomnog vozila

---

**Kelečić, Davor**

**Master's thesis / Diplomski rad**

**2018**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:592276>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-02-05**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**  
**FAKULTET ELEKTROTEHNIKE RAČUNARSTVA I INFORMACIJSKIH**  
**TEHNOLOGIJA**

**Sveučilišni studij**

**Izrada geometrije 3D prostora na temelju sustava 2D**  
**kamera autonomnog vozila**

**Diplomski rad**

**Davor Kelečić**

**Osijek, 2018.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada

Osijek, 24.09.2018.

Odboru za završne i diplomske ispite

**Imenovanje Povjerenstva za obranu diplomskog rada**

<b>Ime i prezime studenta:</b>	Davor Kelečić
<b>Studij, smjer:</b>	Diplomski sveučilišni studij Elektrotehnika, smjer Komunikacije i informatika <sup>1</sup>
<b>Mat. br. studenta, godina upisa:</b>	D 1054, 11.10.2018.
<b>OIB studenta:</b>	27770374746
<b>Mentor:</b>	Izv. prof. dr. sc. Marijan Herceg
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	Danijel Babić
<b>Predsjednik Povjerenstva:</b>	Izv. prof. dr. sc. Mario Vranješ
<b>Član Povjerenstva:</b>	Doc.dr.sc. Ratko Grbić
<b>Naslov diplomskog rada:</b>	Izrada geometrije 3D prostora na temelju sustava 2D kamera autonomnog vozila
<b>Znanstvena grana rada:</b>	<b>Telekomunikacije i informatika (zn. polje elektrotehnika)</b>
<b>Zadatak diplomskog rada:</b>	Dobivanje strukture prostora iz gibanja (engl. structure from motion - SFM) predstavlja dobivanje informacija o geometriji 3D prostora na temelju 2D slika. Na temelju izračunate geometrije prostora autonomno vozilo može poduzimati odgovarajuće radnje (npr. planiranje trajektorije, lokalizacija i sl.). U okviru diplomskog rada potrebno je istražiti metode koje su razvijene za SFM u području autonomne vožnje te dostupne implementacije (Matlab, OpenCV, Point Cloud Library i sl.). Za odabrane metode potrebno je procijeniti njihovu efikasnost s obzirom na točnost procjene geometrije prostora te prikladnost za rad u stvarnom vremenu. Upoznati se s implementacijom SFM u dostupnom SDK-u koji dolazi s razvojnom pločom koja sadrži napredni sustav za asistenciju vozaču (engl. Advanced Driver Assistance System - ADAS)
<b>Prijedlog ocjene pismenog dijela ispita (diplomskog rada):</b>	Dobar (3)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 1 bod/boda Jasnoća pismenog izražavanja: 1 bod/boda Razina samostalnosti: 2 razina
<b>Datum prijedloga ocjene mentora:</b>	24.09.2018.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 13.12.2018.

**Ime i prezime studenta:**

Davor Kelečić

**Studij:**

Diplomski sveučilišni studij Elektrotehnika, smjer Komunikacije i informatika'

**Mat. br. studenta, godina upisa:**

D 1054, 11.10.2018.

**Ephorus podudaranje [%]:**

2

Ovom izjavom izjavljujem da je rad pod nazivom: **Izrada geometrije 3D prostora na temelju sustava 2D kamera autonomnog vozila**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Marijan Herceg

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

# SADRŽAJ

1. UVOD .....	1
2. PREGLED METODA IZRADE GEOMETRIJE 3D PROSTORA NA TEMELJU SUSTAVA 2D KAMERA .....	2
2.1. Definicija problema .....	2
2.2. Tehnike za izradu geometrije 3D prostora iz 2D kamera .....	3
2.2.1. Model kamere.....	3
2.2.2. Triangulacija.....	5
2.2.3. Izdvajanje i spajanje značajki.....	8
2.2.4. Spajanje blokova .....	9
2.2.5. Poluglobalno spajanje .....	10
2.3. Postojeća rješenja za izradu geometrije 3D prostora iz 2D slika .....	12
3. PRIJEDLOG RJEŠENJA ZA DOBIVANJE GEOMETRIJE 3D PROSTORA NA TEMELJU SUSTAVA 2D KAMERA AUTONOMNOG VOZILA .....	16
3.1. Primjer rada rješenja.....	18
3.3. Pokušaj implementacije na ADAS ploču .....	23
3.3.1. ALPHA ploča .....	23
3.3.2. Pokušaj implementacije.....	25
4. VERIFIKACIJA PREDLOŽENOG RJEŠENJA.....	28
5. ZAKLJUČAK .....	32
SAŽETAK.....	33
ABSTRACT .....	33
ŽIVOTOPIS .....	34
LITERATURA.....	35

# 1. UVOD

Rješenje problema pronalaska geometrije 3D prostora iz sustava 2D slika proizlazi iz ljudske sposobnosti spoznavanja udaljenosti pomoću para očiju. Kako su oči razmaknute, čovjek u mozgu dobije dvije slike iz različitih kuteva gledanja. Mozak automatski spaja te dvije slike u jednu cjelovitu sliku s percepcijom udaljenosti. Isti princip je potreban i u ADAS (engl. *Advanced Driver-Assistance Systems*) sustavima gdje bi se koristio za dobivanje geometrije prostora oko automobila. Pomoću geometrije prostora je moguće isplanirati rutu vožnje kojom će se vozilo kretati.

ADAS su napredni sustavi koji pomažu vozaču u vožnji, povećavaju sigurnost u prometu i ugodnost vožnje. Oni se primjenjuju u automobilskoj industriji više od desetljeća. ADAS sustavi koriste različite senzore (kamere, lidar, radar, itd.) postavljene na vozilu kako bi prikupili informacije iz okoline vozila. Računalni sustav prima te informacije i obrađuje ih te sukladno aplikacijskim rješenjima koja su implementirana reagira na te informacije. Neki primjeri ADAS sustava:

- Automatsko parkiranje - automobil u prolazu snima parkiralište te ukoliko ima prazno parkirno mjesto, vozač može odabrati opciju da se automobil sam parkira
- Noćni vid - infracrvena kamera snima prostor ispred automobila te detektirane objekte naglasi na prikazu na kontrolnoj ploči
- Senzor za parkiranje - ovisno o blizini objekta, sustav daje glasnije ili tiše zvučne signale
- ABS - sustav koji omogućuje kontrolu nad vozilom prilikom kočenja na skliskim površinama

U radu je opisan algoritam koji pomoću para kamera stvara dubinsku sliku ispred automobila i pomoću nje pravi 3D geometriju okoline automobila. Neka rješenja rade na principu tako da spajaju samo značajke na slikama, dok neka rješenja spajaju sve elemente slika. U drugom poglavlju su opisana dva postojeća rješenja i potrebne tehnike za njihovo izvođenje. U trećoj cjelini je opisan algoritam pogodan za implementaciju na Alpha ploču te sama implementacija. Četvrta cjelina je testiranje predloženog rješenja.

## **2. PREGLED METODA IZRADE GEOMETRIJE 3D PROSTORA NA TEMELJU SUSTAVA 2D KAMERA**

### **2.1. Definicija problema**

Svakodnevno se u prometu događaju nesreće koje imaju teške materijalne i druge posljedice. Uzroci nesreća često su nepažnja, ometenost ili umor prilikom vožnje što su tipične ljudske pogreške. Kako bi se smanjio broj potencijalnih nesreća razvijaju se ADAS tehnologije koje će pomoći, a potom i u potpunosti zamijeniti čovjeka tijekom vožnje. Na primjer, ako dijete istrči na cestu dok nailazi automobil sustav će preuzeti kontrolu nad automobilom u slučaju da procjeni da vozač neće pravovremeno reagirati. Problem kojim se rad bavi je određivanje udaljenosti objekta ispred vozila.

Problem se rješava tako da se iz snimaka kamera određuje udaljenost vozila od nadolazećih objekata. Snimke tih kamera su ulaz u algoritam koji će određivati geometriju prostora ispred vozila. Određivanjem geometrije prostora ispred vozila, koja je izlazna stavka tog algoritma, omogućuje se planiranje putanje vozila.

Koncept pronalaska 3D geometrije svodi se na uparivanje elemenata slika s istovremeno dobivenih slika iz dvije kamere. Udaljenost pozicija projekcije objekta na slikama je disparitet za taj objekt. Mapa dispariteta sadrži disparitete svih objekata sa slika. Sljedeći korak je računanje inverza geometrijske projekcije, odnosno rekonstrukcija geometrije 3D prostora od svih pronađenih dispariteta. Rekonstrukcija geometrije 3D prostora se računa triangulacijom pomoću mape dispariteta, parametara kamera i udaljenosti između tih kamera.

Za uparivanje elemenata slika se koriste lokalne ili globalne metode. Lokalne metode rade na principu da minimaliziraju više odvojenih funkcija energije na slikama. To rade tako da se uzimaju u obzir cijene podudaranja elemenata slika u prostoru oko promatranog elementa slike. Globalne metode nasuprot lokalnim ne rade na principu odabiranja cijene u prostoru oko promatranog elementa slike nego minimaliziraju energiju slike uzimajući u obzir sve elemente slike na slici.

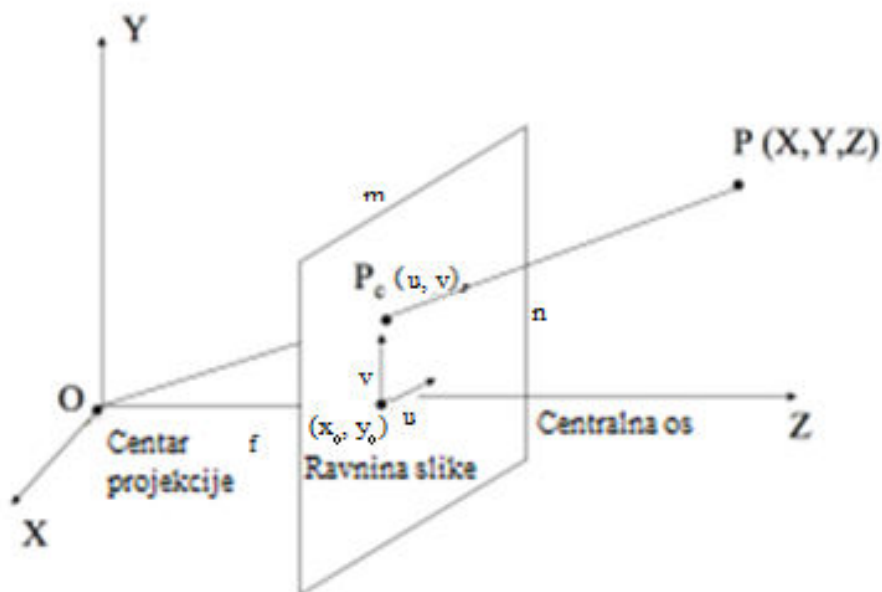
## 2.2. Tehnike za izradu geometrije 3D prostora iz 2D kamera

### 2.2.1. Model kamere

Kamera s točkastim otvorom (engl. *Pinhole camera*) je pojednostavljeni model realne kamere koja projicira objekte iz prostora u ravninu slike. Kamera s točkastim otvorom se sastoji od kutije koja ima rupu na jednoj strani kroz koju svjetlost upada i senzora koji očitava intenzitet svjetlosti koji na njega dolazi. Koriste se tri koordinatna sustava:

- Koordinatni sustav svijeta – trodimenzionalni sustav u kojem se nalaze objekti i scena koje kamera snima
- Koordinatni sustav kamere – trodimenzionalni sustav koji se koristi za definiranje položaja i orijentacije kamere
- Koordinatni sustav slike – dvodimenzionalni sustav u kojem se nalaze intenziteti svjetlosti koji se na nju projiciraju [1]

Na slici 2.1. je prikazana kamera s točkastim otvorom kojoj je ravnina slike postavljena ispred centra projekcije čime se izbjegava invertiranje slike. Ishodište sustava kamere i sustava svijeta se nalazi u istoj točki.



Slika 2.1. Model kamere s točkastim otvorom [2]

Žarišna duljina je udaljenost centra projekcije kamere od ravnine slike i označena je s  $f$ . Točka  $P$  je objekt u sceni koji se nalazi na koordinatama  $X$ ,  $Y$  i  $Z$ , a projicira se na ravninu slike u točki  $P_c$



što predstavlja element slike s koordinatama  $u$  i  $v$ . U izrazima (2 – 1) i (2 – 2) je dan odnos objekta u stvarnom svijetu i njegove projekcije na ravnini slike. Varijable  $m$  i  $n$  predstavljaju dužinu i širinu ravnine slike,  $x_0$  i  $y_0$  su koordinate centra slike, a  $f$  predstavlja žarišnu duljinu.

$$u = \frac{f}{m} * \frac{X}{Z} + x_0 \quad (2 - 1)$$

$$v = \frac{f}{n} * \frac{Y}{Z} + y_0 \quad (2 - 2)$$

Ako se uzme da je ishodište ravnine slike u gornjem lijevom uglu tada izraz (2 – 3) prikazuje homogene koordinate elemenata slike u matičnom zapisu i prikazano je kako se gubi informacija o dubini.

$$\mathbf{Z} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & x_0 & 0 \\ 0 & f & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2 - 3)$$

U izrazu (2 – 4) je prikazana matrica  $\mathbf{K}$  koja se naziva matrica kamere. Vrijednosti u matrici kamere ne ovise o položaju kamere i nazivaju se intrinzičnim parametrima kamere.

$$\mathbf{K} = \begin{bmatrix} f & 0 & x_0 & 0 \\ 0 & f & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2 - 4)$$

U realnim kamerama centar projekcije kamere se ne nalazi u ishodištu koordinatnog sustava svijeta te je u izrazu (2 – 5) prikazana matrica koja povezuje centar projekcije kamere s ishodištem koordinatnog sustava svijeta. Matrica  $[\mathbf{R}|\mathbf{t}]$  predstavlja ekstrinzične parametre kamere i sastoji se od matrice rotacije  $\mathbf{R}$  i vektora translacije  $\mathbf{t}$  i s njom je definiran položaj kamere i njezina orijentacija.

$$[\mathbf{R}|\mathbf{t}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \quad (2 - 5)$$

U izrazu (2 – 6) je prikazan konačni odnos koordinata  $X, Y, Z$  iz sustava svijeta prema sustavu slike u realnim kamerama. Matrica  $\mathbf{P}$  koja se sastoji od matrice ekstrinzičnih i intrinzičnih parametara se zove projekcijska matrica. Prema matrici ekstrinzičnih parametara se transformiraju objekti iz koordinatnog sustava svijeta u koordinatni sustav kamere, a prema

matrici intrinzičnih parametara se transformiraju iz koordinatnog sustava kamere u koordinatni sustav slike, odnosno na ravninu slike.

$$\begin{bmatrix} x^* \\ y^* \\ z^* \end{bmatrix} = \mathbf{P} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} f & 0 & x_0 & 0 \\ 0 & f & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

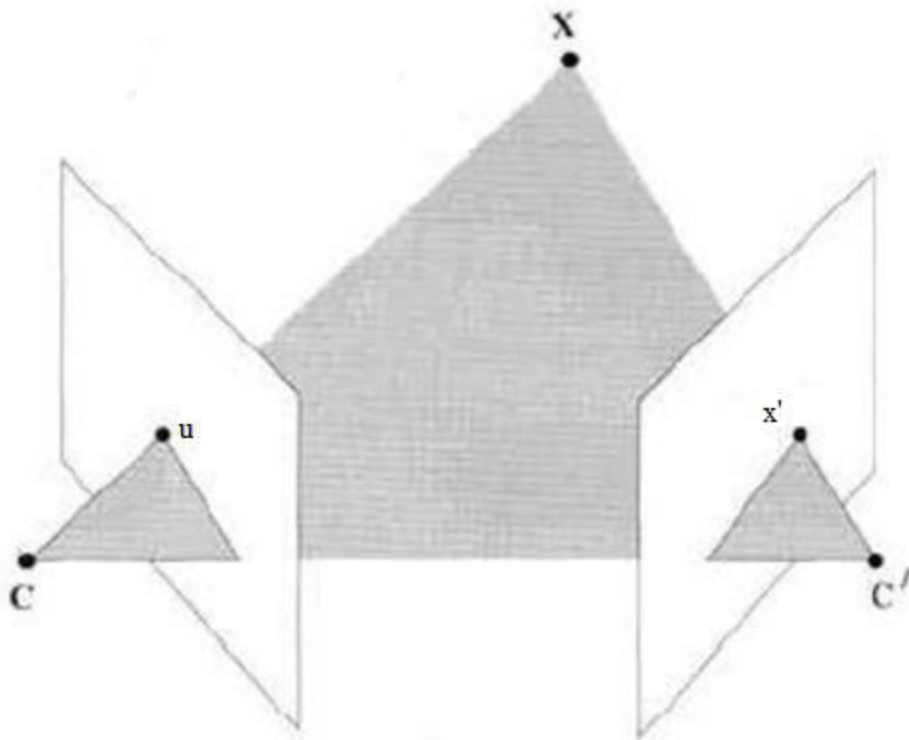
(2 – 6)

$$u = \frac{x^*}{z^*}, v = \frac{y^*}{z^*}$$

Prije obrade slika i primjene algoritama potrebno je kalibrirati kameru. Kalibracija kamere je proces usporedbe objekta na snimci kamere s objektom poznatih geometrijskih veličina kako bi se izračunali ekstrinzični i intrinzični parametri kamere. Osim dosad opisanih parametara postoje i nelinearni intrinzični parametri kamere koji opisuju radijalnu i tangencijalnu distorziju slike. Radijalna distorzija nastaje zbog loma zrake na mjestu upada na leći, a tangencijalna distorzija nastaje jer u realnim kamerama leća i senzor nisu savršeno paralelni. Da bi se kalibrirao stereo sustav kamera potrebno je naći rotacijske i translacijske matrice obje kamere te se iz njih određuje položaj jedne kamere u odnosu na drugu [3].

### 2.2.2. Triangulacija

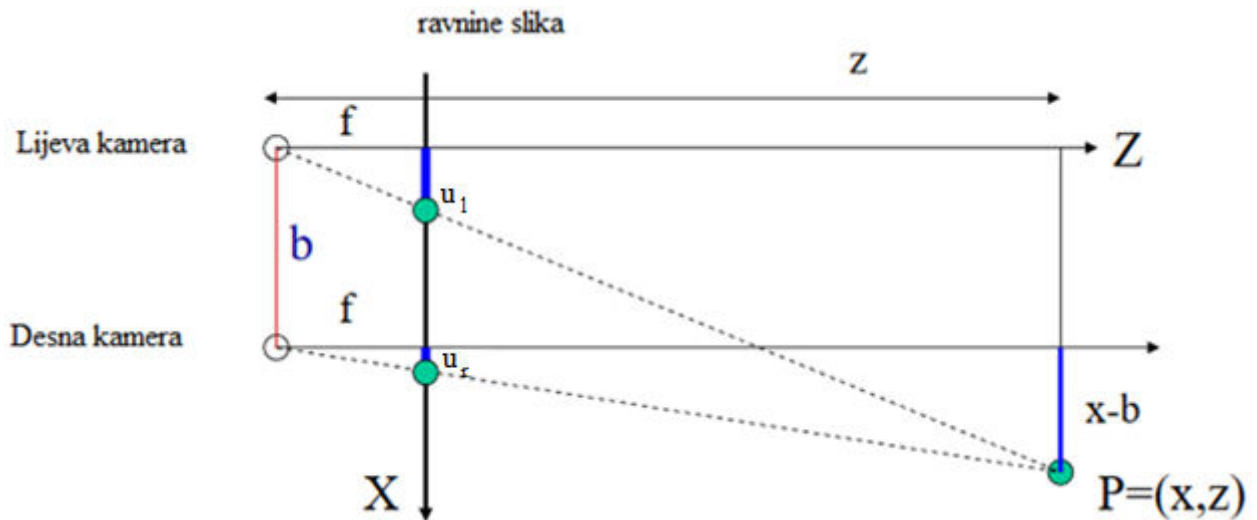
Epipolarna geometrija opisuje odnos između slika s dvije kamere. Na slici 2.2. točka  $X$  predstavlja neku točku koja se projicira na ravnine slike u točke  $u$  i  $x'$ , dok točke  $C$  i  $C'$  predstavljaju centre projekcije tih dviju kamera. Sve tri točke  $X$ ,  $u$  i  $x'$  leže u istoj ravnini te se ta ravnina naziva epipolarna ravnina i na slici je njezina površina obojena sivo. Problematika slučaja je određivanje odgovarajuće točke  $x'$  uz poznatu točku  $u$ . Ona se može odrediti iz epipolarnog ograničenja koje kaže da se tražena točka može nalaziti samo na epipolarnoj liniji. Epipolarna linija se nalazi tamo gdje epipolarna ravnina sječe ravninu slike druge kamere. To je bitno jer se uvelike smanjuje broj točaka na desnoj ravnini slike koje je potrebno provjeriti da li odgovaraju točki  $u$  s lijeve ravnine slike.



**Slika 2.2.** Epipolarna ravnina

Rektifikacija je proces u kojem se slike transformiraju tako da izgledaju kao da su im ravnine slika paralelne i da se nalaze na istim visinama, odnosno da su im  $v$  koordinate jednake. Prethodno se uklanja efekt distorzije. Kamere je fizički nemoguće točno postaviti u taj položaj te se zato koristi rektifikacija. Rektifikacijom ravnine slika postaju paralelne s pravcem koji spaja centre projekcije tih kamera, a epipolarni pravci se nalaze na  $u$  osi na ravninama slika.

Triangulacija je postupak određivanja 3D koordinata nekog objekta iz projiciranih točaka tog objekta na ravninama slika. Na slici 2.3. je prikazana triangulacija na rektificiranoj slici. Prikaz na slici je okomit na  $Y$  os stoga se  $Y$  koordinata ne pojavljuje na slici ali vrijede ista pravila kao i za  $X$  koordinatu. Točkom  $P$  je označen objekt kojem se računaju 3D koordinate u koordinatnom sustavu svijeta,  $b$  označava udaljenost između kamera (engl. *Baseline*), oznaka  $f$  je žarišna udaljenost u kamerama,  $u_l$  i  $u_r$  označavaju  $X$  koordinatu projekcije tog objekta na ravninama slika [3].



Slika 2.3. Triangulacija[4]

Iz poučka o sličnosti trokuta dolazi se do sljedećih izraza:

$$\frac{Z}{f} = \frac{X}{u_l} = \frac{X-b}{u_r} \quad (2-7)$$

$$\frac{Z}{f} = \frac{Y}{v_l} = \frac{Y}{v_r} \quad (2-8)$$

A disparitet se izražava kao:

$$dis_{u,v} = |u_r - u_l| \quad (2-9)$$

Dobiju se izrazi za koordinate objekta u koordinatnom sustavu svijeta:

$$Z = b * f / dis_{u,v} \quad (2-10)$$

$$X = u_l * Z / f \quad (2-11)$$

$$Y = v_l * Z / f \quad (2-12)$$

Udaljenost nije potrebna za 3D geometriju prostora, ali je ljudima lakše percipirati direktnu udaljenost od kamere naspram udaljenosti izražene u 3D koordinatama i ona glasi:

$$udaljenost = \sqrt{X^2 + Y^2 + Z^2} \quad (2-13)$$

### 2.2.3. Izdvajanje i spajanje značajki

Značajke su istaknuti segmenti slike koji se opisuju pomoću svoje okoline. Sve dok su okoline ključnih točaka dovoljno slične, moguće ih je upariti neovisno o njihovoj rotaciji, promjeni svjetline ili ako su skalirane. Najpogodniji algoritam za detekciju značajki je ORB (engl. *Oriented FAST and Rotated BRIEF*) koji je dizajniran tako da koristi manje resursa naspram ostalih metoda, a daje podjednako kvalitetne rezultate u stvarnom vremenu te je besplatan za korištenje. On radi kao kombinacija FAST algoritma (engl. *Features from Accelerated Segment Test*) koji pronalazi značajke na slici i BRIEF algoritma (engl. *Binary Robust Independent Elementary Features*) koji izrađuje deskriptore tih značajki.

Za početak se detektiraju FAST značajke na slici. To se radi tako da se uspoređuju intenziteti svjetlosti između središnjeg elementa slika s intenzitetima svjetlosti elemenata slike na kružnici oko središnjeg elementa slike. Eksperimentalno se FAST-9 pokazao najboljim, što znači da se intenzitet središnjeg elementa slike u kružnici uspoređuje s intenzitetima elemenata slike koji se nalaze u kružnici polumjera devet elemenata slike. Također se eksperimentalno pokazalo da značajke dobivene uzduž rubova imaju jak odziv te se iz tog razloga za njihovo sortiranje primjenjuje Harissova mjera kutova [5]. Parametar koji je potrebno zadati algoritmu je maksimalni broj značajki koje treba pronaći  $N$  te nakon toga se te značajke sortiraju. [6]

FAST algoritam ne posjeduje orijentacijsku komponentu što rezultira manjom vjerojatnošću točne detekcije, stoga ORB algoritam koristi metodu za mjerenje orijentacije kuteva pomoću mjerenja intenziteta težišta. Za orijentaciju se uzima vektor koji se uzima kao pomak od intenziteta težišta do geometrijskog središta tog segmenta slike odnosno značajke. Rosin [7] je definirao količinu gibanja izrazom:

$$moment_{pq} = \sum_{u,v} u^p v^q I(u, v) \quad (2 - 14)$$

Gdje je  $I(u, v)$  intenzitet elementa slike na koordinatama  $u$  i  $v$ . Varijable  $p$  i  $q$  predstavljaju red momenta slike, odnosno utjecaj lokacije elementa slike na težište segmenta slike  $C$  u kojem se elementi nalaze kao što je prikazano u sljedećem izrazu:

$$C = \left( \frac{moment_{10}}{moment_{00}}, \frac{moment_{01}}{moment_{00}} \right) \quad (2 - 15)$$

Nakon toga se radi vektor od intenziteta težišta do središta značajke, a orijentacija se dobiva sljedećim izrazom gdje  $atan2$  predstavlja funkciju arkus tangens koja osim iznosa orijentacije daje i pripadajući kvadrant.

$$\theta = \text{atan2}(m_{01}, m_{10}) \quad (2 - 16)$$

Deskriptori proizašli iz BRIEF algoritma su nizovi bitova koji opisuju odnos središnjeg elementa slike i elementa slike na rubu promatrane kružnice. BRIEF binarni test  $t(\mathbf{p}; u, v)$  daje „0“ ako je središnji element slike svjetliji od rubnog elementa slike ili „1“ ako je tamniji. Varijabla  $\mathbf{p}$  predstavlja segment slike koji se promatra. Ključna točka  $f_n(\mathbf{p})$  se definira vektorom od  $n$  binarnih testova kao što je prikazano u izrazu (2 – 17). Najčešće se uzima da  $n = 256$  binarnih testova.

$$f_n(\mathbf{p}) = \sum_{i=1}^n 2^{i-1} t(\mathbf{p}; u, v) \quad (2 - 17)$$

Moguće je koristiti više tipova distribucije testova, ali najbolje rezultate daje Gaussova distribucija oko središta promatranog segmenta slike. Kako rotacija od svega par stupnjeva jako utječe na uspješnost poklapanja značajki koristi se nadogradnja standardnog BRIEF algoritma. Nadograđeni BRIEF algoritam se naziva rBRIEF (engl. *Rotated BRIEF*). U rBRIEF algoritmu definira se matrica  $\mathbf{S}$  veličine  $2*n$  za svaku ključnu točku na lokaciji  $(u_i, v_i)$ .

$$\mathbf{S} = \begin{pmatrix} u_1 & \dots & u_n \\ v_1 & \dots & v_n \end{pmatrix} \quad (2 - 18)$$

Pomoću orijentacije se dobije matrica rotacija  $\mathbf{R}_\theta$  i uz matricu  $\mathbf{S}$  dobije se  $\mathbf{S}_\theta$ .

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (2 - 19)$$

$$\mathbf{S}_\theta = \mathbf{R}_\theta \mathbf{S} \quad (2 - 20)$$

Te se može izraziti usmjereni BRIEF operator:

$$G_n(\mathbf{p}, \theta) := f_n(\mathbf{p}) | (u_i, v_i) \in \mathbf{S} \quad (2 - 21)$$

Potom se izrađuje tablica s deskriptorima pomoću usmjerenog BRIEF operatora s koracima od 12 stupnjeva te se ona uspoređuje s potencijalnom odgovarajućom značajkom. [6]

#### 2.2.4. Spajanje blokova

Metoda spajanja blokova spada u lokalne metode za uparivanje elemenata slika. Radi na principu usporedbe elemenata slika u prozoru oko promatranog elementa slike i elemenata slika u prozoru oko svakog elementa slike s druge slike koji se nalaze na istoj epipolarnoj liniji. Uspoređivanje se vrši po nekom od kriterija, a sličnost blokova se izražava u cijeni podudaranja, a svaki kriterij računa cijenu na drukčiji način. Najčešće korišteni kriteriji su [8]:

- suma apsolutnih razlika (engl. *Sum of Absolute Differences* – SAD)
- srednja kvadratna pogreška (engl. *Mean Square Error* – MSE)
- apsolutna kvadratna razlika (engl. *Mean Absolute Difference* – MAD)

Te su navedeni kriteriji prikazani u sljedećem izrazu:

$$\begin{aligned}
 SAD(u, v) &= \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{n}{2}}^{\frac{n}{2}} |I_1(u+i, v+j) - I_2(u+i, v+j)| \\
 MSE(x, y) &= \frac{1}{N} \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{n}{2}}^{\frac{n}{2}} (I_1(u+i, v+j) - I_2(u+i, v+j))^2 \quad (2-22) \\
 MAD(x, y) &= \frac{1}{N} \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{n}{2}}^{\frac{n}{2}} |I_1(u+i, v+j) - I_2(u+i, v+j)|
 \end{aligned}$$

Gdje je  $I_1(u+i, v+j)$  element slike s lijeve kamere na koordinatama  $(u+i, v+j)$ , a  $I_2(u+i, v+j)$  element slike sa slike iz desne kamere na koordinatama  $(u+i, v+j)$ . Varijabla  $n$  je veličina bloka koji se uspoređuje, odnosno kako su sve stranice bloka jednake predstavlja veličinu stranice. Varijabla  $N$  označava broj elemenata slika koji se nalaze u bloku. Par elemenata slika koji ima najmanju cijenu podudaranja se odabire kao par koji predstavlja isti objekt na obje slike. Udaljenost koordinata tih uparenih elemenata slika predstavlja disparitet za promatrani element slike. S obzirom na to da se ti elementi slika nalaze na istoj epipolarnoj liniji, njihova  $y$  koordinata je jednaka te se jednadžba za disparitet nalazi u sljedećem izrazu.

$$dis_{u,v} = |u_1 - u_2| \quad (2-23)$$

Ova metoda je odabrana kao najpogodnija za implementaciju na Alpha ploču te će biti detaljno opisana u sljedećem poglavlju.

### 2.2.5. Poluglobalno spajanje

Metoda poluglobalnog spajanja koristi kombinirane koncepte globalnih i lokalnih stereo metoda za precizno spajanje elemenata slika. Metoda se sastoji od tri koraka:

- Izračun mape cijena
- Skupljanje težina po slici
- Odabir dispariteta s najmanjom težinom

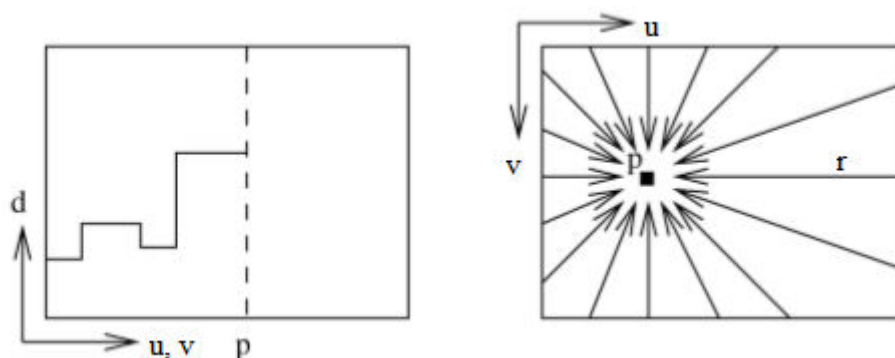
Prvi korak je izračun mape cijena za svaki element slike koji se nalazi na putevima. U originalnom radu [9] Hirschmüller je za prikupljanje mape cijena koristio MI (engl. *Mutual Information*). MI se dobiva prema izrazu (2 – 24) pomoću računanja entropija  $H$  iz dviju slika gdje je  $P$  distribucija vjerojatnosti intenziteta. Entropija je mjera neuređenosti što znači da će slike s različitim percepcijom i većim razlikama imati veću entropiju, a sličnije slike gdje je moguće predvidjeti jednu iz druge će imati manju entropiju. Distribucija vjerojatnosti se ne računa za cijele slike nego samo dijelove koji si odgovaraju inače bi entropije bile konstantne. To se radi tako da se uzimaju u obzir samo odgovarajući redovi  $i$ .

$$MI_{I_1, I_2} = H_{I_1} + H_{I_2} + H_{I_1, I_2}$$

$$H_I = - \int_0^1 P_I(i) \log P_I(i) di \quad (2 - 24)$$

$$H_{I_1, I_2} = - \int_0^1 \int_0^1 P_{I_1, I_2}(i_1, i_2) \log P_{I_1, I_2}(i_1, i_2) di_1 di_2$$

Drugi korak je prikupljanje cijena za svaki od 16 smjerova označenih s  $r$  na slici 2.4. desno. U originalnom radu se predlaže korištenje od osam do 16 puteva.



**Slika 2.4.** Osam optimizirajućih puteva iz različitih smjerova[9]

Za svaki element slike  $es$  s referentne slike i svaki disparitet  $d$  se prikuplja ukupna cijena  $S(es, d)$  na način da se skupljaju cijene svakog elementa za svaki put  $r$  kao što je prikazano na slici 2.4. lijevo. Cijena  $L_r$  za liniju  $u$  u smjeru  $r$  se definira sljedećim izrazom:

$$L_r(es, d) = C_{MI}(es, d) + \min[L_r(es - r, d),$$

$$L_r(es - r, d - 1) + P1,$$

$$L_r(es - r, d + 1) + P1,$$

$$\min(L_r(es - r, i) + P2] - \min L_r(es - r, k) \quad (2 - 25)$$



$C_{MI}$  predstavlja cijenu iz mape cijena izračunate u prvom koraku, varijabla  $P_1$  je cijena za manje promjene u disparitetu te je konstantna, dok je  $P_2$  cijena za veće razlike dispariteta i dana je izrazom (2 – 26). U tom izrazu  $I_{lp}$  i  $I_{Lq}$  predstavljaju susjedne elemente slike na lijevoj slici, a  $P_2'$  je konstantna čija je svrha osigurati da je ispunjen uvjet  $P_1 \leq P$  [1].

$$P_2 = \frac{P_2'}{|I_{lp} - I_{Lq}|} \quad (2 - 26)$$

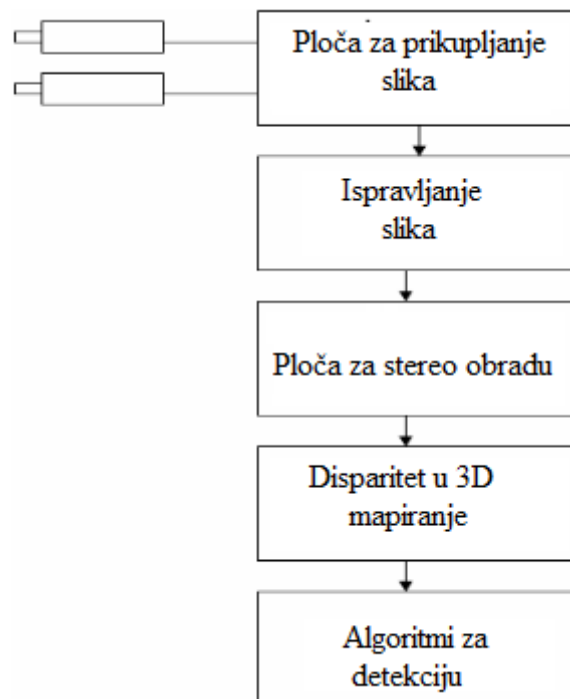
U ovom koraku se za trenutni element slike skuplja unaprijed izračunata cijena te joj se dodaje minimalna cijena puta prethodnog elementa slike umanjena za cijenu tog elementa slike. Nakon toga se zbrajaju cijene svih puteva te se dobiva težina za svaki disparitet  $d$ .

$$S(p, d) = \sum_r L_r(p, d) \quad (2 - 27)$$

Treći korak je odabir dispariteta s najmanjom težinom. Za element slike  $es$  na koordinatama  $(u, v)$  koji ima najmanju težinu  $S(es, d)$  ima disparitet  $d$  [9].

### 2.3. Postojeća rješenja za izradu geometrije 3D prostora iz 2D slika

U prvom rješenju se opisuje način sprečavanja prometnih nesreća pomoću obrade stereo slika koristeći značajke na slikama. Primjer detekcije i uparivanja značajki koji je prethodno opisan u ovom poglavlju. Rješenje je osmišljeno po blokovima kao što je prikazano na slici 2.5.

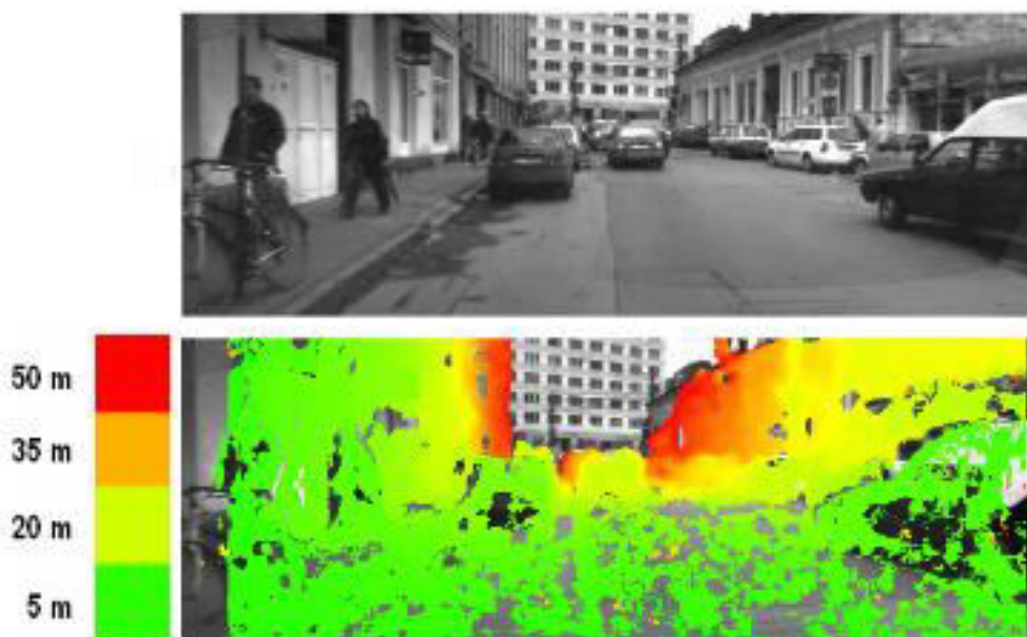


Slika 2.5. Arhitektura sustava prvog rješenja

Sustav za prikupljanje slika uključuje dvije kamere rezolucije 1380x1030 i 6.5 milimetara žarišne duljine. Kamere su postavljene na čvrsti nosač s udaljenosti 320 milimetara između njih. Na sljedećem bloku slike se ispravljaju jer sljedeći blokovi rade sa slikama bez distorzije uzrokovane lećom. Na njima se vrši obrada slike tako da se slike poduzorkuju kako bi se zadovoljili parametri ploče te kako bi smanjile greške na slikama nastale njihovim ispravljanjem. Taj cijeli proces se obavlja u jednom koraku jer se koristi obrnuto mapiranje i bilinearna interpolacija.

3D rekonstrukcija scene se izvodi na predodređenoj ploči. Ulaz u ploču su dvije ispravljene slike, a izlaz može biti dubinska mapa ili Z-mapa u odnosu na lijevu kameru. Na Z-mapi se prikazuje geometrija 3D prostora. Preferira se Z-mapa jer je jednostavnije postaviti automobil u središte koordinatnog sustava dok se objekti nalaze referentno od automobila.

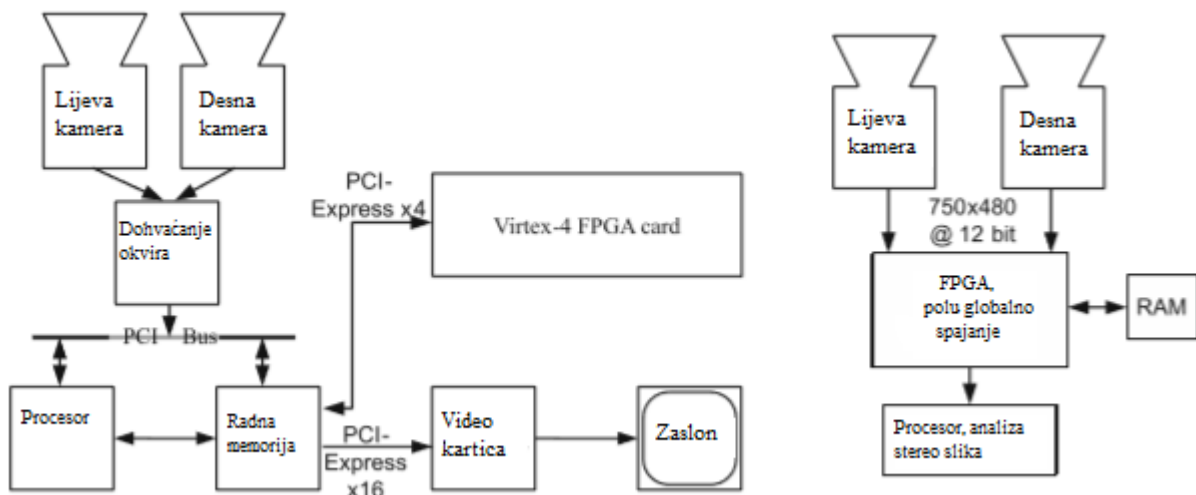
S trenutnim postavkama raspon detekcije je pogodan za gradsku vožnju. Najmanja udaljenost detektiranog objekta je 0.5 metara ispred automobila, odnosno 2.5 metara od kamera. Najveća udaljenost na kojoj se može rekonstruirati scena je 50 metara. Na slici 2.6. je prikazana dubinska mapa generirana stereo vidom te scena od koje je računata dubinska mapa [10].



**Slika 2.6.** Dubinska mapa generirana stereo vidom[10]

U drugom rješenju je objašnjeno kako je izvedena 3D rekonstrukcija pomoću kamera jer dosadašnja rješenja s radarom i lidarom su skupa. U rješenju algoritam poluglobalno spajanje je prvo eksperimentalno implementiran djelomično na računalo i djelomično na FPGA (engl. *Field-*

*Programmable Gate Array*), a potom samo na FPGA. FPGA koji je korišten je lošijih performansi jer je cilj rada implementirati zahtjevni algoritam poluglobalnog spajanja da radi u stvarnom vremenu na slabijem *hardware*-u. Poluglobalno spajanje radi na principu računanja težine za osam puteva kroz sliku. Kako na FPGA postoji ograničenje da je moguće samo četiri pristupa memoriji u jednom trenutku, dolazi do nemogućnosti paralelizacije algoritma. Stoga je moguće paralelno samo četiri puta računati u jednom trenutku. Na slici 2.7. lijevo je prikazano rješenje implementirano preko računala i FPGA, a desno samo na FPGA.

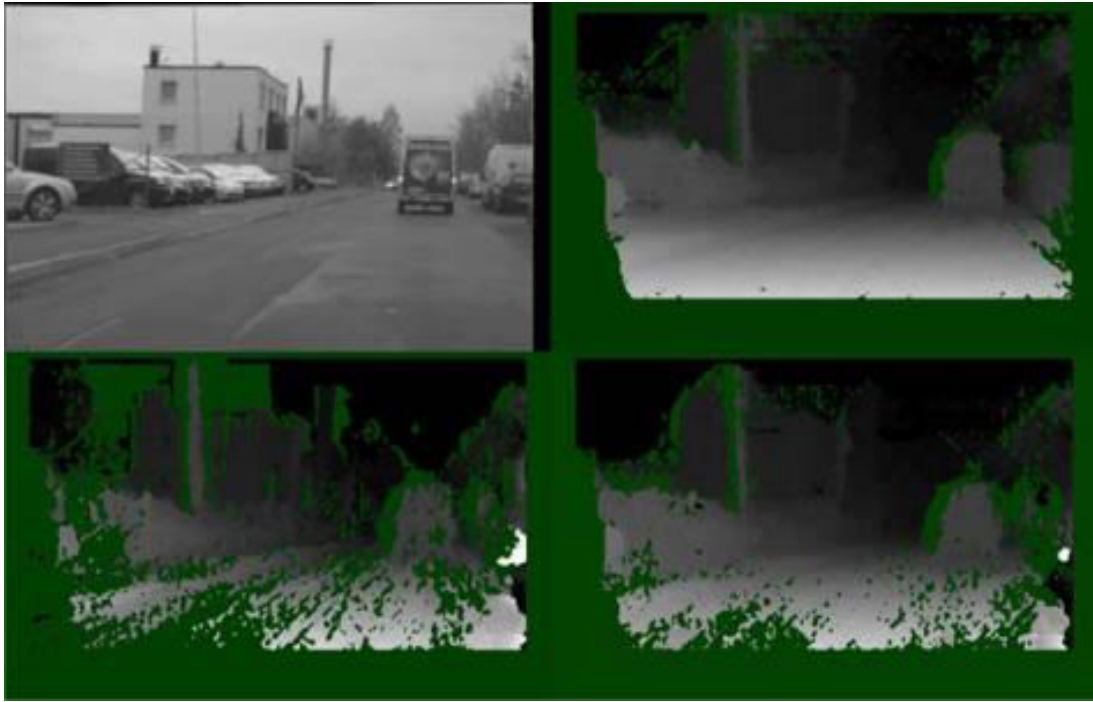


**Slika 2.7.** Blokovski prikaz rješenja, lijevo računalo, desno FPGA

Eksperimentalni dio je implementiran na općenamjenskom x86 procesoru. Par slika se pohranjuju u radnu memoriju. Ulazne slike su rezolucije 750x480. Pred-procesiranje se vrši u procesoru gdje se slike izrezuju na rezoluciju 680x400 jer se scene ne preklapaju na rubovima. Nakon toga slike se šalju na FPGA preko *PCI-Express* gdje je implementiran algoritam poluglobalnog spajanja. Korišteni FPGA je *Virtex-4* koji je za potrebe boljih performansi nego FPGA *Spartan-3* koji se koristi za samostalno rješenje. Dubinska mapa se vraća nazad u radnu memoriju i prikazuje pomoću video kartice na zaslonu.

Nakon toga je rješenje implementirano kompletno na FPGA. U ovom slučaju su kamere direktno spojene na FPGA i na njemu se izvodi cijelo pretprocesiranje ulaznih slika i algoritam poluglobalnog spajanja. Kasnije se ulazne i rezultatne slike šalju na računalo radi analize. Dvije su razlike naspram eksperimentalne implementacije. Prva razlika je što FPGA radi ispravljanje slika u pretprocesiranju. Druga je to što *Spartan-3* posjeduje samo jednu jezgru naspram *Virtex-4* koji posjeduje dvije što mu omogućava provjeru postavljanjem druge kamere kao referentne.

Na slici 2.8. su prikazani rezultati testiranja. Slika gore lijevo prikazuje snimku kamere. Slika gore desno prikazuje algoritam koji je koristio *ZSAD* (engl. *Mean-free sum-of-absolute differences*). Slike dole lijevo prikazuje verziju poluglobalnog spajanja algoritama prema [11], dok slika dole desno prikazuje verziju algoritma s hijerarhijski zajedničkim informacijama. Male vrijednosti dispariteta su označene tamnim bojama, a zelenom bojom su označeni elementi slika koje nije moguće upariti.



**Slika 2.8.** Rezultati testiranja rješenja na FPGA s lošim performansama[12]

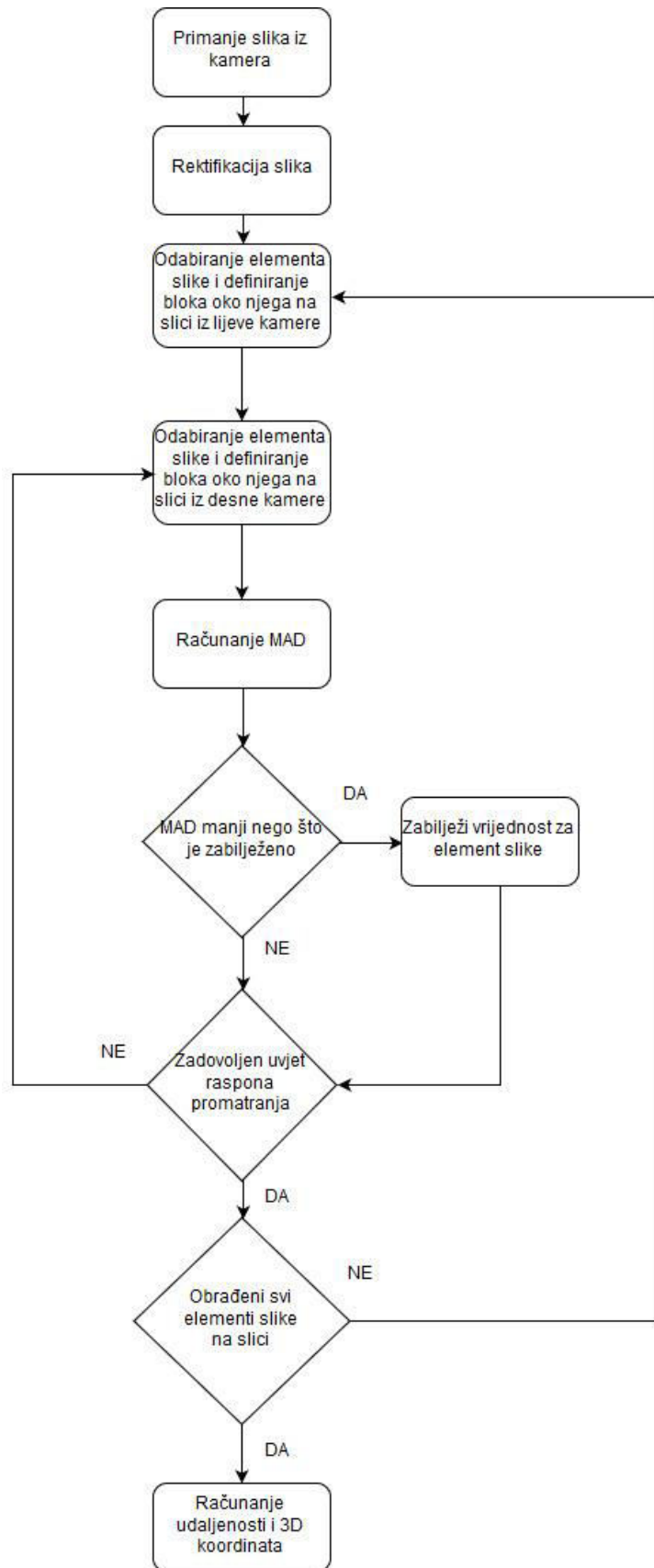
Verzija algoritma koji je koristio *ZSAD* daje nabolje rezultate. Rezultati su ohrabrujući te je rješenje stvarnovremenskog računanja dispariteta pomoću ove tehnike moguće koristeći poduzorkovanje, ponovno korištenje informacija izračunatih na slici pune rezolucije i paralelizacijom većine blokova [12].

### 3. PRIJEDLOG RJEŠENJA ZA DOBIVANJE GEOMETRIJE 3D PROSTORA NA TEMELJU SUSTAVA 2D KAMERA AUTONOMNOG VOZILA

Zadatak diplomskog rada je istražiti pogodan algoritam za izradu geometrije 3D prostora iz seta 2D slika za primjenu u ADAS industriji. Odabrani algoritam je potrebno implementirati na Alpha ploču koja je prototip računalnog sustava za obradu podataka u vozilu. Izlaz algoritma je geometrija 3D prostora koja bi se koristila za planiranje puta kojim će se vozilo kretati.

Rješenje je prvotno razvijeno u C++ programskom jeziku uz upotrebu *OpenCV* (engl. *Open Source Computer Vision Library*) biblioteke. Korištena je tehnika *StereoBM* kratko opisana u potpoglavlju 2.2.4. Spajanje blokova koja je davala zadovoljavajuće rezultate u pogledu kvalitete, ali nije bio zadovoljen uvjet izračunavanja geometrije 3D prostora u stvarnom vremenu. Rješenje je razvijeno i uz pomoć *StereoSGBM* tehnike. Ta tehnika je opisana u potpoglavlju 2.2.5. Poluglobalno spajanje i ona je davala izvrsne rezultate ali je vrijeme izvođenja višestruko duže nego kod tehnike *StereoBM*.

Radi optimizacije rješenja i mogućnosti implementacije na ALPHA ploču, rješenje je razvijeno u C programskom jeziku i imitacija je tehnike *StereoBM*. Prednost rješenja razvijenog u C programskom jeziku je to što se može podijeliti na sve jezgre procesora na Alpha ploči dok samo jedna jezgra podržava korištenje *OpenCV* biblioteke te se algoritam ne bi mogao podijeliti po jezgrama. Na slici 3.1. je prikazan dijagram toka rješenja razvijenog u C programskom jeziku.



**Slika 3.1.** Dijagram toka algoritma spajanja blokova

### 3.1. Primjer rada rješenja

Prvi korak je primanje slike  $I_1$  iz lijeve kamere i slike  $I_2$  iz desne kamere. Za svaki element slike  $s$   $I_1$  na koordinatama  $(u, v)$ , gdje  $u$  poprima vrijednosti u rasponu od 1 do širine slike  $W$ , a  $v$  u rasponu od 1 do visine slike  $H$ , se određuje disparitet. Slike korištene za prikazivanje rada algoritma su preuzete s [13] iz seta podataka nazvan *2010\_03\_05\_drive\_0017*. Slike su prethodno rektificirane i poznati su ekstrinzični i intrinzični parametri kamera kojim su snimane. Rezolucija im je 1351x374. Za ispravan rad rješenja slike moraju biti monokromatske.



**Slika 3.2.** Slika  $I_1$  iz lijeve kamere



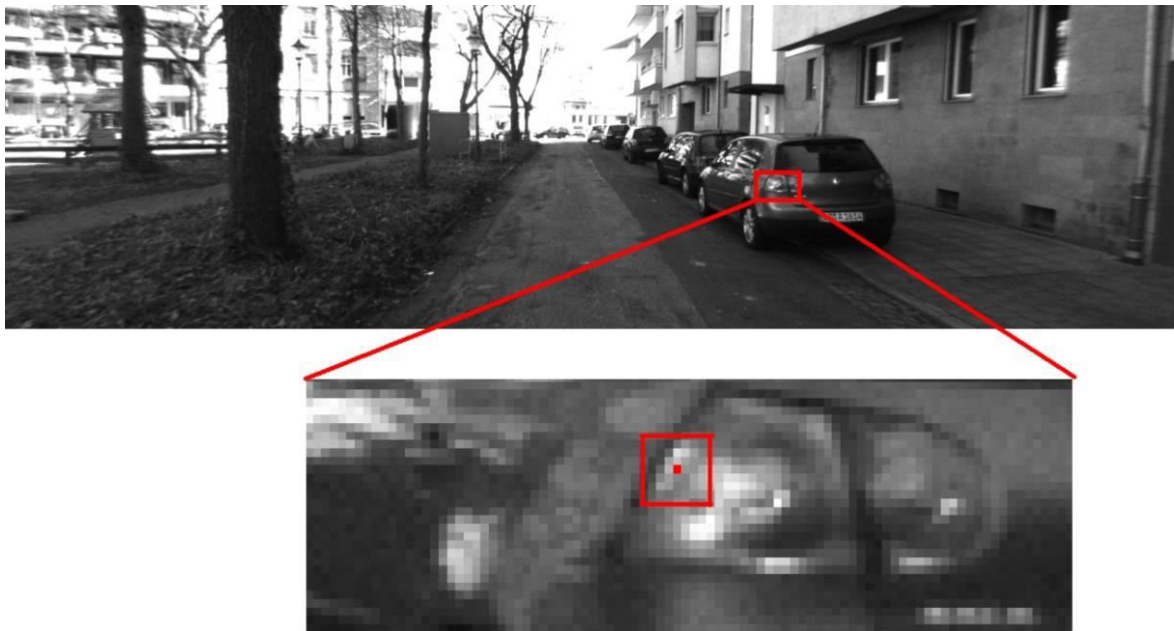
**Slika 3.3.** Slika  $I_2$  iz desne kamere

Sljedeći korak je definiranje bloka  $B_{u,v}$  na  $I_1$  sa središtem u  $(u, v)$  veličine  $M \times M$  prema izrazu (3 – 1). Blokovi se definiraju od gornjeg lijevog ruba lijeve slike i po završetku određivanja dispariteta za promatrani element slike kreću dalje u desno, odnosno dolje. U bloku se nalaze intenziteti svjetlosti obuhvaćenih elemenata slika.

$$B_{u,v} = \begin{bmatrix} I_1(u - \lfloor \frac{M}{2} \rfloor, v - \lfloor \frac{M}{2} \rfloor) & \cdots & I_1(u + \lfloor \frac{M}{2} \rfloor, v - \lfloor \frac{M}{2} \rfloor) \\ \vdots & \ddots & \vdots \\ I_1(u - \lfloor \frac{M}{2} \rfloor, v + \lfloor \frac{M}{2} \rfloor) & \cdots & I_1(u + \lfloor \frac{M}{2} \rfloor, v + \lfloor \frac{M}{2} \rfloor) \end{bmatrix} \quad (3 - 1)$$



U izrazu (3 – 1) je korištena funkcija najveće cijelo (engl. *Floor*) koja vraća najveću cijelu vrijednost od unesenog broja. Varijabla  $M$  predstavlja veličinu bloka koji obuhvaća elemente slika. Do optimalne veličine bloka se dolazi eksperimentalnim putem. Veće veličine bloka će produžiti vrijeme izvođenja, zagladiti sliku i griješiti će kod rubova, dok kod manjih veličina bloka dolazi do uspoređivanja s nedovoljno informacija te dolazi do pogrešnog detektiranja. U primjeru je korištena veličina bloka  $M = 7$ . Na slici 3.4. uvećana je okolina oko  $I_1(883, 204)$ . Crvenom točkom je označen  $I_1(883, 204)$ , a crvenim kvadratom je označen blok  $B_{u, v}$  koji će se uspoređivati s blokovima na slici  $I_2$ .



**Slika 3.4.** Blok na slici  $I_1$  snimljenoj lijevom kamerom

Vrijednosti iz bloka  $B_{u, v}$  su prikazane u tablici 1. Kako su slike  $I_1$  i  $I_2$  u monokromatskom zapisu njihove vrijednosti boje se kreću u intervalu od 0 do 255. Crna boja se označava s 0, a bijela s 255.

89	76	63	115	150	125	110
84	67	99	223	202	159	131
63	71	162	193	163	146	144
60	103	182	185	149	147	172
73	132	158	135	132	164	186
88	162	152	122	131	162	181
93	120	112	124	136	147	156

**Tablica 1.** Vrijednosti elemenata slika definiranih blokom  $B_{u, v}$

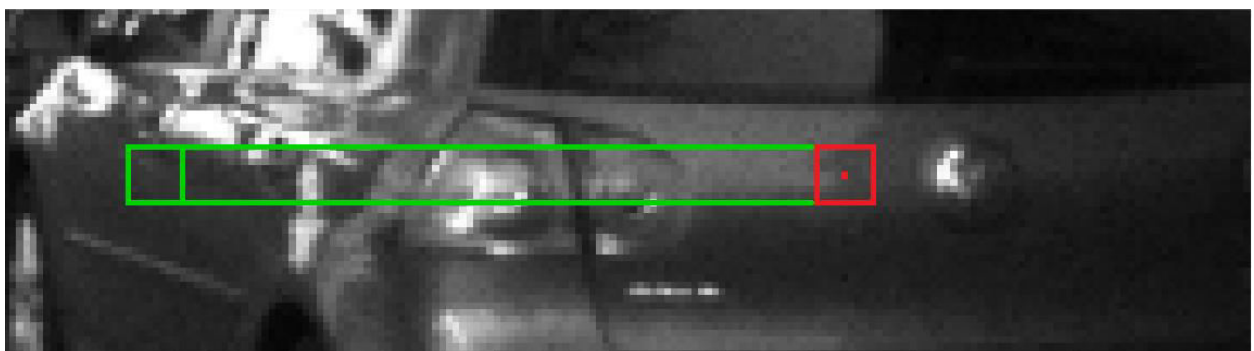


Potom se definira blok  $B'_{u-k, v}$  s kojim će se blok  $B_{u, v}$  uspoređivati. Blok  $B'_{u-k, v}$  ima središte na koordinatama  $(u-k, v)$  i također je veličine  $M \times M$ , a  $k$  se mijenja od 0 do raspona pretraživanja. Raspon pretraživanja je prostor unutar kojeg se definiraju blokovi  $B'_{u-k, v}$  koji će se uspoređivati s blokom  $B_{u, v}$ . Kako su slike prethodno ispravljene, blokovi  $B'_{u-k, v}$  se definiraju na epipolarnoj liniji te se nalaze na jednakoj  $v$  koordinati kao i blok  $B_{u, v}$ .

$$B'_{u-k, v} = \begin{bmatrix} I_2(u - \lfloor \frac{M}{2} \rfloor - k, y - \lfloor \frac{M}{2} \rfloor) & \cdots & I_2(x + \lfloor \frac{M}{2} \rfloor - k, y - \lfloor \frac{M}{2} \rfloor) \\ \vdots & \ddots & \vdots \\ I_2(x - \lfloor \frac{M}{2} \rfloor - k, y + \lfloor \frac{M}{2} \rfloor) & \cdots & I_2(x + \lfloor \frac{M}{2} \rfloor - k, y + \lfloor \frac{M}{2} \rfloor) \end{bmatrix} \quad (3-2)$$

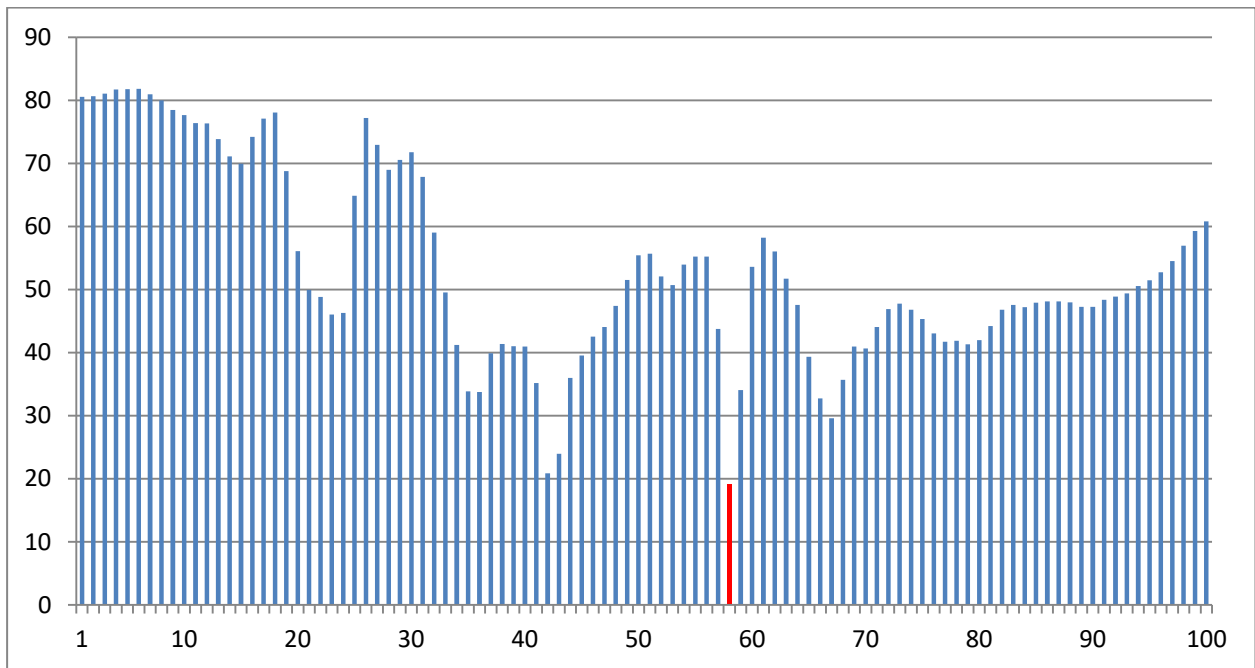
Za bliže objekte potreban je veći raspon pretraživanja te isto vrijedi za duljinu razmaka između kamera zbog veće udaljenosti objekta u scenama koje kamere snime. Ukoliko je razmak između kamera prevelik, kamere će zbog različitih percepcija različito snimiti isti objekt te on neće izgledati dovoljno slično na slikama obaju kamera da bi ga algoritam mogao ispravno upariti. Povećanjem udaljenosti između kamera dobiva se veća preciznost koordinata objekta i pomiče se maksimalna udaljenost do koje algoritam može odrediti dubinu slike, ali se povećava vrijeme izvođenja koda i postoji mogućnost krivog uparivanja objekata. Raspon pretraživanja se postavlja na fiksnu vrijednost prilikom implementacije te je u primjeru korišten raspon  $0 > k \geq 100$ .

Na slici 3.5. je prikazan uvećani dio sa slike  $I_2$ , koordinate crvenog kvadrata su koordinate bloka sa slike  $I_1$  te se može primijetiti kako blokovi na istim koordinatama ne predstavljaju isti objekt u svijetu. Zelenom bojom je označen prostor koji će se koristiti za definiranje blokova  $B'_{u-k, v}$ , a kvadrat na kraju označava zadnji blok iz raspona pretraživanja koji se uzima u obzir.



**Slika 3.5.** Raspon pretraživanja na snimci  $I_2$

Blokovi  $B_{u, v}$  i  $B'_{u+k, v}$  se uspoređuju pomoću  $MAD$  kriterija prema izrazu u (2 – 21). U slučaju kada se blokovi nalaze na rubu slike i nije moguće prikupiti intenzitete jer nema elemenata slika tada se na njihovo mjesto upisuje 0 i oni se isključuju iz varijable  $N$ . Odnosno  $N$  se smanjuje za svaki element slike kojem nije moguće odrediti intenzitet. Rezultat uspoređivanja je cijena i računa se za svaki blok. Blok  $B'_{u+k, v}$  s najmanjom cijenom je najbližnji promatranom bloku  $B_{u, v}$ . U histogramu 3.1. su prikazane cijene za svaki  $k$ , a crvenom bojom je označen  $k$  s najmanjom cijenom. Najmanja cijena je za  $k = 58$  i blok s tim  $k$  se uzima kao najbližnji promatranom bloku sa slike  $I_I$ .



**Histogram 3.1.** Prikaz cijena za svaki  $k$

U tablici 2. su prikazane vrijednosti bloka  $B'_{u-k, v}$  za koji je  $k = 58$ .

91	82	91	126	162	138	110
85	73	165	200	154	134	152
83	117	191	142	133	148	162
84	139	164	135	147	163	171
103	130	125	127	150	163	177
112	164	119	134	147	148	147
101	109	106	123	125	128	120

**Tablica 2.** Vrijednosti elemenata slika definiranih blokom  $B'_{u+k, v}$  gdje je  $k = 58$

Potom se tom bloku prema izrazu (2 – 22) računa disparitet i zapisuje se u mapu dispariteta na iste koordinate na kojima se blok  $B_{u, v}$  nalazi na slici  $I_l$ . Disparitet je udaljenost između središta uparenih blokova i označava dubinu za taj element slike i koristit će se za računanje geometrije 3D prostora. Mapa dispariteta za cijelu sliku je prikazana na slici 3.6.



**Slika 3.6.** Mapa dispariteta

Kada je izračunata mapa dispariteta za cijelu sliku, počinje korak u kojem se računaju 3D koordinate za svaki element mape dispariteta što predstavljaju geometriju prostora ispred vozila. 3D koordinate se računaju prema izrazima (2 – 10), (2 – 11) i (2 – 12).

Uz primjer sa slika 3.2 i 3.3. su priloženi i parametri kamera kojim su slike snimljene:

- Razmak između kamera  $b = 0.5778$
- Žarišna duljina  $f = 890.6439$
- Koordinate centra na  $u$  osi  $x_0 = 675.1762$
- Koordinate centra na  $v$  osi  $y_0 = 244.4143$

Kada se ishodište ravnine slike postavi na centar slike tada koordinate točaka iz primjera glase:

- Koordinate točke na lijevoj slici  $u1' = 207, v1' = -40$
- Koordinate točke na desnoj slici  $u2' = 149, v2' = -40$

Uvrštavanjem parametara i dispariteta u izraze dobiju se koordinate točke na svjetlu automobila u sustavu svijeta i kamere izražene u metrima (2.06, -0.4, 8.87). Udaljenost od kamere do objekta se dobiva prema izrazu (2 – 13) i iznosi 9.11 metara

### 3.3. Pokušaj implementacije na ADAS ploču

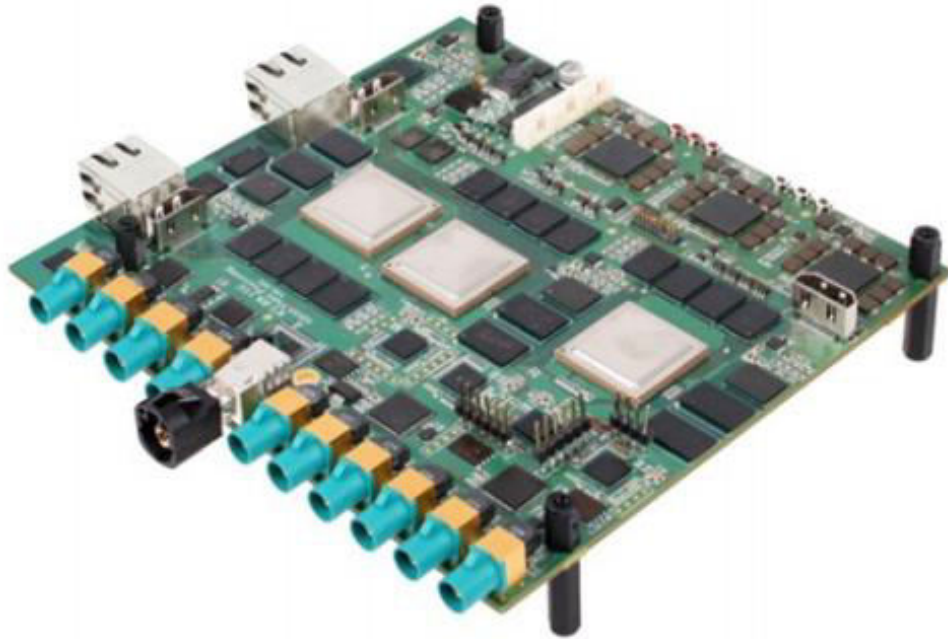
U svrhu lakšeg upravlja automobilom te povećanja sigurnosti vožnje razvija se potreba za ADAS (engl. *Advanced Driver Assistance Systems*) sustavima koji će pomagati vozaču u vožnji. Kako većina prometnih nesreća nastaje kao posljedica umora vozača, ometenosti tijekom vožnje te iz raznih razloga nemogućnosti pravovremenog reagiranja na novonastalu situaciju razvijaju se ADAS sustavi s ciljem smanjenja tih nesreća. Razine autonomnosti su grupirane u šest razina, ovisno koliko je potrebno vozačevo sudjelovanje:

- Nulta razina: osnovna razina u kojoj ne postoje nikakvi ADAS sustavi i vozač samostalno kontrolira sve funkcije vozila
- Prva razina: razina u kojoj vozač obavlja gotovo sve funkcije u vožnji osim nekih specifičnih kao što su skretanje gdje vozilo zadržava smjer vožnje ili ubrzavanje vozila gdje vozilo održava konstantnu brzinu. Vozač mora biti sposoban preuzeti kontrole u svakom trenutku.
- Druga razina: sustav je automatiziran tako da je automatizirano i povezano skretanje i akceleracija ovisno o situaciji u okolini vozila. Vozilo može samo ubrzati, kočiti i kontrolirati upravljač. Primjer je parkiranje vozila bez pomoći vozača. Vozač također mora pratiti promet i biti spreman preuzeti kontrole
- Treća razina: Vozilo može samo voziti u određenim uvjetima, ali je vozač potreban i treba reagirati u slučaju potrebe. Dakako nije potrebno posvetiti pažnju kao u prethodnim razinama
- Četvrta razina: Vozilo je potpuno autonomno i može pravilno i pravodobno reagirati na gotovo sve sigurnosno kritične funkcije. Vozač je još uvijek potreban jer razina ne pokriva sve moguće slučajeve koji se mogu dogoditi u vožnji
- Peta razina: Vozilo je potpuno autonomno te vozač ne mora biti prisutan tijekom vožnje [14]

Razni proizvođači sudjeluju u „utrci“ u kojoj je cilj napraviti vozilo pete razine. Za pokretanje aplikacija za autonomnu vožnju koriste se ADAS ploče. Neki od proizvođača ploča su *NVIDIA* [15], *Renesas Electronics* [16] i *Qualcomm* [17].

#### 3.3.1. ALPHA ploča

ALPHA ploča je ADAS ploča koja koristi tri SoC-a (engl. *System on Chip*) naziva TDA2xx koje je razvila tvrtka *Texas Instruments*. Na slici 3.7. je Alpha ploča na koju se implementira algoritamsko rješenje.



**Slika 3.7.** *Alpha ploča*

Prvi SoC se zove SCV (engl. *Surround Camera View*) i podržava do šest ulaza za RDACM24 tip kamera. Taj čip se koristi za prikaz okoline vozila. Drugi SoC se zove FFN (engl. *Front view camera near angle stereoscopic view, Front view camera wide angle, Night vision camera*) i on ima četiri ulaza kamera tipa RDACM23 i koristi se za snimanje stanja ispred vozila. Treći SoC se zove FUS (engl. *FUSion*) i koristi informacije dobivene iz prva dva SoC-a za algoritme koji će odlučivati o točnosti podataka koje dobije. FUS SoC nema video ulaz.

SoC-ovi su međusobno i s periferijom povezani raznim sabirnicama za brzu komunikaciju kao što su *Ethernet*, *USB*, *CAN*, *SPI*, itd. Svaki SoC sadrži deset procesora. Četiri od njih su dvojezgreni procesori opće namjene od kojih dva procesora imaju *ARM Cortex M4*, a dva *ARM Cortex A15* arhitekturu. Dva procesora su *Texas Instruments C66x* DSP arhitekture koji se koriste za digitalnu obradu video signala, a preostale četiri procesorske jedinice koriste se za obradu video signala niskog i srednjeg opterećenja nazvane *EVE* (engl. *Embedded Vision Engine*). Ploča sadrži i *VPE* (engl. *Video Processing Engine*) odjeljak koji se koristi u obradi video sadržaja, kao na primjer za skaliranje veličine slike ili promjena načina kodiranja slike.

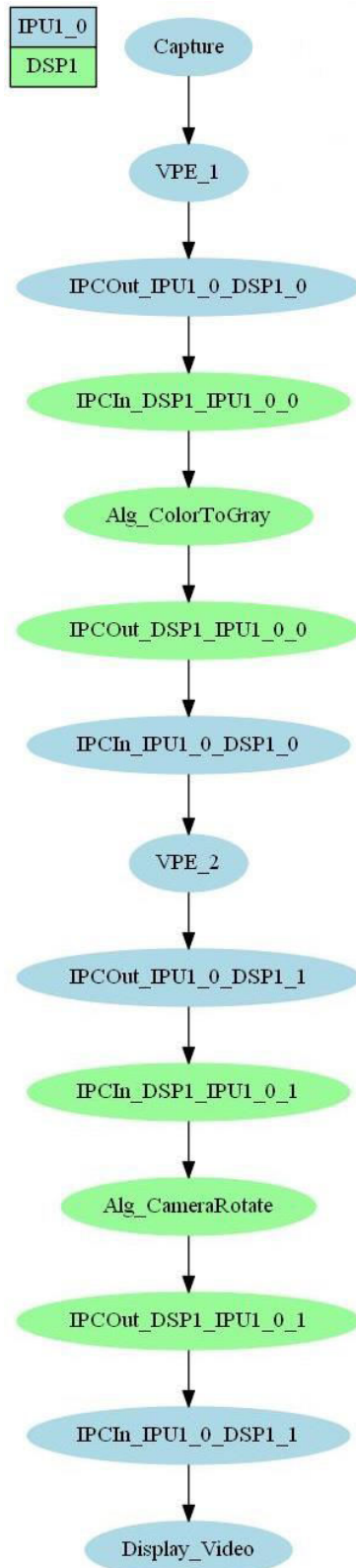
Za komunikaciju ploče s računalom se koristi *UART* (engl. *Universal Asynchronous Receiver/Transmitter*) i *Ethernet* sučelje. *I2C* (engl. *Inter-Integrated Circuit*) sučelje se koristi za komunikaciju s dijelovima zaduženim za deserializaciju podataka pristiglih s kamera. Sustav

se podiže preko *Micro SD* kartice, a HDMI (engl. *High Definition Media Interface*) sučelje se koristi za prikaz slike na monitoru.

Alpha ploča koristi razvojno okruženje zvano *VisionSDK* (engl. *Vision Software Development Kit*) koje je višeprocesorska platforma za razvoj ADAS aplikacija. On sadrži osnovne primjere nekih aplikacija za ADAS sustave. *VisionSDK* radi na principu „veze i lanci“ (engl. *Links and Chains*) gdje su veze predstavljene kao cjelina na koju se algoritmi implementiraju, a poveznice između tih veza su lanci [18].

### **3.3.2. Pokušaj implementacije**

Razvijeno rješenje je bilo potrebno prilagoditi i implementirati na ALPHA ploču. Na slici 3.8. je prikazan korisnički slučaj u kojem je implementiran algoritam. Plavom bojom su označene veze koje se pokreću na IPU\_1 jezgri dok se veze označene zelenom bojom pokreću na DSP1 jezgri procesora.



**Slika 3.8.** Raspored veza i lanaca u korisničkom slučaju

*Capture* veza se koristi za primanje okvira s ulaza ploče, u ovom slučaju je dohvaćala okvire iz dva kanala s dvije različite kamere. *VPE\_1* veza je promijenila način kodiranja elemenata slika iz

*YUV420* u *YUV422*. Taj način kodiranja je potreban za sljedeću vezu koja se zove *Alg\_ColorToGray*. Ta veza je odgovorna za pretvorbu slike u boji u monokromatsku. Veza *IPCOut\_IPU1\_0\_DSP1\_0* prebacuje tok korisničkog slučaja s jezgre procesora IPU1 na DSP1 jezgru, a veza *IPCIn\_DSP1\_IPU1\_0\_0* prima taj tok. *VPE\_2* veza mijenja način kodiranja okvira iz *YUV422* u 8-bitni *bitmap* zapis. U vezu *Alg\_Disparity* se implementirao odabrani algoritam. Sve veze korištene u korisničkom slučaju osim *Alg\_Disparity* su zadano postavljene u *VisionSDK*. Veza *Alg\_Disparity* je lancima povezana s vezom *Display\_Video* koja je odgovorna za prikaz slike preko HDMI-a.

Implementacija nije moguća jer na memorijskoj hrpi (*Heap*) nema dovoljno prostora za spremanje mape dispariteta. Na slici 3.9. je prikazan isječak iz prozora s kojim se vrši komunikacija s ALPHA pločom. Na isječku se vidi kako je na ploči ukupno slobodno 736 kilobajta memorije. Mapa dispariteta koja bi trebala biti pohranjena na taj prostor zauzima 1 bajt za svaki element slike dobiven s kamere rezolucije 1280x1080, odnosno 1350 kilobajta. Mapa dispariteta se sprema kao slika s načinom kodiranja *Bitmap* koji dva elementa mape dispariteta spremi na adresu koja potroši 4 bajta memorije iako nije potrebno za kvalitetan zapis. Zbog takvog načina spremanja se gubi još više raspoložive memorije. Pokušano je promijeniti vezu *VPE\_1* da skaliranjem smanji rezoluciju za pola, ali niti taj pothvat nije pošao za rukom jer još uvijek nije bilo dovoljno memorije. Ploča vjerojatno sprema još neke podatke te je nepoznat stvaran podatak o dostupnoj memoriji.

```
[DSP1 ] 10.993167 s: IPC_OUT_0 : Create in progress !!!
[DSP1 ] 10.993259 s: IPC_OUT_0 : Create Done !!!
[DSP1 ] 11.314737 s: IPC_IN_1 : Create in progress !!!
[DSP1 ] 11.315287 s: IPC_IN_1 : Create Done !!!
[DSP1 ] 11.315470 s: ALGORITHM: Create in progress (algId = 39) !!!
[DSP1 ] 11.315561 s: ALGORITHM: Create Done (algId = 39) !!!
[DSP1 ] 11.315714 s: IPC_OUT_1 : Create in progress !!!
[DSP1 ] 11.315805 s: IPC_OUT_1 : Create Done !!!
[DSP1 ] 12.821905 s: SYSTEM: SH Message Box Msg Pool, Free Msg Count = 102
3
[DSP1 ] 12.821936 s: SYSTEM: Heap = LOCAL_L2 @ 0x00800000, Tot
al size = 227264 B (221 KB), Free size = 227264 B (221 KB)
[DSP1 ] 12.821966 s: SYSTEM: Heap = LOCAL_DDR @ 0x00000000, Tot
al size = 524288 B (512 KB), Free size = 518464 B (506 KB)
[DSP1 ] 12.900933 s:
[DSP1 ] 12.900963 s: ### XDC ASSERT - ERROR CALLBACK START ###
[DSP1 ] 12.900994 s:
[DSP1 ] 12.901085 s: out of memory: handle=0x899d5038, size=230408
[DSP1 ] 12.901116 s:
[DSP1 ] 12.901146 s: ### XDC ASSERT - ERROR CALLBACK END ###
[DSP1 ] 12.901146 s:
[DSP1 ] 12.901299 s: ti.sysbios.heaps.HeapMem: line 221: ti.sysbios.heaps.H
eapMem: line 221: out of memory: handle=0x899d5038, size=230408
```

Slika 3.9. Isječak iz prozora za komunikaciju s pločom



## 4. VERIFIKACIJA PREDLOŽENOG RJEŠENJA

Verifikacija rješenja je obavljena na KITTI setu podataka dostupnog na [19]. Set podataka koji je korišten za verifikaciju se sastoji od slika snimljenih u vožnji. Set podataka se sastoji od 388 uzoraka sa slikom iz lijeve i desne kamere. Rezolucije slika su 1226x370 i 1242x375. Testiranje je provedeno tako što su izračunate mape dispariteta metodama *StereoSGBM* i *StereoBM* iz OpenCV-a. Mapa dispariteta dobivena metodom *StereoSGBM* se koristi kao referentna te se s njom uspoređuju druge dvije metode. Prema izrazu (4 – 1) se svaki element mape dispariteta  $dis_{SGBM}(u, v)$  oduzima od elementa na istoj lokaciji iz mape dispariteta  $dis(u, v)$  dobivene rješenjem opisanom u trećem poglavlju. Također se na isti način za usporedbu izračunavala i razlika između mapa dispariteta dobivenih *StereoSGBM* i *StereoBM* metodama prema izrazu (4 – 2).  $Rez(u, v)$  predstavlja razliku dispariteta predloženog rješenja, odnosno dispariteta dobivenog pomoću *StereoBM* metodom, i dispariteta dobivenog *StereoSGBM* metodom. Razlike su radi lakše analize svrstane po grupama i prikazane u histogramima 4.1. i 4.2.

$$Rez_{Predloženo}(u, v) = dis(u, v) - dis_{StereoSGBM}(u, v) \quad (4 - 1)$$

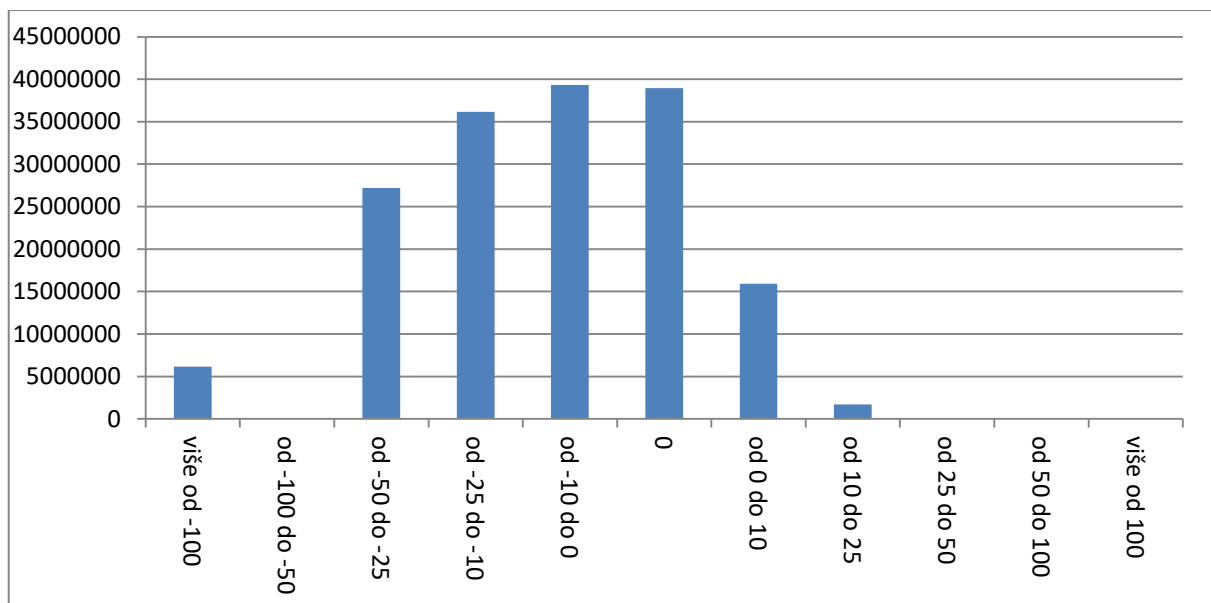
$$Rez_{StereoBM}(u, v) = dis_{StereoBM}(u, v) - dis_{StereoSGBM}(u, v) \quad (4 - 2)$$

Algoritam s lijeve strane mape dispariteta ostavlja prazan prostor jer nije u mogućnosti upariti elemente i izračunati disparitet. Razlog tomu je to što se zbog pomaka kamere dio slike ne vidi na snimci desne kamere. Taj dio nije uzet u obzir prilikom verifikacije rada algoritma. Uz setove podataka na kojima je testiran algoritam nisu navedeni parametri kamere. Stoga se algoritam verificirao na prethodno opisani način koristeći mapu dispariteta što je jednako dobro kao i verifikacija pomoću 3D točaka.

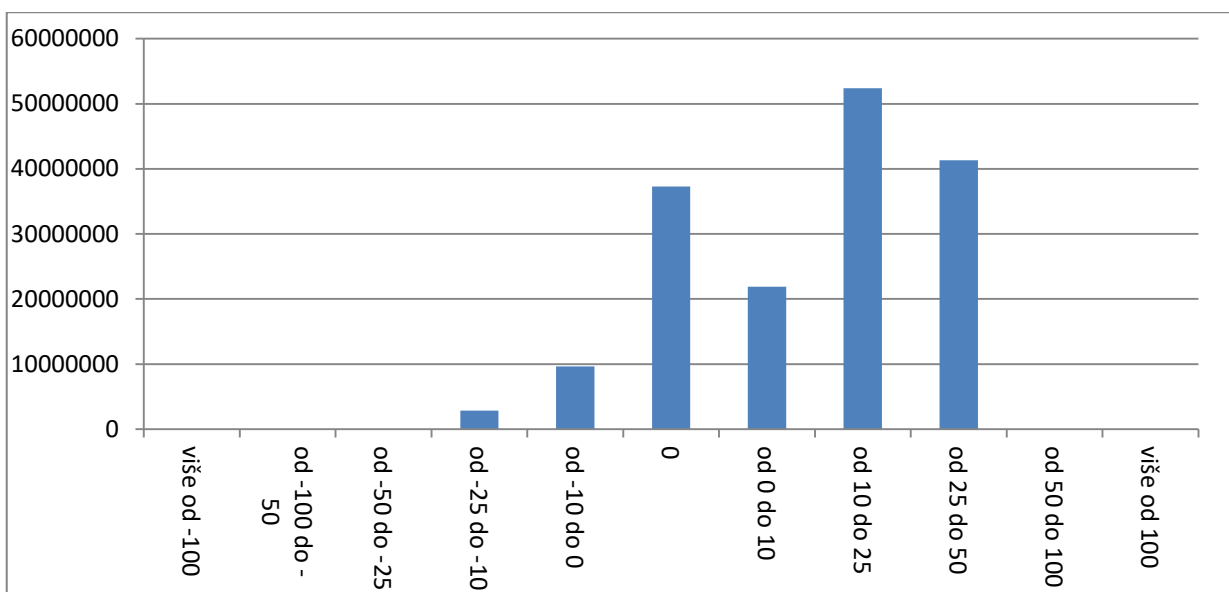
Na histogramu 4.1. su prikazani rezultati verifikacije predloženog rješenja, a na histogramu 4.2. se nalaze rezultati verifikacije *StereoBM* metode. Na apscisama su razlike dispariteta svrstane u grupe dok se na ordinatama histograma prikazuje količina verificiranih elemenata slika koji se nalaze u svakoj grupi. Tako primjerice se iz histograma 4.1. može iščitati kako je točno upareno, odnosno razlika dispariteta je 0, približno 39 miliona elemenata slika od ukupnih 165,398 miliona. Negativne vrijednosti u grupama označavaju da se elementi slika koji predstavljaju objekt nalaze bliže nego što se nalaze na mapi dispariteta dobivenom *StereoSGBM* metodom. Pozitivne vrijednosti govore da su ti objekti, koje dispariteti predstavljaju, udaljeniji na mapi dispariteta dobivenoj *StereoSGBM* metodom. Maksimalna moguća razlika između vrijednosti

elementa na mapama dispariteta je 255 što je najgori slučaj, dok je razlika vrijednosti elemenata mapa 0 točan rezultat.

Predloženo rješenje je točnije od metode *StereoBM* iz OpenCV-a jer se pogreške nalaze u blizini točnog dispariteta dok se većina rezultata na histogramu 4.2. nalazi u opsegu pogrešaka od 10 do 25 i 25 do 50 što je generalno lošiji rezultat zbog većeg odstupanja od točne vrijednosti. Razvijeno rješenje elemente kojima ne može odrediti disparitet označava s 255 i te pogreške spadaju u opseg „više od -100“. Na slikama 4.1. do 4.5 je prikazan primjer testiranja.



**Histogram 4.1.** Prikaz verifikacije razvijenog algoritma i SGBM



**Histogram 4.2.** Prikaz verifikacije algoritama BM i SGBM

Generalno sve lokalne metode spajanja elemenata slika nisu u mogućnosti točno upariti elemente slike koji se nalaze primjerice na bijelom zidu kuće prikazanom na slikama 4.1. i 4.2. jer je zid uniformne boje i algoritam bez globalnih metoda, odnosno korištenja cijele slike nije u mogućnosti odlučiti koji dio zida je par dijelu zida na drugoj slici. Iz sličnih razloga također nije moguće točno upariti elemente slike koji predstavljaju lišće prikazano na desnoj strani na slikama jer je lišće previše slično jedno drugome i algoritam ih ne razlikuje. Isti se problem javlja i kod ceste. Na slikama 4.3. do 4.5. su prikazane mape dispariteta dobivene pojedinim algoritmom. Svjetlo crvena boja označava da je objekt sa slike bliži dok tamnije nijanse crvene označavaju da je objekt udaljeniji. Metoda *StereoBM* elemente slike koje ne može upariti označava s 0 što je doprinijelo boljim rezultatima na histogramu 4.2. jer su vrijednosti dispariteta bliže 0 nego 255.



**Slika 4.1.** *Slike lijeve kamere*



**Slika 4.2.** *Slika desne kamere*



**Slika 4.3.** *Rezultat dobiven pomoću vlastitog rješenja*



**Slika 4.4.** *Rezultat dobiven pomoću StereoSGBM metode*



**Slika 4.5.** *Rezultat dobiven pomoću StereoBM metode*

## 5. ZAKLJUČAK

Cilj rada je bio istražiti algoritam za izradu 3D geometrije na temelju sustava 2D kamera autonomnog vozila te implementirati ga na Alpha ploču. Postoje brojne metode za pronalazak geometrije prostora oko autonomnog vozila te je zadatak bio pronaći najprikladnije rješenje. Predloženo je rješenje koje spaja sve elemente slika. Po primitku para slika iz kamera za blok oko svakog elementa slike iz prve kamere traži se njemu najbliži blok u blizini njegovih koordinata na slici iz druge kamere. Potom se udaljenost među uparenim blokovima pohranjuje za svaki element slike iz lijeve kamere. Ta udaljenost predstavlja disparitet za taj element slike. Mapa dispariteta se uz žarišnu duljinu kamera i udaljenost između kamera koristi za izračun geometrije 3D prostora. Dobivene koordinate se u ADAS-u koriste za određivanje udaljenosti prepreka na cesti koje vozilo mora izbjeći. Potom se planira putanja vozila u odnosu na te prepreke.

Algoritam nailazi na poteškoće ako se u blizini promatranog elementa slike nalaze blokovi izgledom slični traženom bloku. Primjerice, bijeli zid, rub dvorišne ograde i cesta zbog uniformnosti svog izgleda dovode do pogrešne detekcije traženog bloka oko elementa slike. Lokalne metode se ne bi trebale koristiti u ADAS sustavima jer nisu dovoljno pouzdane, a cijena pogreške je veća materijalna šteta ili čak ljudski život. Točnost lokalnih metoda bi se mogla povećati ukoliko se koristi uz detekciju objekata jer bi se tada detektirani objekt uzimao kao blok koji bi se koristio za uspoređivanje. Algoritam nije bilo moguće implementirati na ALPHA ploču jer nije bilo dovoljno memorije za spremanje mape dispariteta.

## SAŽETAK

Cilj ovog rada je izrada algoritma koji će pomoću seta slika primljenih s dvije paralelno postavljene kamere stvoriti geometriju 3D prostora ispred automobila. Algoritam je osmišljen tako da uspoređuje blok oko zadanog elementa slike s blokovima oko elemenata slike koji se nalaze na istoj ordinati kao zadani element slike na drugoj slici. Postupak se ponavlja za svaki element slike na slici snimljenoj s lijevom kamerom koja se uzima da je referentna. S pronađenim parovima elemenata slike se računaju se 3D koordinate pomoću triangulacije kako bi se odredila udaljenost objekata, odnosno potencijalnih prepreka, od vozila. U radu je opisan pokušaj implementacije algoritma na ALPHA ploču koja sadrži 3 SoC-a naziva TDA2xx kojeg je razvila tvrtka Texas Instruments.

Ključne riječi: Alpha ploča, triangulacija, stereo vid, spajanje elemenata slika, TDA2xx

### **Creation of 3D space geometry based on a 2D camera system in autonomous vehicle**

#### **ABSTRACT**

The goal of this paper was to develop the algorithm which will, using a set of shots from two parallel cameras placed on top of a car, create the geometry of 3D space in front of a car. The algorithm is designed to compare a block around the given pixel with the blocks around the pixels that are set on the same ordinate as the given pixel on the second shot. The process is repeated for every pixel on the shot taken with the left camera which is the reference camera. Using the matched pixels the algorithm calculates 3D coordinates using triangulation to determine the distance between the objects, meaning potential obstacles, and the vehicle. Then is described attempt of algorithm implementation on ALPHA board which consists of three SoC named TDA2xx which is developed by the Texas Instruments company.

**Keywords:** Alpha board, triangulation, stereo vision, matching pixels, TDA2xx

## ŽIVOTOPIS

Davor Kelečić

Rođen je 22. veljače 1994. godine u Virovitici, dok mu se prebivalište nalazi u Suhopolju. Završio je osnovnu školu „OŠ Suhopolje“ u Suhopolju. Potom je završio srednju školu „Tehnička škola Virovitica“ 2011./2012. godine sa zanimanjem elektrotehničar. Tema maturalnog rada bila je „Niskofrekventno pojačalo“ koje je bilo potrebno uz teorijsku podlogu i praktično napraviti. Godine 2014./2015. završava preddiplomski studij elektrotehnike na Elektrotehničkom fakultetu u Osijeku. Tema završnog rada je bila „Konfiguriranje virtualnih lokalnih mreža“. Potom upisuje sveučilišni diplomski studij elektrotehnike, smjer komunikacije i informatika na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Te godine postaje stipendist instituta RT-RK u Osijeku što se nastavilo i sljedeće godine. Dobro poznaje engleski jezik. Posjeduje vozačku dozvolu B kategorije.

---

## LITERATURA

- [1] V. Braut, Detekcija objekta u gustoj stereoskopskoj rekonstrukciji, Zagreb, 2014.
- [2] PERPETUAL ENIGMA, Understanding Camera Calibration, dostupno na <https://prateekvjoshi.com/2014/05/31/understanding-camera-calibration/> [2.11.2018]
- [3] P. Marče, Lokalizacija objekta primjenom stereoskopske rekonstrukcije, Zagreb, 2015.
- [4] Depth Biomechanics, dostupno na <http://www.depthbiomechanics.co.uk/?p=102> [2.11.2018]
- [5] C. Harris, M. Stephens, A combined corner and edge detector, Alvey Vision Conference, str. 147 – 151, 1988.
- [6] M. Pul, Usporedba dinamičnih slika zasnovana na izdvajanju i usporedbi ključnih točaka, Osijek, 2016.
- [7] P. L. Rosin, Measuring corner properties, Computer Vision and Image Understanding, 73(2): str. 291 – 307, 1999.
- [8] S. T. Khawase, S. D. Kamble, N. V. Thakur, A. S. Patharkar, An Overview of Block Matching Algorithms for Motion Vector Estimation, Proceedings of the Second International Conference on Research in Intelligent and Computing in Engineering, str. 217 – 222, 2017.
- [9] H. Hirschmuller, Stereo processing by semiglobal matching and mutual information, IEEE Transactions on Pattern Analysis and Machine Intelligence, sv. 30, str. 328 – 341, 2008.
- [10] S. Nedeveschi, R. Danescu, C. Pocol, M. M. Meinecke, Stereo Image Processing for ADAS and Pre-Crash Systems, 2008.
- [11] S. Birchfield, C. Tomasi, Depth discontinuities by pixel-to-pixel stereo. In Proceedings of the Sixth IEEE International Conference on Computer Vision, str. 1073-1080, India, 1998.
- [12] S. K. Gehrig, F. Eberli, T. Meyer, A Real-Time Low-Power Stereo Vision Engine Using Semi-Global Matching, M. Fritz, B. Schiele, J. H. Piater, Computer Vision Systems str. 134 – 143, Belgium, 2009.
- [13] Autonomus Vision Group, dostupno na [http://www.cvlibs.net/datasets/karlsruhe\\_sequences/](http://www.cvlibs.net/datasets/karlsruhe_sequences/) [2.11.2018]



- [14] H. Reese, Updated: Autonomous driving levels 0 to 5: Understanding the differences, dostupno na: <https://www.techrepublic.com/article/autonomous-driving-levels-0-to-5-understanding-the-differences/> [2.11.2018]
- [15] NVIDIA DRIVE, NVIDIA, dostupno na <https://www.nvidia.com/en-us/self-driving-cars/drive-platform/> [2.11.2018]
- [16] RENESAS, dostupno na : <https://www.renesas.com/sg/en/solutions/automotive/adas.html> [2.11.2018]
- [17] Drive Data Platform, Qualcomm, dostupno na <https://www.qualcomm.com/solutions/automotive/drive-data-platform> [2.11.2018]
- [18] N. Perkovic, M. Pranjic, I. Kolak, G. Velikic, System for automotive machine vision, 2017 IEEE 7th International Conference on Consumer Electronics – Berlin, 2017.
- [19] The KITTI Vision Benchmark Suite, dostupno na [http://www.cvlibs.net/datasets/kitti/eval\\_stereo\\_flow.php?benchmark=stereo](http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=stereo) [19.11.2018]