

# Detekcija pješaka upotrebom više različitih kamera smještenih u automobilu

---

Avramović, Nemanja

Master's thesis / Diplomski rad

2020

*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:200:239577>

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

*Download date / Datum preuzimanja:* **2025-04-01**

*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**

**DETEKCIJA PJEŠAKA UPOTREBOM VIŠE  
RAZLIČITIH KAMERA SMJEŠTENIH U  
AUTOMOBILU**

**Diplomski rad**

**Nemanja Avramović**

**Osijek, 2020.**

# SADRŽAJ

1. UVOD.....	1
2. PROBLEM DETEKCIJE PJEŠAKA I PROCJENE NJEGOVE UDALJENOSTI OD VOZILA.....	3
2.1. Detekcija pješaka .....	3
2.2. Proračun udaljenosti detektiranih objekata.....	4
2.3. Postojeća rješenja za detekciju objekata i proračun udaljenosti detektiranih objekata .....	4
3. ALGORITAM ZA ISCRTAVANJE 2D MAPE POLOŽAJA PJEŠAKA U OKOLINI VOZILA ZASNOVAN NA KORIŠTENJU ČETIRIJU KAMERA .....	19
3.1. Opis VisionSDK razvojnog okruženja i ADAS platforme na kojoj je rješenje implementirano	20
3.2. Programsko rješenje za iscrtavanje 2D mape položaja pješaka u okolini vozila koristeći četiri kamere u vozilu .....	23
3.2.1. Algoritam za detekciju pješaka .....	27
3.2.2. Kalibracija širokokutnih kamera .....	29
3.2.3. Algoritam za procjenu udaljenosti detektiranih pješaka metodom 1 .....	30
3.2.4. Algoritam za procjenu udaljenosti pješaka metodom 2 .....	33
3.2.5. Algoritam za iscrtavanje 2D mape položaja pješaka u okolini vozila .....	34
3.3. Programsko rješenje za iscrtavanje 2D mape položaja pješaka u okolini vozila koristeći dvije kamere.....	40
3.4. Programsko rješenje za procjenu udaljenosti pješaka koristeći jednu kameru na vozilu .....	44
3.5. Način pokretanja vlastitog rješenja za iscrtavanje 2D mape položaja pješaka u okolini vozila .....	46
4. TESTIRANJE RADA PREDLOŽENOG RJEŠENJA ZA ISCRTAVANJE 2D MAPE POLOŽAJA PJEŠAKA U OKOLINI VOZILA KORISTEĆI VIŠE KAMERA .....	51
4.1. Testiranje performansi algoritma za detekciju pješaka .....	51
4.2. Testiranje ispravnosti rada algoritma za procjenu udaljenosti.....	55
4.2.1. Procjena udaljenosti detektiranih pješaka unutar okvira rezolucije 640x360 elemenata slike .....	56
4.2.1. Procjena udaljenosti detektiranih pješaka unutar okvira rezolucije 960x540 elemenata slike .....	58
4.2.2. Procjena udaljenosti detektiranih pješaka unutar okvira rezolucije 1280x720 elemenata slike .....	62
4.3. Testiranje ispravnosti rada algoritma za iscrtavanje 2D mape položaja pješaka u okolini vozila koristeći više kamera .....	66

4.4. Osvrt na dobivene rezultate .....	69
5. ZAKLJUČAK .....	71
LITERATURA.....	73
SAŽETAK .....	75
ABSTRACT .....	76
ŽIVOTOPIS.....	77
PRILOZI .....	78

# 1. UVOD

Trenutno najbrže rastuća industrija u svijetu je upravo automobilska industrija. Na cesti se susreće sve više vozila koji imaju ugrađene sustave za obavljanje složenih operacija tijekom vožnje. Oni konstantno prate okolinu oko vozila i smanjuju rizik od prometnih nesreća. Autonomni sustavi, koji se ugrađuju u vozila, nazivaju se ADAS (engl. *Advanced Driver Assistance Systems*). Glavni uzročnik prometnih nesreća je čovjek [1]. ADAS sve više zamjenjuju čovjeka u mnogim radnjama, kao što su samostalno parkiranje vozila, vožnja bez potrebe za držanjem ruku na upravljaču (u određenim prometnim scenarijima, ne još uvijek u svim), detekcija prometnih znakova, pješaka i drugih vozila, itd. ADAS vozilima omogućava visoku razinu autonomije. Unutar istog sinkronizirano komunicira velik broj senzora s procesorima. Senzori šalju podatke procesorima, na kojima se, na osnovu obrađenih podataka sa senzora, izvršavaju određeni algoritmi. Neki od senzora su LiDAR, radar, kamere, ultrazvučni senzori, itd. Složeni algoritmi obrađuju veliku količinu podataka primljenu s različitih senzora, kako bi mogli poslati poruku aktuatorima, koju radnju je potrebno provesti za obrađene ulazne podatke. Trenutno je automobilska industrija dostigla visoku razinu autonomije nivoa 3 [2], kod kojeg vozač mora biti prisutan za upravljačem prilikom vožnje i spreman da u bilo kojem trenutku intervenira, ako neki od sustava zakaže [3]. U skorijoj budućnosti se očekuje potpuno autonomna vožnja, u kojoj neće biti nužna prisutnost vozača. Samim time će ljudski faktor, kao glavni uzročnik prometnih nesreća, biti u velikoj mjeri, ili čak u potpunosti uklonjen.

Kao što je ranije navedeno, jedna od radnji koju obavljaju ADAS, je detekcija pješaka koji se kreću oko vozila. Ova funkcionalnost modernih vozila je potrebna, jer često u velikim gradovima u gužvi, čovjek prilikom vožnje ne može pratiti cjelokupnu okolinu vozila. Postoji mogućnost da se neki pješak približava cesti, bez da pazi da li postoji opasnosti na cesti. U tom trenutku može se dogoditi da vozač takvu situaciju ne uoči. Zato je potrebno da moderna vozila imaju implementiranu funkcionalnost automatske detekcije pješaka, koji se nalaze u okolini vozila, kako bi u slučaju opasne situacije mogli pravovremeno obavijesti vozača ili samostalno reagirati poduzimanjem ispravne radnje. Algoritam za detekciju pješaka zasniva se na sinkroniziranoj obradi podataka s više različitih kamera smještenih u vozilu. Namjena algoritma je da sustav konstantno nadgleda okolinu vozila, te da informacije o detektiranim pješacima dostavlja vozaču u realnom vremenu. Prvi od problema s kojim se susreće algoritam je sama detekcija pješaka. Detekcija pješaka se zasniva na prepoznavanju po određenim značajkama, svojstvenim za ljude. Nakon ispravne detekcije, drugi problem je točna procjena udaljenosti

pješaka. Ako u vozilu postoje dodatni senzori za procjenu udaljenosti, kao što su laseri za precizno određivanje udaljenosti, ili ako se koristi princip stereo vizije (odnosno sustav dviju kamera s poznatim parametrima, koje promatraju istu scenu, s poznatim položajem jedne u odnosu na drugu), točnost procijenjene udaljenosti će biti poprilično visoka. No to s druge strane znači potrebu za obradom veće količine podataka i veće zahtjeve za računalnom moći procesora. Međutim, ako se udaljenost detektiranih pješaka procjenjuje na osnovu samo jedne kamere, točnost procjene je puno izazovniji zadatak. Pozitivna strana je da se smanjuju zahtjevi za računalnom moći, što je izrazito bitno zbog velike količine ADAS, koji trebaju istovremeno funkcionirati. S druge strane, znatno je teže postići zadovoljavajuću točnost.

Zadatak diplomskog rada je proučiti načine detekcije pješaka i procjene njihove udaljenosti od vozila te osmisliti vlastito rješenje za proračun udaljenosti i iscrtavanje položaja detektiranih pješaka na 2D mapu. Kamere bi trebale biti postavljene na vozilo tako da svaka snima prostor s jedne od četiriju strana vozila (prednja, stražnja, lijeva i desna). S obzirom na to da je potrebno da za svaki par susjednih kamera postoji dio prostora, kojeg obje kamere snimaju istovremeno, u ovakvim se sustavima koriste širokokutne kamere. Proračun udaljenosti zasnivao bi se samo na upotrebi matematičkih proračuna i jedne kamere. Nakon što bi se proračunala udaljenost pješaka, trebalo bi se izvršiti iscrtavanje položaja pješaka na 2D mapu, koja bi predstavljala pogled na vozilo odozgor. Na kraju bi se rješenje trebalo testirati na određenom broju snimljenih videozapisa, kako bi se ispitali točnost i preciznost istog.

U drugom poglavlju opisana su postojeća rješenja za detekciju objekata te za proračun udaljenosti istih. Treće poglavlje opisuje proces izrade vlastitog rješenja za detekciju pješaka oko vozila. U četvrtom poglavlju opisan je način provođenja testiranja algoritma te su dani rezultati testiranja. Posljednje poglavlje sadrži zaključke rada.

## 2. PROBLEM DETEKCIJE PJEŠAKA I PROCJENE NJEGOVE UDALJENOSTI OD VOZILA

Djelomična, a potom i potpuna, autonomna vožnja jedan je od ciljeva kojemu trenutno stremi automobilska industrija. Neki od problema i zadataka koji trebaju biti riješeni, kako bi autonomna vožnja bila omogućena, su detekcija pješaka i procjena njihove udaljenosti od vozila.

### 2.1. Detekcija pješaka

Detekcija objekata predstavlja složenu računalnu tehniku. Prema [4], algoritmi za detekciju objekata mogu se podijeliti u dvije kategorije:

1. Detekcija objekata zasnovana na klasičnim tehnikama računalnog vida, koje koriste razne algoritme za izdvajanje značajki objekata iz slike, nakon čega se značajke referentnih objekata pohranjuju. Kada se na testnoj slici pomoću istih algoritama detektiraju značajke, pokušavaju se upariti značajke referentnog objekta i objekta na testnoj slici. Tu spadaju algoritmi kao što su SIFT (engl. *Scale Invariant Feature Transform*), SURF (engl. *Speeded Up Robust Features*), ORB (engl. *Oriented FAST and rotated BRIEF*), FAST (engl. *Features from Accelerated Segment Test*), Houghova transformacija itd..
2. Detekcija objekata zasnovana na strojnom učenju, koje se zasniva na dugotrajnom treniranju algoritma za detekciju. Ako govorimo o detekciji objekata u slikama, ovi algoritmi su sposobni detektirati objekte bez dodatnog definiranja značajki objekta. Iako nije potrebno definirati značajke objekata, modele strojnog učenja je neophodno dugotrajno trenirati. Treniranje podrazumijeva davanje nekoliko tisuća slika, često i desetina tisuća slika, jednog objekta mreži, koji treniranjem sa što više različitih slika jednog objekta, postiže visok stupanj točnosti detekcije određenog objekta u novim slikama koje nisu bile korištene pri treniranju. Tu spadaju metode za detekciju objekata primjenom algoritama Viola – Jones i YOLO (engl. *You Only Look Once*), kao i druge metode koje primjenjuju konvolucijske neuronske mreže (engl. *Convolutional Neural Networks – CNNs*), konvolucijske neuronske mreže zasnovane na regiji (engl. *Region – based Convolutional Neural Networks – R – CNNs*) itd.

## **2.2. Proračun udaljenosti detektiranih objekata**

Nakon uspješne detekcije objekata, javlja se problem procjene udaljenosti istih od kamere, tj. od vozila. Prilikom vožnje, neophodno je da vozilo zna koliko je detektirani objekt udaljen u slučaju da vozač ne primijeti opasnost na cesti. Kod ADAS, procjena udaljenosti može se temeljiti na korištenju dodatnih laserskih senzora za točno mjerenje udaljenosti detektiranog objekta, kao što je LiDAR senzor. Jedna od metoda za procjenu udaljenosti je i korištenje stereo vizije, koja se zasniva na procjeni udaljenosti objekata pomoću dviju kamera tako da se dobiju dva pogleda na istu scenu, na osnovu kojih se određuje dubina pojedinog elementa scene. Točnost procjene udaljenosti objekata pomoću laserskih senzora i stereo vizije je poprilično visoka. Međutim cijena koštanja LiDAR-a i stereo sustava kamera nije još na razini prihvatljivoj za sve klase vozila (pogotovo nižu i srednju). Dodatno, obrada velike količine podataka zahtjeva i veću računalnu moć, a samim time i veću cijenu koštanja.

Nešto veći problem javlja se onda kada je udaljenost detektiranog objekta potrebno procijeniti na osnovu samo jedne kamere. S obzirom na to da ova metoda najsklonija davanju pogrešnih rezultata pri procjeni udaljenosti iz više razloga, algoritamska rješenja za procjenu udaljenosti s jednom kamerom trebaju biti što kvalitetnije razvijena. Postoji više načina na osnovu kojih se može proračunati udaljenost detektiranog objekta pomoću jedne kamere, od kojih će dva biti detaljnije opisana u trećem poglavlju.

## **2.3. Postojeća rješenja za detekciju objekata i proračun udaljenosti detektiranih objekata**

U ovom potpoglavlju dan je kratki pregled postojećih radova koji se bave rješavanjem pojedinačnih problema ovog diplomskog rada. Analizirani su tako radovi za detekciju objekata općenito, za detekciju pješaka te za procjenu udaljenosti detektiranih objekata od kamere.

U radu [5] uspoređuju se dvije metode detekcije i praćenja objekata zasnovane na klasičnim tehnikama računalnog vida, koje se zasnivaju na izdvajanju značajki objekata iz slike. Te dvije metode su SIFT (engl. *Scale Invariant Feature Transform*) i SURF (engl. *Speed Up Robust Features*). Kako bi se izvršila detekcija objekata, prvo je potrebno izvući značajke objekta. Prilikom izračunavanja povezanosti između dviju slika za praćenje objekata koji su u pokretu, potrebno je međusobno povezati značajke objekata s obju slika. SIFT i SURF metode služe za izdvajanje značajnih točaka objekta, nakon čega se može detektirati određeni objekt čak i ako je objekt skriven u sjeni ili iza nekog drugog objekta.



SIFT je algoritam [6], koji se koristi u području računalnog vida. Služi za pronalaženje lokalnih značajki slike. Robustan je na promjenu kuta gledanja, svjetline i na skrivenost objekta u sjeni ili iza nekog drugog objekta. Generira velik broj značajki, što je od izuzetne važnosti za uspješnu detekciju objekta. Značajke se prvo izdvajaju iz referentnih slika, koje se potom spremaju u bazu podataka. Nakon što su značajke pohranjene u bazu podataka, prepoznavanje objekata na testnoj slici se izvršava tako da se svaka značajka testne slike uspoređuje sa značajkama iz baze podataka proračunom Euklidove udaljenosti vektora značajki. Sastoji se od 4 faze za izdvajanje značajki:

1. **Detekcija ekstrema u prostoru različitih veličina s istim omjerom visine i širine** – pretraživanje po slikama različitih veličina istog omjera širine i visine. Učinkovito se provodi korištenjem Gaussove funkcije za identificiranje potencijalnih točaka značajki, koje su nepromjenjive prema omjeru i orijentaciji;
2. **Lokalizacija ključnih točki** – na svakoj lokaciji značajke korišten je detaljan model za određivanje omjera i položaja na slici. Ključne točke su odabrane na osnovu mjera njihove stabilnosti;
3. **Dodjela orijentacije** – jedna ili više orijentacija dodijeljene su lokaciji svake pojedine značajke na osnovu smjerova lokalnih gradijenata slike. Sve buduće operacije izvode se na slikovnim podacima, koji su transformirani u odnosu na dodijeljenu orijentaciju, omjer i lokaciju za svaku značajku, pružajući nepromjenjivost na te transformacije;
4. **Deskriptor ključnih točaka** – lokalni gradijenti slike mjere se na odabranoj skali u regiji oko svake značajke. Transformirani su u prikaz koji omogućava značajne razine lokalnog izobličenja i promjene osvjetljenja.

SURF algoritam [7] radi na istim principima kao SIFT te se sastoji od istih faza. Razlika je u detaljima, koji karakteriziraju svaku od faza. Brzina obrade, poput izračuna značajki i podudaranja, veća je kod SURF algoritma., čime je i robusnost na transformacije slike veća. SURF detektor koristi aproksimaciju *Hessianove* matrice. Također se koriste i integralne slike, koje služe za povećanje performansi i smanjivanje vremena izračuna.

Na slici 2.1 [5] se može vidjeti rezultat pronalaska značajki primjenom pojedinog algoritma.



a)

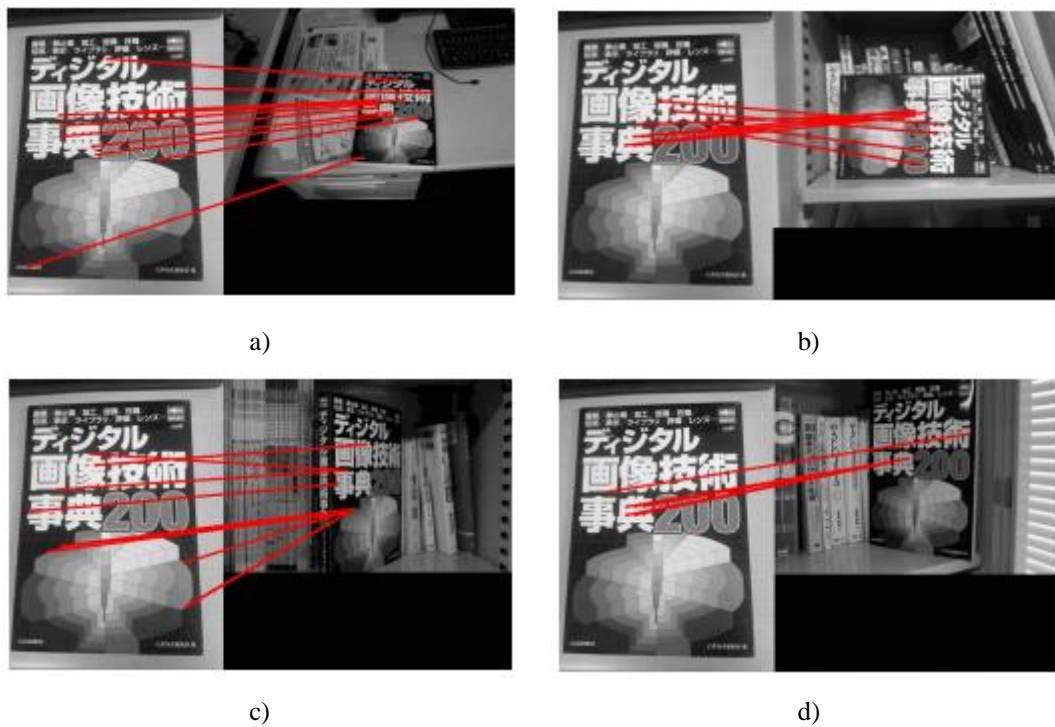


b)

Sl. 2.1. Rezultat pronalaska značajki za (a) SIFT i (b) SURF algoritam u [5]

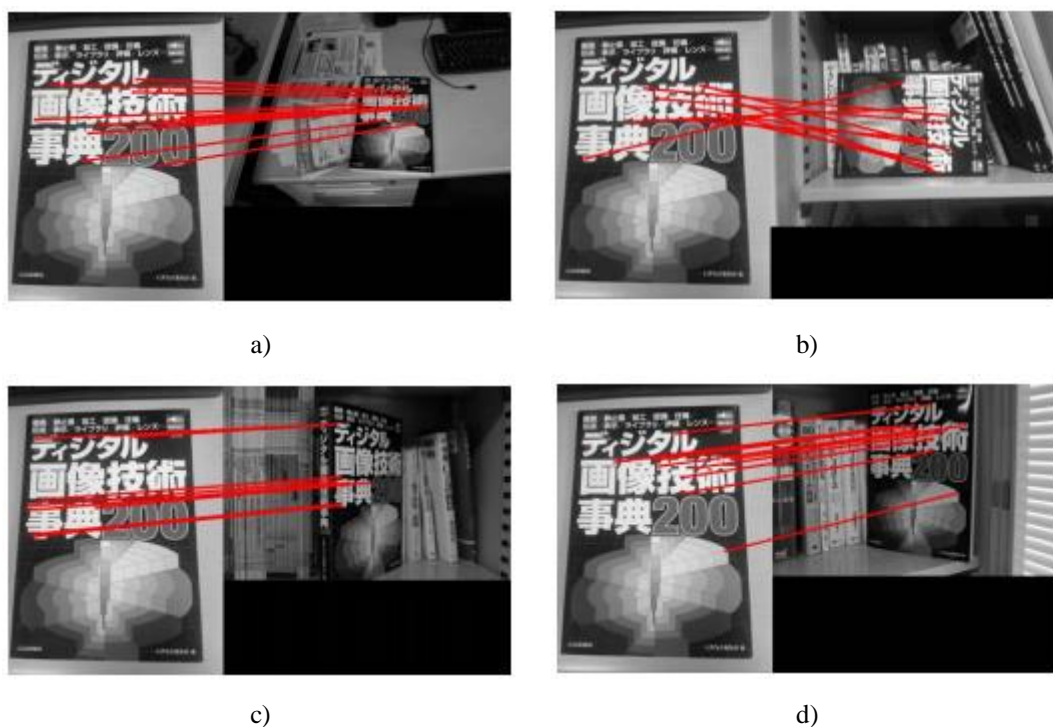
Dio sustava za detekciju i praćenje objekata u [5] radi na Apple Mac pro računalu uz korištenje OpenCV biblioteka. Sustav se također sastoji od senzora i bežične kamere, koji su povezani s malim Raspberry Pi računalom, na čijem se Linux sustavu Raspbian pokreće sustav za nadzor. Eksperimenti su izvedeni u zatvorenom okruženju. Slike iz testnog skupa uspoređene su sa slikama iz referentnog skupa podudaranjem odgovarajućih značajki. Objekt na referentnoj slici s najvećim brojem podudaranja u odnosu na objekte na testnoj slici odabran je kao prepoznati objekt. Testiranje ova dva algoritma sprovedeno je tako što su se testne slike, u odnosu na referentnu, razlikovale na četiri načina. Na prikazu prve testne slike nalazi se objekt koji je udaljen od kamere u odnosu na isti objekt na referentnoj slici, na kojoj objekt zauzima gotovo čitavu površinu slike. Druga testna slika prikazuje objekt koji je rotiran za  $90^\circ$  i udaljeniji je od kamere u odnosu na objekt na referentnoj slici. Treća i četvrta testna slika prikazuju objekt koji se nalazi pod određenim kutem, nije precizirano koliki je taj kut, te je udaljeniji od kamere nego objekt prikazan na referentnoj slici.

Na slici 2.2 [5] mogu se vidjeti rezultati podudaranja značajki objekta na testnim slikama sa značajkama objekta na referentnoj slici kod SIFT algoritma.



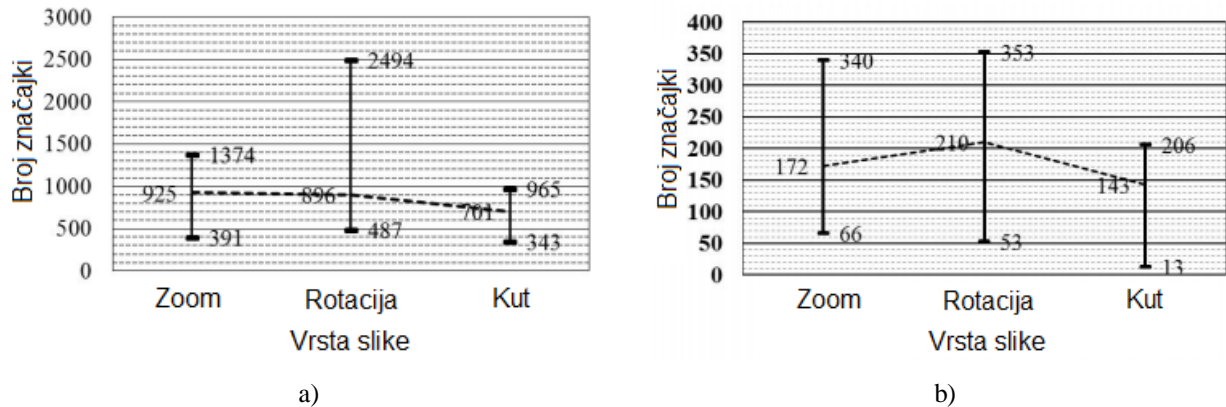
Sl. 2.2. Rezultati podudaranja značajki objekta na (a) prvoj, (b) drugoj, (c) trećoj i (d) četvrtoj testnoj slici sa značajkama objekta na referentnoj slici kod SIFT algoritma u [5]

Na slici 2.3 [5] mogu se vidjeti rezultati podudaranja značajki objekta na testnim slikama sa značajkama objekta na referentnoj slici kod SURF algoritma.



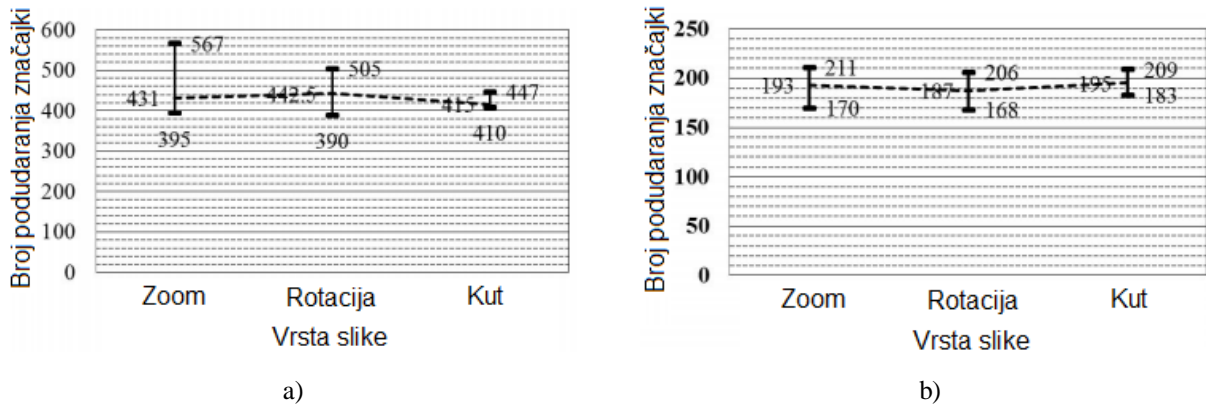
Sl. 2.3. Rezultati podudaranja značajki objekta na (a) prvoj, (b) drugoj, (c) trećoj i (d) četvrtoj testnoj slici sa značajkama objekta na referentnoj slici kod SURF algoritma u [5]

Testiranja su sprovedena na različitim slikama primjenjujući oba algoritma. Na slici 2.4 [5] mogu se vidjeti rezultati broja značajki, koje su pronađene na trima vrstama testnih slika kod SIFT algoritma.



Sl. 2.4. Rezultati broja značajki koje su pronađene kod (a) SIFT i (b) SURF algoritma u [5]

Rezultati su izraženi s tri broja po vrsti testnih slika. Ta tri broja predstavljaju maksimum, minimum i medijan pronađenih značajki određene vrste testnih slika. Iz dobivenih rezultata može se vidjeti da je broj pronađenih značajki veći kod SIFT algoritma. Na idućoj 2.5 [5] slici mogu se vidjeti rezultati podudaranja značajki izraženi također s maksimumom, medijanom i minimumom.

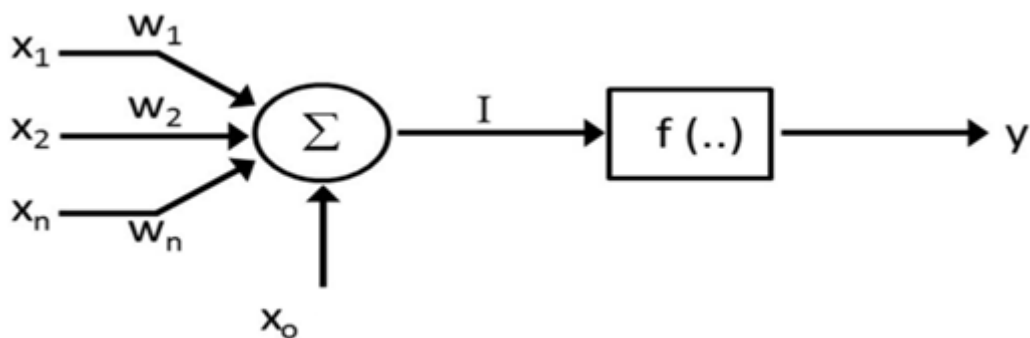


Sl. 2.5. Rezultati broja podudaranja značajki koje su pronađene kod (a) SIFT i (b) SURF algoritma u [5]

Iz dobivenih rezultata može se vidjeti da je točnost podudaranja značajki veća kod SURF algoritma. U ovom radu predložene su dvije metode za praćenje i detekciju objekata zasnovane na izdvajanju značajki iz objekata na slikama. Iako oba algoritma rade na istom principu, SURF algoritam je superiorniji, zbog čega dovodi do točnijih rezultata i boljih performansi, odnosno bržeg pronalaska značajki i izvršavanja procesa podudaranja značajki između testnih i referentnih slika. Brže i suvremenije metode za detekciju objekata od metoda spomenutih u ovom radu, bit će predstavljene u nekim od idućih radova, a to su metode zasnovane na strojnom učenju.

U radu [8] uspoređuju se dvije metode za detekciju objekata. Jedna metoda za detekciju pokretnih objekata koristi umjetnu neuronsku mrežu (engl. *artificial neural network* – ANN), dok se druga zasniva na Viola – Jones algoritmu [9]. Oba algoritma pripadaju onima koji su zasnovani na strojnom učenju.

ANN je alat za obradu skupa informacija, koji predstavlja složen sustav sličan ljudskom mozgu. ANN se sastoji od velikog broja neurona koji su obično organizirani u slojeve da tvore mrežu. Neuroni su povezani tako da omogućuju pohranjivanje podataka i modeliranje ulaza i izlaza procesa. Ključni element ovog modela je nova struktura sustava za obradu informacija. Sastoji se od triju glavnih komponenata, a to su neuroni, slojevi i aktivacijske funkcije. Svaki neuron ima ulaznu vrijednost ( $x_j$ ), težine ( $w_j$ ) i vrijednost pristranosti (engl. *bias* –  $x_0$ ). Izlaz svakog neurona dobiva se kao suma umnožaka odgovarajućih težina i ulaznih vrijednosti ( $w_j * x_j$ ) uvećana za vrijednost pristranosti ( $x_0$ ) i propuštena kroz aktivacijsku funkciju. Slika 2.6 [8] prikazuje jedan neuron s više ulaza i jednim izlazom.

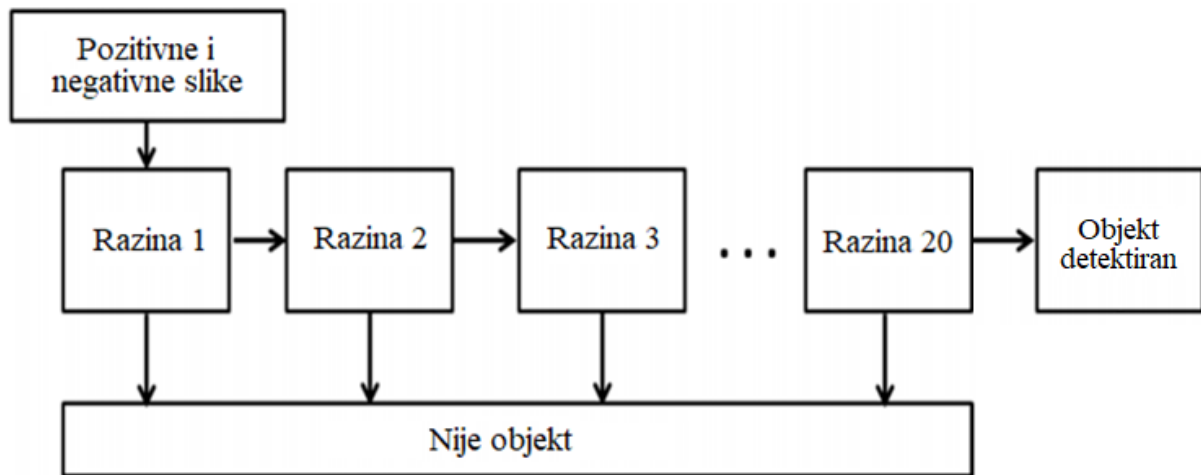


Sl. 2.6. Prikaz jednog neurona s više ulaza i jednim izlazom [8]

Ideja ove metode je da se svaki ulaz dostavi na ulaz neurona, prođe kroz niz iteracija i naposljetku da na izlazu rezultat. Zato se detekcija i klasifikacija objekta odvijaju istovremeno. Neuronska mreža može sadržavati sljedeće tipove slojeva: ulazni, skriveni i izlazni sloj. U ulaznom sloju podaci se isporučuju mreži, dok se u skrivenom sloju isti obrađuju. U izlaznom sloju se nalaze rezultati. Prijenosne funkcije se nalaze između slojeva te služe za postizanje željenog izlaza. Prilikom treniranja ANN u radu [8] su dodijeljena dva ulaza u obliku prepoznatljivih karakteristika za objekte, koji se nalaze na slici.

Viola – Jones algoritam predstavlja metodu za brzo otkrivanje ljudskih lica na slici. No kako, osim kaskade pojačanih Haarovih karakteristika, koristi i algoritme strojnog učenja, može se koristiti i za pronalaženje drugih objekata na slici. Koristi Haarove karakteristike, AdaBoost

algoritam i integralnu sliku. Za treniranje Viola – Jones algoritma koriste se pozitivne i negativne slike. Na pozitivnim slikama se nalaze objekti od interesa, dok se na negativnim ne nalaze. Za brže računanje Haarovih karakteristika se koristi integralna slika. AdaBoost algoritam računa jaki binarni klasifikator, koji predstavlja linearnu kombinaciju više slabih klasifikatora. Viola – Jones koristi na kraju kaskadu jakih klasifikatora, koja je sastavljena od više razina, od kojih svaka predstavlja jedan jaki klasifikator, koji je sastavljan od više slabih klasifikatora. Kaskada se koristi radi dodatnog smanjenja vremena računanja te ubrzanja procesa detekcije. U prvim razinama kaskade, dijelovi slike koji ne sadrže objekt od interesa, označuju se kao negativni i odbacuju se. Dio slike koji se označi kao pozitivan, prelazi na iduću razinu i tek kad prođe sve razine, taj dio slike se klasificira kao objekt od interesa. Slika 2.7 prikazuje dijagram toka rada kaskadne arhitekture [8].



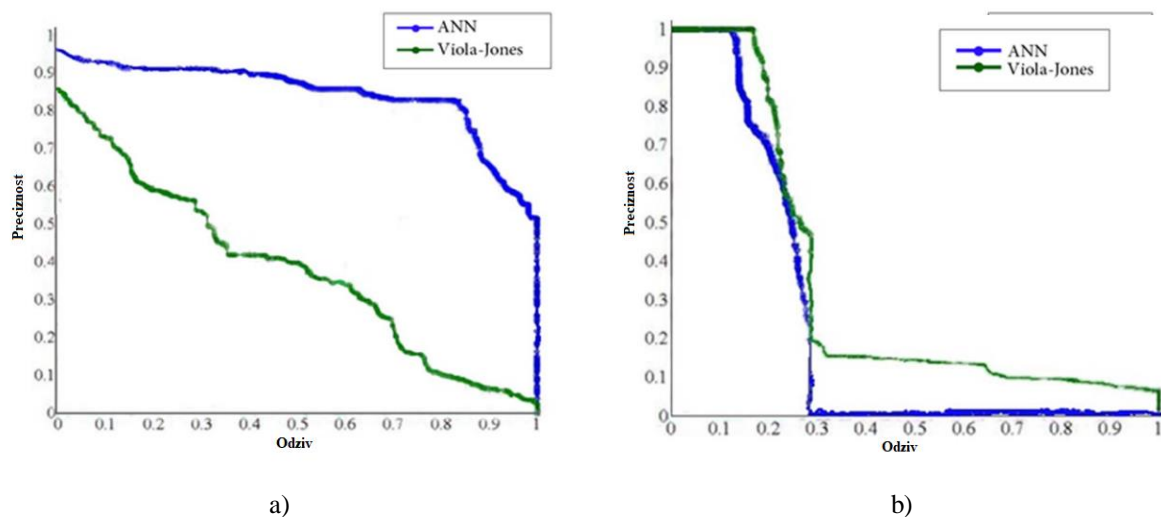
Sl. 2.7. Dijagram toka rada kaskadne arhitekture u [8]

Testiranje ovih dvaju algoritma sprovedeno je tako da su se računali preciznost i odziv rješenja. Preciznost i odziv se zasnivaju na mjernim pokazateljima istinito pozitivne (engl. *true positive* – TP), lažno pozitivne (engl. *false positive* – FP) i lažno negativne (engl. *false negative* – FN) detekcije. Preciznost i odziv definirani su jednadžbama 2.1 i 2.2.

$$Preciznost = \frac{|TP|}{|TP| + |FP|} \quad (2.1.)$$

$$Odziv = \frac{|TP|}{|TP| + |FN|} \quad (2.2.)$$

Preciznost predstavlja omjer broja ispravno detektiranih objekata u slici i broja svih detektiranih objekata u slici (događa se da se prilikom detekcije pješaka npr. detektira drvo). Odziv predstavlja omjer broja ispravno detektiranih objekata u slici i broja objekata koji trebaju biti detektirani. Oba rješenja su testirana pomoću dvaju videa koji prikazuju snimke s ulice. Prvi video prikazuje ulični promet, dok se na drugom nalaze statični objekti. Kod videa s uličnim prometom, ANN se pokazala puno boljom od Viola – Jones algoritma, dok kod videa sa statičnim objektima Viola – Jones algoritam radi jako dobro, jer za svaku razinu postoje klasifikatori za pojedine objekte, koji se redom izvršavaju u svakom trenutku, čime povećava performanse detekcije uz značajno smanjenje vremena računanja. Na slici 2.8 [8] mogu se vidjeti grafovi preciznosti i odziva za oba videa.

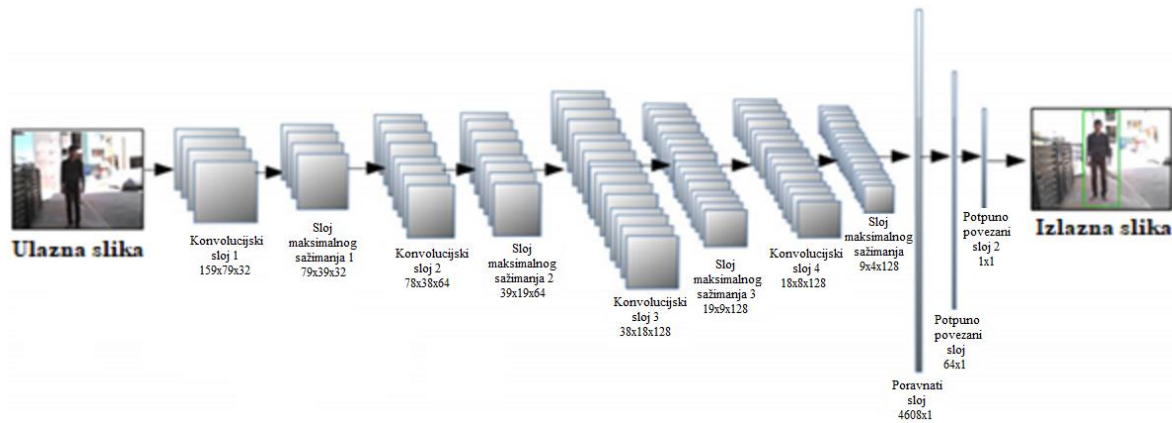


Sl. 2.8. Grafovi preciznosti i odziva za video (a) uličnog prometa i (b) statičnih objekata u ulici u [8]

ANN ima bolje performanse u odnosu na Viola – Jones algoritam prilikom detekcije i klasifikacije više različitih objekata u pokretnom videu, dok Viola – Jones ima bolje performanse prilikom detekcije i klasifikacije statičnih objekata. Kako bi algoritmi što bolje radili, potrebno im je isporučivati što više pozitivnih i negativnih primjera objekta prilikom treninga (kod Viola – Jones algoritma pozitivnih slika objekta, a što više značajki pojedine klase objekta kod ANN).

Autori rada [10] bave se detekcijom pješaka tehnikama dubokog učenja. Sustav za detekciju pješaka u stvarnom vremenu, kao pomoć za autonomna vozila, dizajniran je koristeći CNNs. CNN, koja je razvijena u ovom znanstvenom radu, sastoji se od niza slojeva.

Slojevi CNN mogu se vidjeti na slici 2.9 [10].



Sl. 2.9. Konvolucijska neuronska mreža korištena u [10] za detekciju pješaka

Sastoji se od četiri konvolucijska sloja i dva potpuno povezana sloja. Na kraju, klasa za koju se na izlazu dobije najveći broj (izlaz) smatra se klasom predviđenom za ulaz.

Testiranja točnosti algoritma sprovedena su korištenjem INRIA [11], PETA – CUHK [12] baze podataka slika pješaka i video signala snimanih u stvarnom vremenu. Od ukupnog broja slika iz pojedine baze i video signala snimanih u stvarnom vremenu, 80% korišteno je za treniranje CNN, dok je 20% korišteno za testiranje. U tablici 2.1 [10] može se vidjeti odnos broja testnih slika i slika za treniranje.

Tablica 2.1. Broj trening slika i slika za testiranje iz korištenih skupova podataka [10]

	INRIA baza	PETA – CUHK baza	Video signali snimani u stvarnom vremenu
Broj trening slika	1948	5000	2500
Broj testnih slika	488	1000	500
<b>Ukupan broj slika</b>	<b>2436</b>	<b>6000</b>	<b>3000</b>

Ukupno vrijeme potrebno da se pješak detektira na slici pohranjenoj na prijenosnom računaru na kojem su sprovedena testiranja je 49.99 ms, dok je vrijeme da se pješak detektira iz pojedinog okvira videa manje i iznosi 46.01 ms. Ova vremena ovise o moći pojedinog računala, ali se iz ovoga može zaključiti da se ukupna detekcija s učitavanjem i klasifikacijom izvršava brže prilikom učitavanja slika iz stvarno-vremenskog videa jer računaru treba manje vremena da uhvati



okvir iz videa nego da učita sliku iz datoteke. U tablici 2.2 [10] se mogu vidjeti potrebna vremena za učitavanje slika iz različitih izvora i za klasifikaciju pomoću CNN.

Tablica 2.2. Potrebna vremena za učitavanje slika iz različitih vrsta izvora i klasifikaciju [10]

Funkcije	Vrijeme izvršavanja
Baza podataka	
Čitanje iz datoteke	24.50ms
CNN klasifikacija	25.49ms
<b>Ukupno</b>	<b>49.99ms</b>
Starno-vremenski video	
Uhvaćeni okvir	19.27ms
CNN klasifikacija	26.74ms
<b>Ukupno</b>	<b>46.01ms</b>

Postignuta točnost detekcije može se vidjeti u tablici 2.3 [10], s visokim postotkom detekcije pješaka od 96.73% do 100%.

Tablica 2.3. Točnost detekcije pješaka ovisno o ulaznoj bazi podataka slika [10]

Baza slika	Broj slika	Broj pješaka u slikama	Broj detektiranih pješaka	Postotak točnosti detekcije
INRIA (cijela slika)	288	521	504	96.3%
INRIA (testne slike)	1126	1126	1108	98.40%
PETA – CUHK	400	400	400	100%
Stvarnovremenski ulazi	100	160	157	93.13%

Točnost detekcije se zasniva na mjernim pokazateljima istinito pozitivne (engl. *true positive* – TP), lažno pozitivne (engl. *false positive* – FP), lažno negativne (engl. *false negative* – FN) i istinito negativne (engl. *true negative* – TN) detekcije. Definirana je jednadžbom 2.3.

$$Točnost = \frac{|TP| + |TN|}{|TP| + |FP| + |FN| + |TN|} \quad (2.3.)$$

Iz prethodno navedene formule, točnost je udio točno klasificiranih primjera u skupu svih primjera.

Sama implementacija CNN predstavlja dugotrajan i težak posao, ali korištenjem istih može se postići visoka točnost sustava za detekciju željenih objekata. Točnost detekcije se dodatno može povećati tako da se poveća broj trening slika, jer što više ima različitih slika pješaka uhvaćenih s

različitih strana u različitim položajima, to je algoritam bolje istreniran i spremniji za što ispravniju klasifikaciju u slučaju novih slika.

U radu [13] predstavljen je način određivanja udaljenosti vozila od kamere upotrebom jedne kamere. Objašnjena je metoda koja koristi CNN za detekciju i lociranje vozila, nakon čega se proračunava udaljenost vozila prema parametrima dobivenim iz CNN. U prvom dijelu rada objašnjena je metoda detekcije vozila upotrebom CNN. Kamera se koristi za dobivanje slika, na kojima zatim sustav otkriva položaj i veličinu vozila. Prilikom detekcije vozila dobivaju se četiri broja, od kojih dva predstavljaju koordinate  $(x, y)$  gornjeg lijevog ugla graničnog okvira detektiranog vozila, jedan predstavlja visinu  $(h)$ , a jedan širinu  $(w)$  istog. Širina i visina graničnog okvira detektiranog vozila su bitni parametri, jer se preko širine graničnog okvira detektiranog vozila izračunava udaljenost istog. Za detekciju se koriste brza CNN zasnovana na regiji (engl. *Fast R-CNN*), koja se izračunava u GPU (engl. *graphics processing unit*) koja koristi mrežu predloženih regija (engl. *Region Proposal Network – RPN*) za generiranje graničnih okvira detektiranog vozila. U drugom dijelu rada proračunava se udaljenost detektiranih vozila. Odnos veličine graničnog okvira oko detektiranog vozila i udaljenosti su obrnuto proporcionalne, što znači da što je veličina graničnog okvira detektiranog vozila manja, udaljenost je veća. Napravljeno je istraživanje podataka najprodavanijih vozila u 2015. godini kako bi se došlo do prosječne širine vozila, koja iznosi 1.83 m. Prema formuli 2.4 se može proračunati udaljenost.

$$D = \frac{W}{cP} \quad (2.4)$$

$W$  predstavlja poznatu širinu vozila u stvarnom svijetu, fiksna je i iznosi 1.83 m u [13] radu.  $P$  predstavlja širinu graničnog okvira detektiranog vozila izraženu u elementima slike,  $D$  predstavlja stvarnu udaljenost od kamere do vozila i  $c$  predstavlja konstantu koja je svojstvena za svaku kameru ( $c$  je recipročna vrijednost žarišnoj duljini,  $F$ , pa je i mjerna jedinica, u kojoj se  $c$  izražava 1/element slike). Na osnovu nekoliko izmjerenih udaljenosti i širina graničnih okvira detektiranih vozila na određenoj udaljenostima,  $c$  se može izračunati. Napravljena je baza od 2081 slike rezolucije 800x450 elemenata slike. Točnost detekcije vozila iznosi više od 95%. S obzirom na to da širina područja graničnog okvira oko detektiranog vozila obično bude šira od stvarne širine vozila, algoritam se dodatno morao modificirati, kako bi širina graničnog okvira hvatala rubove stop-svjetala pojedinog vozila. Nakon toga sprovedeno je određivanje eksperimentalnih parametara kamere.

U tablici 2.4 [13] se može vidjeti koliku širinu izraženu u elementima slike zauzima granični okvir detektiranog vozila na određenim udaljenostima za sliku rezolucije 800x450 elemenata slike.

Tablica 2.4. Udaljenost izražena u metrima i širina detektiranog objekta izražena u elementima slike [13]

udaljenost D [m]	3	5	6	8	9	10	12	13	14	15
širina graničnog okvira detektiranog vozila, P [element slike]	412	262	226	171	155	140	124	112	106	101

Testiranja su prvo provedena tako da su se prvo snimali procesi vožnje vozila s fiksnom kamerom. Iz videa su odabrane slike s vozilima, nakon čega se izvršavala detekcija vozila i proračun udaljenosti detektiranog vozila na tim slikama. U tablici 2.5 [13] mogu se vidjeti konačni rezultati eksperimenta.

Tablica 2.5. Izračun udaljenosti u odnosu na stvarnu udaljenost [13]

Stvarna udaljenost vozila (m)	Izračunata udaljenost vozila (m)	Apsolutna pogreška(%)
10	10.247	2.47
20	20.263	1.32
30	29.537	1.54
40	41.2	3.00

Rezultati pokazuju da je pogreška na udaljenostima od 10 m do 30 m manja od 2%, dok je na udaljenosti od 40 m pogreška 3%. Metoda za proračun udaljenosti korištenjem slike sa samo jedne kamere, korištena u ovom znanstvenom radu, predstavlja jednostavan princip koji ne iziskuje visoke troškove i koji radi jako brzo. Ima zadovoljavajuće rezultate na bližim udaljenostima, do 40m. Što se udaljenost više povećava, rezultati imaju veću pogrešku, jer što je auto udaljeniji, to pogreška od jednog elementa slike u samom graničnom okviru unosi veću pogrešku u procjenu udaljenosti. Kako bi rezultati bili točniji i na većim udaljenostima, više značajki vozila je potrebno koristiti, kako bi detekcija vozila na tim udaljenostima bila što točnija. Što je detekcija vozila točnija i bolje ocrtavanje okvira oko vozila, to je i proračunata udaljenost točnija.

U radu [14] opisan je proces proračunavanja udaljenosti upotrebom dviju kamera koje čine stereo sustav kamera. Razvoj računalnog vida, koji je u principu blizak ljudskom vidu, najbolje se

može obaviti korištenjem stereo kamera. Prilikom računanja udaljenosti objekata, prvo se računa mapa dispariteta između odgovarajućih elemenata slika dobivenih s lijeve i desne kamere. Nakon toga, mjerenja udaljenosti detektiranog objekta izvode se primjenom parametara kamere nad odgovarajućim vrijednostima disparitetne mape. Stereo vizija je proces izdvajanja 3D informacija scene iz dva 2D pogleda na istu scenu. Ulaz u proces stereo vizije je par stereo slika. Sastoji se od lijevog i desnog pogleda na scenu. Nadalje se računa mapa dispariteta slike dobivene s jedne od kamera. Vrijednosti zapisane unutar mape dispariteta koriste se za proračun udaljenosti objekata trodimenzionalne scene od kamere. Par stereo slika oponaša par ljudskih očiju. Svako oko ima drugačiji pogled na istu scenu. Dva pogleda na scenu ljudski mozak spaja u jednu sliku. Rezultat spajanja je 3D slika scene i mogućnost bolje procjene udaljenosti pojedinog dijela scene od očiju.

Testiranje za otkrivanje kretanja prema naprijed i proračun udaljenosti sprovedeno je pomoću dvaju videozapisa, koji se sastoje od 150 i 201 okvira. Točnost za detektiranje kretanja i proračun udaljenosti se izračunava prema formuli 2.5.

$$Točnost = \frac{T}{n} * 100\% \quad (2.5.)$$

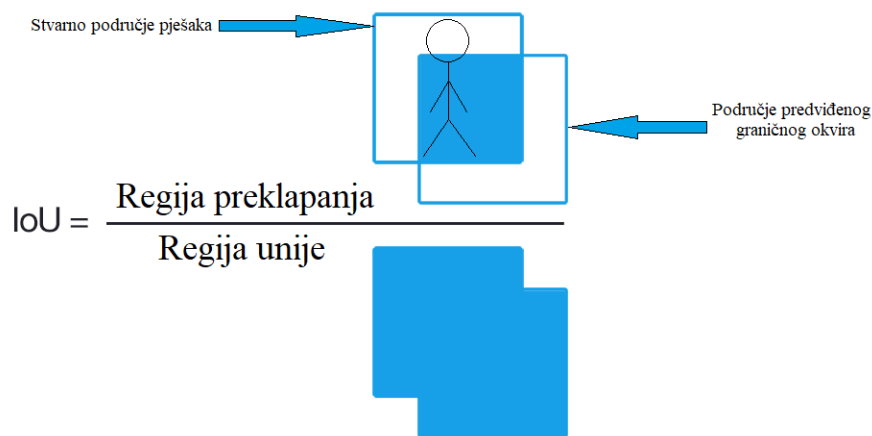
Prema jednadžbi 2.1.  $T$  predstavlja  $|TP| + |TN|$ , a  $n$  predstavlja  $|TP| + |TN| + |FP| + |FN|$ . U prvom videu ljudi koji se kreću, u jednom trenutku zastanu i miruju, a zatim nastave hodati prema naprijed. Sustav je uspio otkriti ispravno kretanje prema naprijed uz 76.8% točnosti. U drugom videu objekt se kreće samo prema naprijed bez ikakvog mirovanja. Točnost dostiže 89.4%. Prosjek točnosti oba videa je 83.1%. Stvarna udaljenost objekta od kamere iznosi 3 do 4 metra, dok je sustav procijenio da se objekt nalazi na udaljenosti između 0 i 5 metara. Dakle, stvarna udaljenost objekta spada pod procijenjeni raspon udaljenosti. Na osnovu dobivenih rezultata, može se zaključiti da sustav stereo vizije radi jako dobro. Jedan od problema koji utječu na točnost je svakako utjecaj osvjetljenja iz okoline. Optimizacijom rješenja, tako da ono bude robusno na promjenjivo osvjetljenje, točnost se može povećati. Iako najbliže ljudskom vidu, računalni vid ima nedostatke kao što su složenost, visoki troškovi i teško postizanje računanja u stvarnom vremenu.

Rad [15] opisuje detekciju pješaka zasnovanu na YOLO mrežnom modelu. Mrežni model YOLO uspješno otkriva pješake, ali će se u ovom radu njegove mogućnosti pokušati dodatno poboljšati. Glavno poboljšanje ovog rada je dodavanje 3 prolazna sloja ispred duboke mreže i promjena broja prolaznih slojeva veza u izvornoj YOLO mrežnoj strukturi iz sloja 16 u sloj 12, radi stvaranja novog modela mreže YOLO-R. U ovom je pokusu za ispitivanje korištena INRIA baza podataka. Proces detekcije pješaka kod YOLO modela započinje dijeljenjem slike u rešetku veličine  $S \times S$  regija jednakih dimenzija. Ako se pješak nalazi na slici, mreža je odgovorna za

detektiranje istog. Svaka regija predviđa  $B$  graničnih okvira za detekciju i pouzdanost (engl. *confidence*) detekcije za svaki od tih predviđenih okvira. Broj detektiranih graničnih okvira za svaku sliku je  $S_x S_y B$ . Svaki okvir za detekciju ima 5 predviđenih vrijednosti, a tu spadaju dvije koordinate  $(X, Y)$ , koje predstavljaju pomak središta okvira za predviđanje u odnosu na granicu ćelije,  $W$  i  $H$  su omjeri predviđene širine i visine okvira prema širini i visini cijele slike, a  $Conf$  predstavlja pouzdanost okvira za otkrivanje. Svaka regija predviđa uvjetnu vjerojatnost pješaka,  $Pr$ , pod uvjetom da regija sadrži pješake. Prilikom detektiranja, uvjetna vjerojatnost se množi s prediktivnom vrijednošću različitih graničnih okvira, kako bi se dobila pouzdanost za svaki granični okvir pojedinačno. Pouzdanost se računa prema formuli 2.6.

$$Pouzdanost = Pr * \frac{TruthIoU}{PredIoU} \quad (2.6.)$$

$Pr$  predstavlja vjerojatnost pojavljivanja pješaka u regija, a IoU predstavlja omjer područja preklapanja previđenog graničnog okvira i stvarnog graničnog okvira i područja unije tih dvaju okvira. Što je IoU veći (maksimalna vrijednost je 1), to je točnost detekcije pješaka veća. Na slici 2.10 [15] se može vidjeti IoU.



Sl. 2.10. Ilustracija vrijednosti IoU korištene u [15]

Testiranje ove mreže sprovedeno je pomoću INRIA baze podataka, koja se najčešće koristi za statičku detekciju pješaka. Trening skup se sastojao od 614 pozitivnih slika, na kojima se nalazilo 2416 pješaka, dok se testni skup sastojao od 288 pozitivnih slika, na kojima se nalazilo 1126 pješaka. Na slikama se uglavnom nalazi veći dio ljudskog tijela u stojećem položaju i s visinom većom od 100 elemenata slike jer slika ima visoku razlučivost. Napravljeno je usporedno testiranje YOLO v2 i YOLO – R modela.

U tablici 2.6 [15] može se vidjeti da je YOLO – R model, koji predstavlja poboljšanu verziju YOLO v2, uistinu superiorniji od YOLO v2 modela u pogledu preciznosti, točnosti i IOU. Uspoređeni su postoci propuštenih detekcija, te je on kod YOLO – R niža u odnosu na YOLO v2.

Tablica 2.6. Usporedba postotaka YOLO v2 i unaprijeđenog YOLO – R modela na bazi slika korištenoj u radu [15]

	YOLO v2	YOLO – R
Preciznost	97.37%	98.56%
Točnost	89.33%	91.21%
IOU	74.46%	76.18%
Postotak propuštenih detekcija	11.29%	10.05%

Konačno, u ovom radu predstavljen je proces poboljšanja YOLO v2 modela za detekciju pješaka. Poboljšanje je uspješno postignuto u pogledu točnosti i preciznosti detekcije pješaka te postoci propuštenih detekcija. Performanse i brzina su također poboljšani, što se ogleda u postizanju i do 25 obrađenih okvira u sekundi, čime se ispunjavaju zahtjevi detekcije pješaka u stvarnom vremenu.

Radovi analizirani u ovom poglavlju bave se detekcijom objekata i procjenom njihove udaljenosti od kamere. Oba ta zadatka sastavni su dio onoga što je u sklopu ovog diplomskog rada bilo potrebno napraviti. Treba napomenuti da se nijedan od javno dostupnih radova ne bavi konkretno problemom kojim se bavi ovaj diplomski rad, a to je iscrtavanje 2D mape položaja pješaka u okolini vozila koristeći više kamera u vozilu. Način na koji je taj problem u ovom radu riješen opisan je u sljedećem poglavlju

### **3. ALGORITAM ZA ISCRTAVANJE 2D MAPE POLOŽAJA PJEŠAKA U OKOLINI VOZILA ZASNOVAN NA KORIŠTENJU ČETIRIJU KAMERA**

Zadatak ovog diplomskog rada je unaprijediti detekciju pješaka istovremenim korištenjem slika s više kamera lociranih na različitim pozicijama u vozilu te iste detekcije pješaka na zadani način prikazati na ekranu. Za detekciju pješaka će se koristiti gotovi algoritam dostupan u definiranom razvojnom okruženju. Zatim, potrebno je procijeniti udaljenost svakog pješaka pojedinačno, te izraditi 2D mapu položaja pješaka oko vozila. Prilikom izrade potrebnih algoritama, korištene su četiri kamere, za svaku stranu vozila po jedna. Nakon sinkronizirane obrade okvira (reskaliranje, promjena formata boja i smanjivanje rezolucije) s pojedinačnih kamera, prvo je potrebno izvršiti detekciju pješaka koji stoje ili se kreću oko vozila. Nakon detekcije, potrebno je što točnije procijeniti udaljenost detektiranih pješaka. Za procjenu udaljenosti detektiranih pješaka na pojedinačnoj kameri koristit će se samo slika dobivena s te jedne kamere, uz primjenu matematičkih proračuna bez upotrebe dodatnih senzora (npr. LiDAR...) ili dodatne kamere (npr. stereo vizija). Procjenjivanje udaljenosti objekata upotrebom samo jedne kamere puno je izazovniji zadatak od procjene udaljenosti upotrebom stereo vizije ili LiDAR-a. Zbog toga će se nastojati izraditi što točniji algoritam za procjenu udaljenosti.

Za izradu 2D mape detektiranih pješaka oko vozila, prethodno je potrebno imati informacije o procijenjenoj udaljenosti pojedinačnog pješaka, kao i informaciju na slici s koje kamere se detektirani pješak nalazi. 2D mapa bi trebala predstavljati pogled na vozilo odozgor na crnoj pozadini, na kojoj bi se kao rezultat trebali nalaziti 2D model vozila i oznake detektiranih pješaka na procijenjenim udaljenostima i na točnim lokacijama u odnosu na vozilo. Problem, koji se javlja prilikom izrade 2D mape je mapiranje pješaka, koji je istovremeno detektiran u slikama dobivenim pomoću dviju susjednih kamera. S obzirom na to da su kamere, koje se koriste za detekciju pješaka, širokokutne, dio područja koje obuhvaća jedna kamera preklapa se djelomično s područjem susjednih kamera (s njezine lijeve i desne strane). Zbog toga, prilikom mapiranja pješaka (određivanja položaja pješaka na 2D mapi) koji je detektiran na slikama dobivenim pomoću dviju kamera istovremeno, dolazi inicijalno do iscrtavanja dvaju lokacijski bliskih modela istog pješaka, što (ako se vodi računa o tome) kvari krajnji rezultat jer na mapi izgleda kao da se nalaze dva pješaka blizu jedan drugoga, a zapravo je to samo jedan pješak. Za rješenje ovog problema potrebno je napisati dodatni algoritam, koji će prilikom detekcije jednog pješaka na

slikama dobivenim pomoću dviju kamera istovremeno, prepoznati da se radi o jednom pješaku i spojiti te dvije pojedinačne detekcije istog pješaka u jedan model koji će se iscrtavati na 2D mapi.

Cjelokupno rješenje zadatka sastoji se od niza algoritama koji se slijedno izvršavaju, od obrade svakog pojedinačnog okvira s kamera preko matematičkih proračuna, do konačnog rješenja, koje podrazumijeva mapiranje pješaka koji se nalaze u bližoj okolini vozila. U nastavku ovog poglavlja bit će detaljno opisano korišteno razvojno okruženje VisionSDK, a zatim izrada vlastitog slučaja upotrebe (engl. *use case*) te njegova ugradnja u okruženje. Detaljno je objašnjeno programsko rješenje svakog pojedinačnog algoritma, koji je dio rješenja cijelog zadatka, te je na kraju opisan način pokretanja programskog rješenja na realnoj platformi, koja se u diplomskom radu koristila.

### **3.1. Opis VisionSDK razvojnog okruženja i ADAS platforme na kojoj je rješenje implementirano**

Tvrtka Texas Instruments razvila je VisionSDK razvojno okruženje, koje korisnicima omogućava izgradnju različitih slučaja upotrebe, koji izvršavaju određene zadatke, kao što su obrada videa, snimanje videa, prikaz videa na ekran, pokretanje raznih algoritama za obradu videa i sl. VisionSDK razvojno okruženje napravljeno je za ADAS ploče. ADAS sustavi, koji uključuju, osim senzora, komunikacijskih mreža, algoritama, i ADAS ploče, omogućuju pomoć vozaču pri vožnji, a u budućnosti i autonomnu vožnju. Da bi to bilo moguće, na hardver koji posjeduju ADAS ploče implementiraju se razni algoritmi, koji u stvarnom vremenu obrađuju signale sa senzora, od kojih, svaki na svoj način percipiraju okolinu. Signali mogu dolaziti s raznih senzora, kao što su LiDAR, RADAR, video kamere i sl. ADAS ploče obično imaju više SoC-ova (engl. *System on a Chip*), koji se sastoje od više različitih tipova procesora, akceleratora za obradu slike i grafike, RAM i *flash* memorije te raznih ulaznih i izlaznih sučelja, koji služe za komunikaciju sa sensorima i aktuatorima. S obzirom na to da su hardverski resursi ograničeni, a zadaci koje ADAS ploče moraju izvršavati u stvarnom vremenu vrlo zahtjevni, optimizacija i razdvajanje algoritama na više procesora je neophodan korak u razvijanju konačnog rješenja.

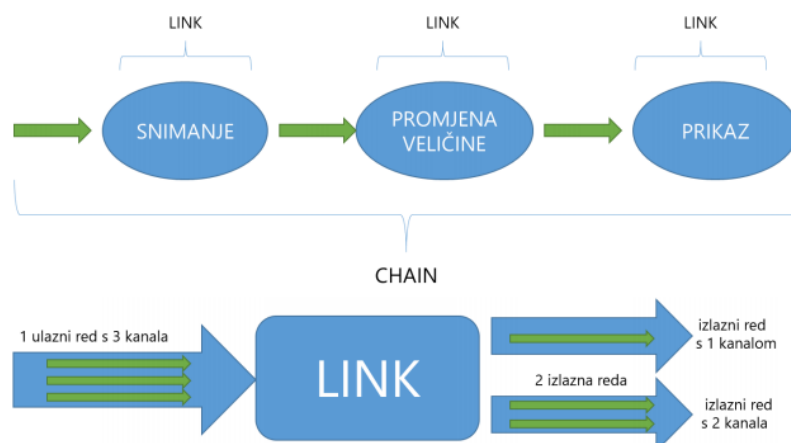


Na slici 3.1 [16] može se vidjeti ADAS ALPHA ploča, koja je korištena u ovom diplomskom radu.



Sl. 3.1. ADAS ALPHA ploča [16]

Okvir “*Links and Chains*” (“Veze i Lanci”) predstavlja okvir na kojem se zasniva VisionSDK, a korisničko programsko sučelje se naziva “*Link API*”. Ovaj okvir se koristi za definiranje slučaja upotrebe. Jedan lanac sastoji se od više veza. Svaka veza se izvršava kao zasebna nit te ona ustvari predstavlja jedan algoritam za obradu signala, kao što je prikazivanje videa na ekran, dohvaćanje videa, promjena rezolucije ulaznog signala, promjena formata boje ulaznog signala, dupliciranje ulaznih redova i sl.. Jedna veza može imati jedan ili više ulaznih i izlaznih redova. Jedan red se može sastojati od jednog ili više logičkih kanala. Slika 3.2 [1] prikazuje princip rada okvira “Veze i Lanci”.



Sl. 3.2. Ilustracija okvira “Veze i Lanci”[1]

Razvoj programskih rješenja izvršava se u C programskom jeziku. Proces izgradnje slučaja upotrebe odvija se tako da se prvo napiše konfiguracijska datoteka. Nakon toga se preko terminala pokreće izgradnja slučaja upotrebe koristeći sami alat, nakon čega se generiraju *.h*, *.c*, *.jpeg* i *log* datoteke. Na kraju se detaljno programira glavna *.c* datoteka, u kojoj se pozivaju glavne funkcije za izvršavanje slučaja upotrebe te se postavljaju svi parametri međusobno povezanih veza [1].

Prema [17] neke od glavnih veza su:

- **CAPTURE** – koristi se za dohvaćanje video okvira s kamera i prosljeđuje ih u sljedeću vezu te predstavlja početnu/ulaznu vezu slučaja upotrebe;
- **DISPLAY** – koristi se za prikaz video okvira na ekran, predstavlja završnu/izlaznu vezu iz slučaja upotrebe;
- **VPE** – koristi se za skaliranje ulaznog okvira (*up/down*), promjenu formata boja te izrezivanje određenog dijela ulaznog okvira. Ima jedan ulazni (originalni okvir) i jedan izlazni (skalirani okvir) red;
- **DUP** – koristi se za multipliciranje ulaznog reda u 2 do 6 izlaznih redova, služi za slanje jednog reda na više mjesta u slučaju upotrebe (npr. dva različita algoritma paralelno obrađuju isti ulazni okvir);
- **MERGE** – koristi se za spajanje kanala više ulaznih redova (do 6) u jedan izlazni red s više kanala;
- **SELECT** – koristi se pri odabiru jednog od kanala iz ulaznog reda. Ulaz u ovu vezu može biti samo jedan s red s više kanala, a izlaz može biti četiri reda s po jednim kanalom;
- **SPLIT** – koristi se razdvajanje kanala iz jednog ulaznog reda u više izlaznih redova;
- **SYNC** – koristi se za grupiranje, odnosno sinkronizaciju spremnika (engl. *buffer*) s više kanala u jedan spremnik zasnovano na vremenskim oznakama. Ima jedan ulazni i jedan izlazni red;
- **NULL** – predstavlja završnu/izlaznu vezu iz slučaja upotrebe. Na ulazu prima podatke iz spremnika od prijašnjih veza. Broj ulaznih redova može biti do 4. Koristi se za neku od idućih operacija:
  - Ne radi ništa s ulaznim spremnikom (*Null* operacija);
  - Zapis ulaznog spremnika u datoteku;
  - Kopiranje spremnika u memoriju;

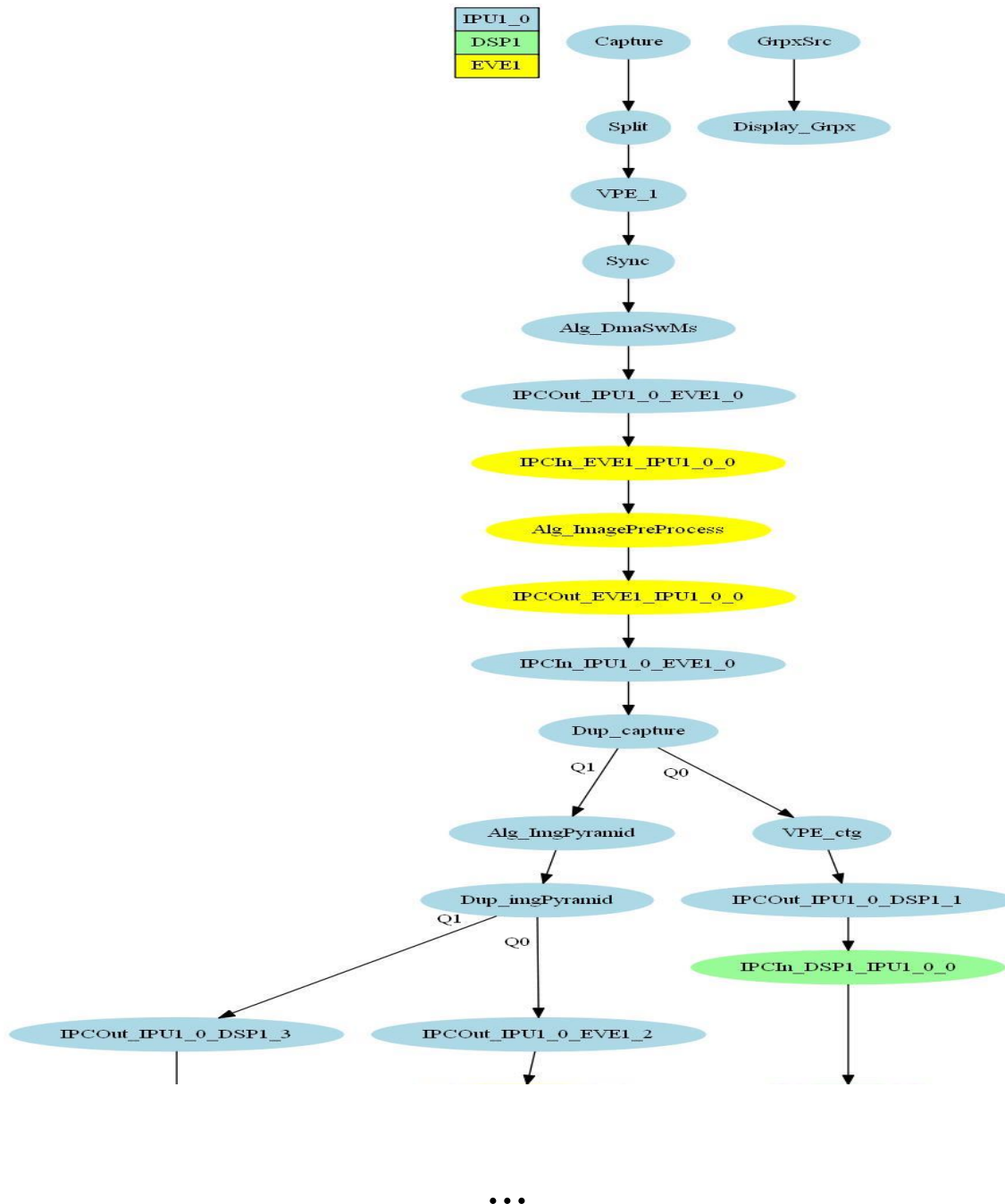
- Slanje spremnika na PC preko Ethernet komunikacijskog sučelja;
- **NULL SOURCE** - predstavlja početnu/ulaznu vezu slučaja upotrebe. Broj izlaznih redova može biti 4. Spremnici se šalju na izlazne redove tako da se prethodno:
  - Čitaju iz memorije;
  - Čitaju iz datoteke;
  - Primaju preko mreže;
- **ENCODE** – koristi se za kodiranje video okvira u skladu s MJPEG ili H264 kompresijom;
- **DECODE** – koristi se za dekodiranje video okvira kodiranih u skladu s MJPEG ili H264 kompresijom;
- **ALGORITAMSKE VEZE** – predstavljaju razne algoritme kojima se obrađuju ulazni signali. Broj ulaznih i izlaznih redova definiran je samim algoritmom. Na primjer, ulaz može biti video okvir nad kojim se vrši obrada, kao što je pretvaranje slike u sliku sive boje, rotacija slike, zrcaljenje slike, detekcija rubova i sl.

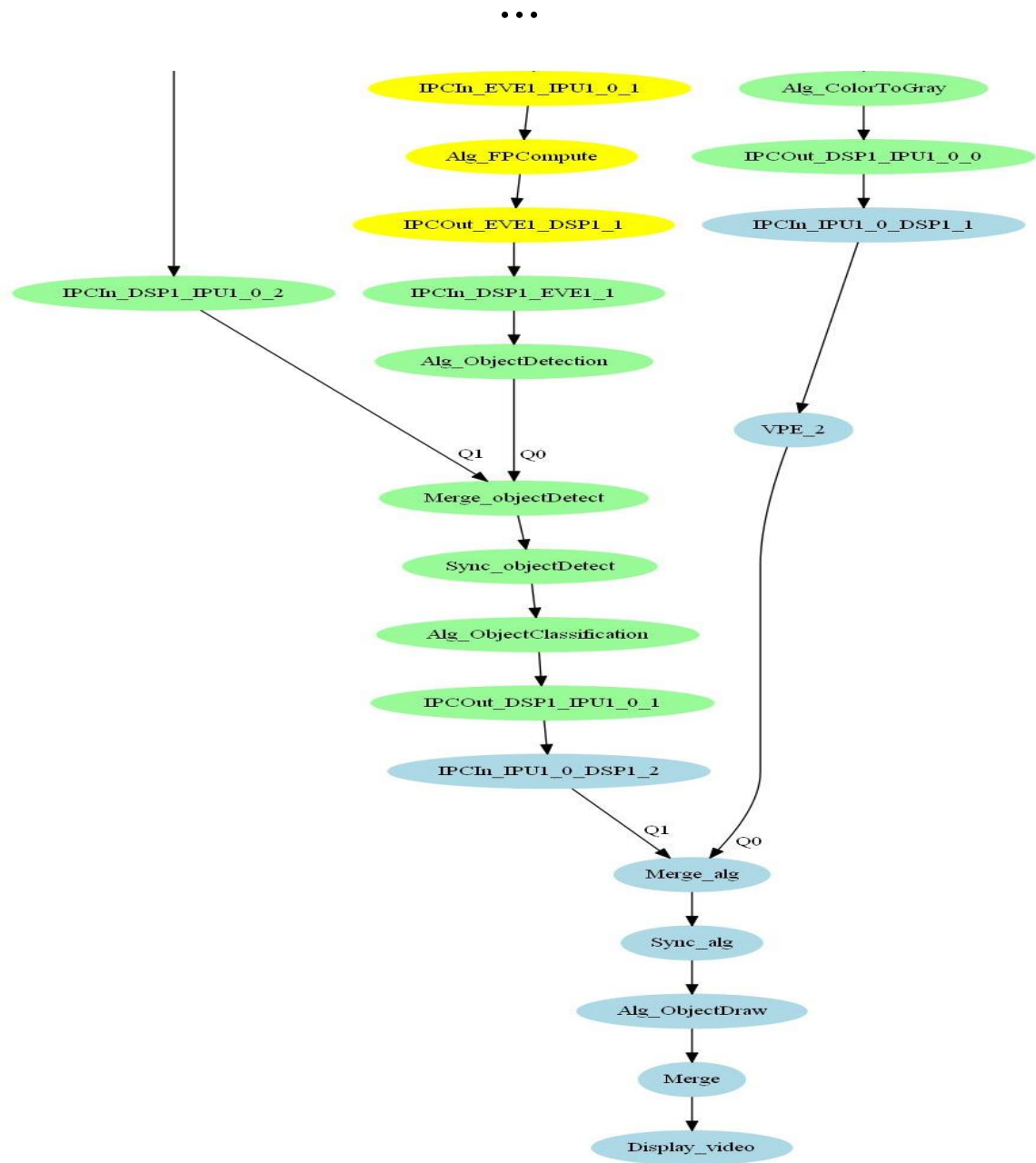
Kamere, koje su korištene prilikom razvoja programskog rješenja su širokokutne kamere RDACM24B-01 [18]. Horizontalni kut koje hvata širokokutna kamera je  $190^\circ$ , dok je vertikalni kut  $98^\circ$ . Rezolucija kamera je 1280x720 elemenata slike, a broj snimljenih okvira u sekundi (engl. *frames per second* – FPS) je 30.

### **3.2. Programsko rješenje za iscertavanje 2D mape položaja pješaka u okolini vozila koristeći četiri kamere u vozilu**

Programsko rješenje za iscertavanje 2D mape položaja pješaka u okolini vozila koristeći četiri kamere u vozilu predstavljeno je slučajem upotrebe VisionSDK okruženja, koji se sastoji od međusobno povezanih veza, odnosno algoritama za obradu ulaznih signala. S obzirom na ograničenje algoritma za detekciju pješaka, kod kojeg se proces detekcije izvršava na okvirima maksimalne rezolucije 1280x720 elemenata slike, video signalima, rezolucije 1280x720 elemenata slike, s četiri kamere, koje snimaju okolinu vozila, morala se smanjiti rezolucija na 640x360 elemenata slike, kako bi se sva četiri okvira pojedinog video signala mogla rasporediti u okvir mozaičnog prikaza rezolucije 1280x720 elemenata slike, na kojem se izvršavala detekcija.

Na slici 3.3 može se vidjeti slika stvorenog slučaja upotrebe za iscrtavanje 2D mape položaja pješaka u okolini vozila koristeći četiri kamera u vozilu.





Sl. 3.3. Slika stvorenog slučaja upotrebe za iscrtaivanje 2D mape položaja pješaka u okolini vozila koristeći četiri kamere u vozilu

Algoritmi, koji se redom izvršavaju, su sljedeći: algoritam za kreiranje mozaičkog prikaza, algoritam za pred-obradu slike, algoritam za stvaranje piramide slika, algoritam za detekciju objekata, algoritam za klasifikaciju objekata, algoritam za iscrtavanje. Algoritam za iscrtavanje je dodatno proširen algoritmima za procjenu udaljenosti detektiranih pješaka i za iscrtavanje 2D položaja pješaka u okolini vozila. Svi algoritmi, osim algoritama za procjenu udaljenosti pješaka i za iscrtavanje 2D mape, dio su VisionSDK okruženja te se trebaju pravilno povezati u funkcionalni slučaj upotrebe.

Prilikom pokretanja slučaja upotrebe, CAPTURE veza prima okvire s 4 pojedinačnih kamera te ih preko SPLIT veze razdvaja u 4 posebna kanala koji se nalaze u jednom redu. Ti kanali odlaze u VPE vezu, u kojoj se radi reskaliranje svakog videa pojedinačnog kanala, smanjuje im se rezolucija i prebacuje u određeni format boja. Nakon VPE veze, red odlazi u SYNC vezu, gdje se izvršava sinkronizacija sva četiri kanala. Iduća veza, u koju odlazi red, je algoritam za kreiranje mozaičkog prikaza. U algoritmu za kreiranje mozaičkog prikaza, prema određenom identifikacijskom broju (engl. *identity document* – ID) kanala izvršava se raspoređivanje prikaza slike s pojedinih kamera na određeni dio ekrana. Veza koja slijedi je veza pred-obrade slike, unutar koje se okvir obrađuje kako bi mu se poboljšale važne značajke (mijenjanje rezolucije ulazne slike, uklanjanje šuma, segmentacija i morfologija), a suzbila neželjena izobličenja. Izlaz iz algoritma pred-obrade slike postaje ulaz DUP\_CAPTURE veze, koja duplicira ulazni red u dva identična reda. Jedan red odlazi u VPE, gdje se podešavaju parametri potrebni za algoritam za zacrnjenje ekrana. Nakon VPE veze, red odlazi u algoritam za zacrnjenje, nakon kojeg se pozadina, na kojoj će biti prikazana 2D mapa, potpuno zacrni. Nakon algoritma za zacrnjenje red odlazi u VPE, u kojem se podešavaju parametri za odlazak u MERGE\_ALG vezu. Drugi red odlazi u algoritam piramide slika. Unutar tog algoritma ulazni okvir se izgladuje odgovarajućim filtrom te se rezolucija smanjuje 2 puta po širini i 2 puta po visini. Postupak se ponavlja za rezultirajuću sliku više puta te se tako svaki idući put dobije zaglađenija slika dvostruko manje rezolucije od prethodne. Nadalje, rezultat ovog algoritma odlazi u DUP\_IMG\_PYRAMID vezu, u kojoj se ulazni red duplicira u dva identična reda, od kojih jedan odlazi u MERGE\_OBJECTDETECTION vezu, a drugi ulazi u algoritam za detekciju objekata na objedinjenom prikazu s četiriju kamera (mozaički prikaz). Algoritam za detekciju objekata detektira pješake, vozila i prometne znakove i zatim te podatke šalje u MERGE\_OBJECTDETECT vezu. Iz te veze red odlazi u SYNC\_OBJECTDETECT, gdje se izvršava sinkronizacija, nakon koje red odlazi u algoritam za klasifikaciju. Unutar algoritma za klasifikaciju izvršava se izdvajanje određene vrste objekata, u ovom slučaju pješaka. Ti podaci se dalje šalju redom u MERGE\_ALG vezu, nakon koje se odlazi

u SYNC\_ALG vezu, gdje se izvršava sinkronizacija podataka i okvira iz mozaičkog prikaza. Iduća veza je algoritam za iscrtavanje, koji u sebi sadrži algoritme za procjenu udaljenosti i iscrtavanje 2D mape položaja pješaka u okolini vozila. Na kraju ide posljednja veza DISPLAY, koja je zadužena za prikaz 2D mape na ekranu.

### **3.2.1. Algoritam za detekciju pješaka**

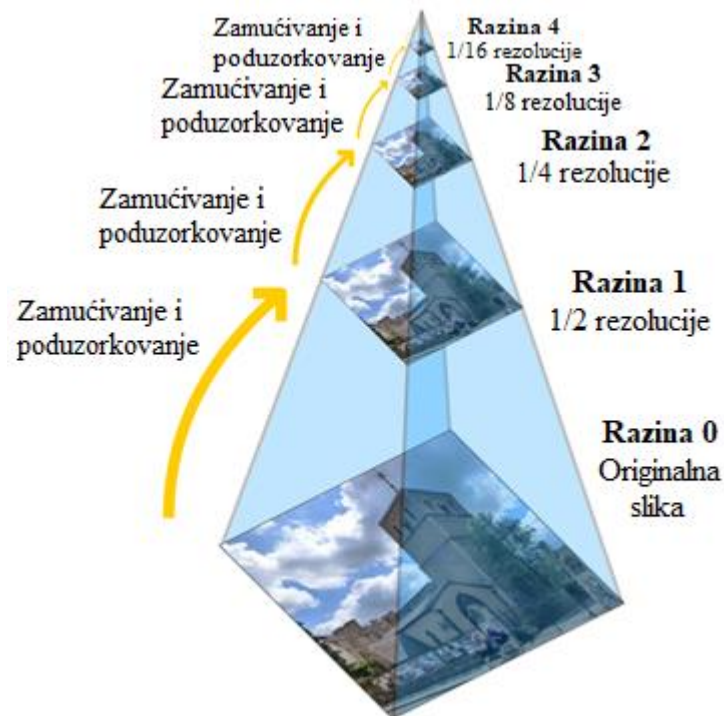
Kako bi algoritam za detekciju objekata točno radio, prethodno je potrebno implementirati algoritam za pred-obradu slike, algoritam za stvaranje piramide slika, a nakon algoritma za detekciju objekata potrebno je implementirati i algoritme za klasifikaciju te završno iscrtavanje na ekran.

Prema [19], algoritam za pred-obradu slike čini ulaznu sliku pogodnom za daljnju obradu. Kao ulaz prima okvire s više kamera spojene u jedan okvir u obliku mozaičkog prikaza. Koraci koje algoritam za pred-obradu slike radi su:

1. Mijenjanje rezolucije ulazne slike – s obzirom na to da ulazne slike u algoritam mogu biti različite rezolucije, rezoluciju svake ulazne slike je potrebno svesti na neku osnovnu rezoluciju za daljnju obradu.
2. Uklanjanje šuma – za uklanjanje šuma u slici koristi se Gaussov filter, koji je niskopropustan, a šum koji se uklanja je obično predstavljen visokim frekvencijama, što rezultira uklanjanjem detalja na slici, odnosno zamućuje sliku.
3. Segmentacija i morfologija – razdvajanje pozadine i objekata koji se nalaze bliže kameri.

Algoritam za stvaranje piramide slika od ulazne slike pravi niz smanjenih i zaglađenih slika u nekoliko koraka. Kada bi se smanjene slike postavile iznad originalne slike, dobila bi se piramida slika. Ovaj algoritam je potreban jer se prilikom detekcije objekata ne zna koje će veličine biti objekt na slici. Stoga se stvara više slika različitih rezolucija, a time i objekta, te se proces detekcije objekata radi na slikama svih rezolucija [20].

Na slici 3.4 [21] se može vidjeti ilustracija piramide dobivenih slika.



Sl. 3.4. Ilustracija dobivene piramide slika [21]

Za detekciju samih objekata od interesa korišten je gotovi algoritam dostupan unutar VisionSDK okruženja, koji je zatvorenog koda te nije moguće doznati detalje o tome kako on konkretno radi. Algoritmom za detekciju objekata se provjerava čitavo područje slike, u ovom slučaju mozaičkog prikaza s četiriju kamera, te se traže objekti na osnovu prethodno definiranih značajki svojstvenih za ljude, vozila i znakove. Nakon pronađenih objekata na slici, informacije o lokacijama objekata na slici se spremaju u strukturu, koja se dalje prosljeđuje algoritmu za klasifikaciju.

Algoritam za klasifikaciju objekata prima od algoritma za detekciju strukturu s informacijama o lokacijama svih detektiranih objekata koje je algoritam za detekciju pronašao. Na osnovu prethodno definiranog odabira, algoritam klasifikacije izdvaja informacije određene vrste objekata iz primljene strukture (u ovom slučaju izdvaja pješake), nakon čega te informacije sprema u novu strukturu, koju dalje prosljeđuje algoritmu za iscertavanje.

Posljednji algoritam, algoritam za iscertavanje, od algoritma za klasifikaciju prima strukturu s informacijama o lokacijama detektiranih pješaka. Na osnovu tih informacija algoritam za iscertavanje izvršava iscertavanje graničnog okvira oko detektiranih pješaka na svaki primljeni okvir



s mozaičkim prikazom svih četiriju kamera. Algoritam je dodatno proširen algoritmima za procjenu udaljenosti i za 2D mapiranje prepoznatih pješaka oko vozila.

### 3.2.2. Kalibracija širokokutnih kamera

Kalibracija kamere je neophodan korak u što točnijem određivanju udaljenosti s ciljem što točnijeg mapiranja pješaka oko vozila, jer su same kamere širokokutne i kao takve iskrivljuju prostor u krajevima ekrana. Još jedan od razloga zašto su se kamere morale kalibrirati je taj što točne informacije o parametrima kamere nisu bile dostupne u trenucima izrade ovog rada. Nakon kalibracije kamere, dobivene su potrebne informacije o parametrima kamere, a to je žarišnoj duljini svake kamere. Kalibracija je napravljena u Python programskom jeziku, uz koji su korištene dodatne biblioteke, kao što su OpenCV i NumPy. Na slici 3.5 se mogu vidjeti rezultati kalibracije napravljene u Python programskom jeziku.



a)



b)

Sl. 3.5. Slika (a) prije i (b) poslije kalibracije

OpenCV je biblioteka otvorenog koda. Koristi se za računalni vid i strojno učenje. Izgrađen je radi pružanja zajedničke infrastrukture za aplikacije računalnog vida i strojnog učenja. Posjeduje više od 2500 optimiziranih algoritama, od kojih su neki namijenjeni za otkrivanje i prepoznavanje lica, prepoznavanje objekata, klasificiranje objekata u videozapisima, praćenje objekata, koji se kreću na kameri, stereo viziju, spajanje više slika u jednu visoke razlučivosti, uklanjanje crvenih očiju sa slika snimljenih bljeskalicom, praćenje pokreta očiju, itd. OpenCV se može implementirati u Python, C++, Javu i MATLAB programske jezike [22]. Upute za instalaciju OpenCV na Windows mogu se pronaći u [23]. NumPy je biblioteka otvorenog koda, koja je namijenjena Python programskom jeziku. Koristi se za korištenje složenih matematičkih funkcija, višedimenzionalnih nizova i matrica [24]. Upute za instalaciju NumPy se nalaze u [25].

Ono što je bitno ovdje naglasiti, kalibracija je rađena u Pythonu, dok se sami algoritam za kalibraciju širokokutnih kamera nije uspio implementirati na ADAS ploču. No parametri kamere, potrebni za kasnije korištenje, su dobiveni.

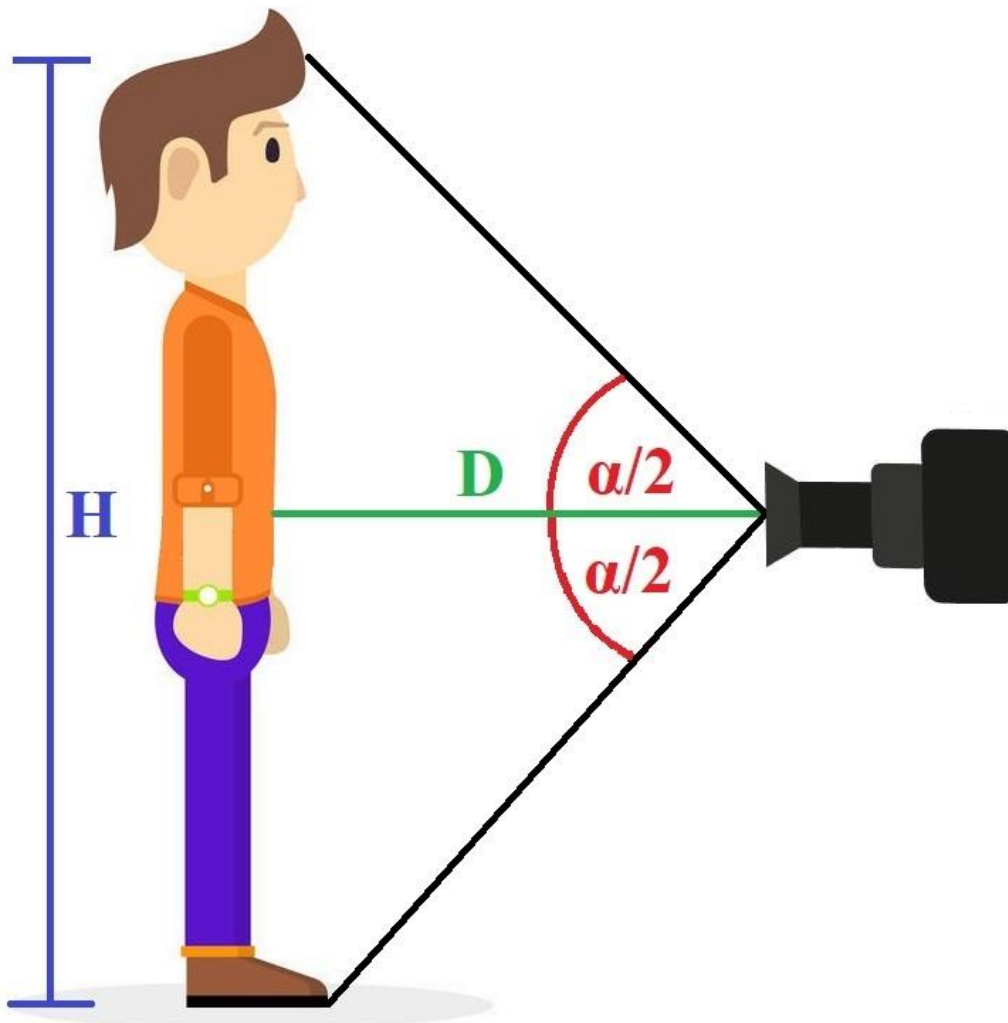
### 3.2.3. Algoritam za procjenu udaljenosti detektiranih pješaka metodom 1

U ovom slučaju se za procjenu udaljenosti koristi slika sa samo jedne kamere i matematički proračuni. S obzirom na to da se koristi širokokutna kamera, rezultati procijenjene udaljenosti više griješe kada se pješak pojavljuje u krajevima područja koja kamera snima, jer u tim područjima dolazi do najvećeg iskrivljenja slike prostora, pa tako i smanjivanja stvarne visine pješaka. Podaci o širokokutnoj kameri nisu bili javno dostupni za vrijeme izrade ovog rada. Rezolucija kamere je poznata i ona iznosi 1280x720 elemenata slike, ali vrijednosti vertikalnog i horizontalnog kuta su se morale proračunati ručno. Stoga točnost proračunatih vrijednosti zasigurno u određenoj mjeri može odstupati od stvarnih. Vertikalni kut se proračunavao tako da se prvo osoba, koja je visoka 180 cm, postavi tako da njezin prikaz na slici s kamere zauzima čitavu visinu slike. Nakon toga se ručno izmjerila udaljenost osobe od kamere. Izmjerena udaljenost iznosila je 78 cm. Na osnovu izmjerene udaljenosti i poznate visine osobe, lako se izračuna vertikalni kut kamere prema formuli 3.1.  $H$  predstavlja visinu čovjeka,  $D$  udaljenost čovjeka od kamere i  $\alpha$  predstavlja vertikalni kut.

$$\alpha = 2 * \tan^{-1} \frac{H}{2} * D \quad (3.1.)$$

Proračunato je da je vertikalni kut širokokutne kamere 98°.

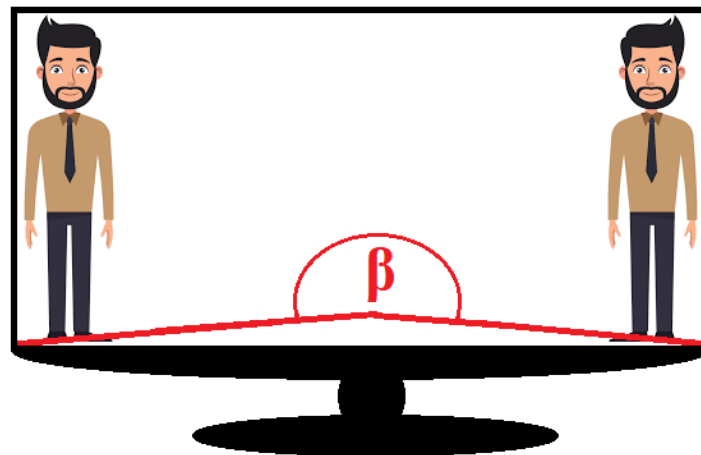
Na slici 3.6 se može vidjeti što predstavljaju koje vrijednosti.



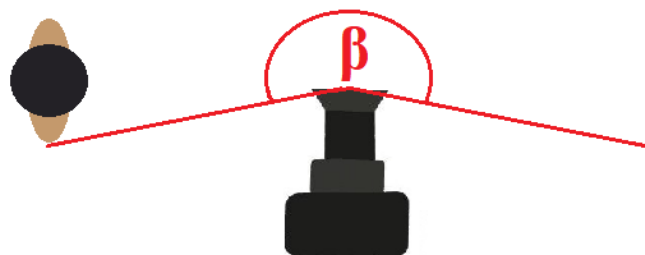
Sl. 3.6. Ilustracija postavki eksperimenta za ručni proračun vertikalnog kuta širokokutne kamere

Horizontalni kut je izmjeren tako da je jedan pješak postavljen u krajnji lijevi dio područja kamere, a zatim drugi pješak postavljen u krajnji desni dio, nakon čega se izmjerio kut između pravaca povučениh od kamere do vanjskih rubova pješaka postavljenih na tim dvjema lokacijama. Izmjeren je kut od  $190^\circ$ .

Na slici 3.7 ilustrirano je određivanje horizontalnog kuta.



a)



b)

Sl. 3.7. Ilustracija proračuna horizontalnog kuta širokokutne kamere pogleda (a) odnaprijed i (b) pogleda odozgor

Za proračun udaljenosti potrebne su informacije o vertikalnom kutu kamere, visini slike izraženoj u elementima slike, visini graničnog okvira detektiranog pješaka izraženoj u elementima slike, te prosječnoj visini čovjeka. Prosječna visina čovjeka u Europi prema [26] iznosi 177 cm. Proračun udaljenosti čovjeka se zasniva prvo na izračunu odnosa visine graničnog okvira detektiranog čovjeka i visine slike. Vrijednost koja se proračuna je manja od 1. Nakon toga se prosječna visina čovjeka dijeli s tim odnosom i dobiva se prividno veća visina čovjeka. Pomoću te visine i poznatog vertikalnog kuta, lako se izračunava udaljenost pješaka. Ukratko rečeno, koliko puta je visina graničnog okvira manja od visine ekrana, toliko puta je udaljenost pješaka veća od proračunate udaljenosti od 78 cm kada pješak zauzima čitavu visinu ekrana. U formuli 3.2  $D$  predstavlja udaljenost pješaka od kamere izražena u metrima.  $AvgHh$  predstavlja prosječnu visinu čovjeka izraženu u metrima,  $Sh$  je visina slike kamere na ekranu izražena u elementima

slike, dok  $Hh$  predstavlja visinu graničnog okvira detektiranog pješaka također izražena u elementima slike.  $\alpha$  predstavlja vertikalni kut širokokutne kamere.

$$D = \frac{\frac{AvgHh}{2} \frac{Sh}{Hh}}{\tan \frac{\alpha}{2}} \quad (3.2.)$$

Dodatne korekcije za proračun udaljenosti pješaka, kada se on ne nalazi ravno ispred kamere nego se nalazi pod nekim kutem od kamere, nije potrebno raditi jer granični okvir je onoliko visok koliko je i visok pješak na slici s kamere bez obzira na to da li je on direktno ispred kamere ili pod nekim kutom. Točnost procijenjene udaljenosti pješaka ovisi o rezoluciji slike na kojoj se izvršava detekcija objekata.

### 3.2.4. Algoritam za procjenu udaljenosti pješaka metodom 2

Kod ove metode za procjenu udaljenosti koristi se slika sa samo jedne kamere, kao i kod prethodne metode, te malo drugačiji matematički proračuni od prethodne metode. Zbog korištenja širokokutne kamere, točnost procijenjene udaljenosti u krajevima područja, koja kamera snima, je manja zbog iskrivljenja prostora. Za metodu procjene udaljenosti detektiranog pješaka upotrebom parametra žarišne duljine, potrebna je naravno informacija o žarišnoj duljini kamere. S obzirom na to da podaci o modelu kamere, koja je korištena za izradu ovog diplomskog rada, nisu dostupni na internetu, prethodno napravljena kalibracija kamere, objašnjena u potpoglavlju 3.2.2., dala je, kao rezultat, parametar žarišne duljine kamere. Žarišne duljine svake od četiriju kamera se malo razlikuju, jer ni jedna žarišna duljina kamere nije identična drugoj žarišnoj duljini neke druge kamere, ali su te razlike zanemarive, stoga parametar žarišne duljine širokokutne kamere rezolucije 1280x720 elemenata slike, koji se dobio kalibracijom, iznosi 330 elemenata slike. S obzirom na to da se procjena udaljenosti, u slučaju upotrebe konačnog rješenja, računa na slici rezolucije 640x360 elemenata slike, pod pretpostavkom da je to ista širokokutna kamera, samo smanjene rezolucije dva puta po širini i dva puta po visini, uzeta je u obzir polovina iznosa žarišne duljine, dobivene kalibracijom kamere. Razlika između kamera i korištenje polovine iznosa žarišne duljine dobivene kalibracijom kamere unosi pogrešku pri proračunu udaljenosti ovom metodom, što će detaljnije biti objašnjeno u idućem poglavlju.

Kao i za prethodno opisanu metodu, prilikom računanja udaljenosti detektiranog pješaka od kamere metodom, u kojoj se upotrebljava parametar žarišne duljine kamere, osim parametra žarišne duljine, potrebni su i podaci o prosječnoj visini čovjeka prema [26] i visini graničnog okvira detektiranog pješaka. Prema radu [13], koji je opisan u drugom poglavlju i u kojem se

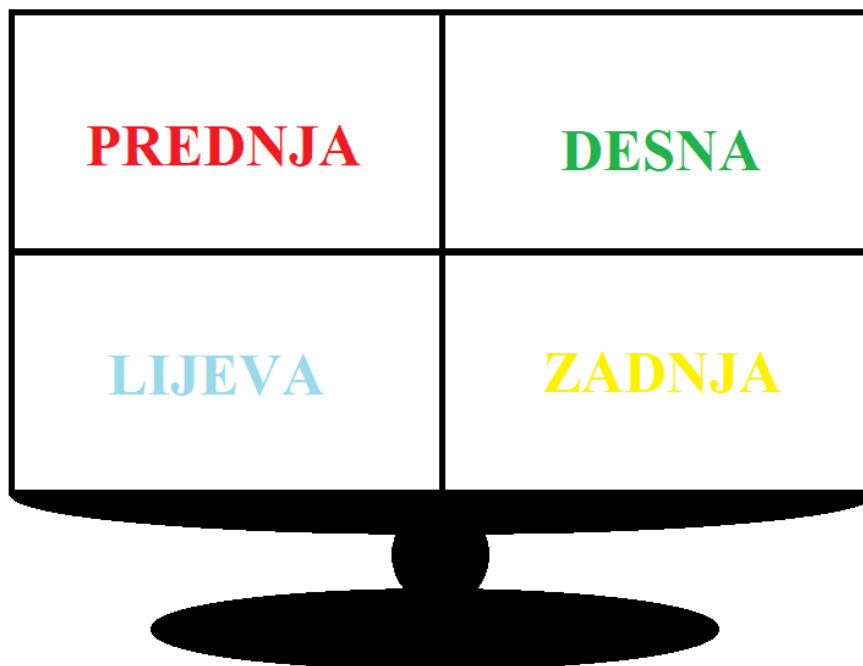
računa udaljenost detektiranih vozila primjenom ove metode, udaljenost detektiranog pješaka se računa tako da se prosječna visina čovjeka izražena u metrima, pomnoži sa žarišnom duljinom, izraženom u elementima slike, te se to sve podijeli s visinom graničnog okvira detektiranog pješaka, izraženom u elementima slike. Proračun ovom metodom definiran je jednačbom 3.3.

$$D = \frac{H * F}{P} \quad (3.3.)$$

U prethodnoj formuli  $D$  predstavlja udaljenost detektiranog pješaka,  $H$  predstavlja prosječnu visinu čovjeka,  $F$  predstavlja žarišnu duljinu dok  $P$  predstavlja visinu graničnog okvira detektiranog pješaka.

### 3.2.5. Algoritam za iscrtavanje 2D mape položaja pješaka u okolini vozila

Iscrtavanje 2D mape položaja pješaka u okolini vozila slijedi iz proračuna udaljenosti pojedinog pješaka. S obzirom na to da se detekcija pješaka izvršava na okviru koji se sastoji od 4 manja prikaza (mozaički prikaz istovremenih slika s četiriju kamera), informacija o tome na kojoj slici (s koje kamere) je pronađen pješak mora se izračunati na osnovu pozicije graničnog okvira oko detektiranog pješaka. Jedan okvir je rezolucije 1280x720 elemenata slike, što znači da svakoj kameri pripada područje rezolucije 640x360 elemenata slike u mozaičkom prikazu. Na slici 3.8 se može vidjeti koje područje ekrana zauzima koja kamera.



Sl. 3.8. Raspored prikaza slike s četiriju kamera u mozaičkom prikazu.

Mapa okoline vozila trebala bi predstavljati pogled iz ptičje perspektive na crnoj pozadini s modelom vozila u sredini mape te modelima svakog detektiranog pješaka, koji stoji ili se kreće oko vozila. Svaki se pješak se iscrtava (modelira) kao poseban kvadratić. S obzirom na to da se detekcija, proračun udaljenosti i mapiranje izvršavaju za svaki okvir, a to znači 30 puta u sekundi, može se dobiti fluidan prikaz kretanja pješaka oko vozila. Prilikom detektiranja pješaka, algoritam za detekciju objekata sprema informacije o svakom pješaku u polje struktura. Informacije, koje su potrebne za određivanje lokacije slike na kojoj se pješak nalazi, su  $x$  i  $y$  koordinate gornjeg lijevog kuta graničnog okvira detektiranog pješaka. Nakon određivanja kamere slijedi postupak mapiranja pješaka.

Na slici 3.9 se može vidjeti kod funkcije za određivanje na kojoj slici (s koje kamere) je pronađen pješak.

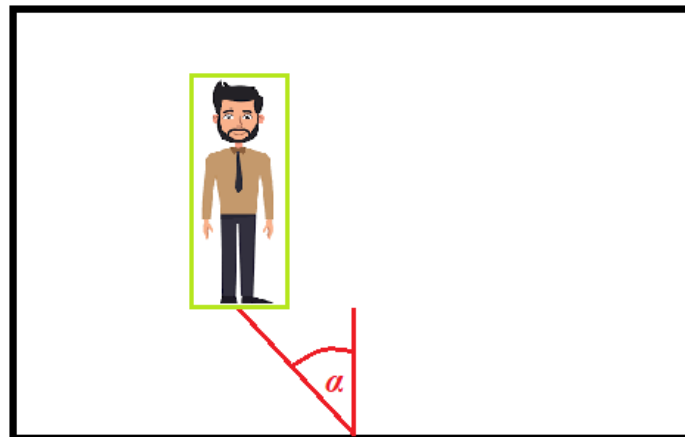
```

1. int FunWhichCamera(int xPos, int yPos, int objWidth, int objHeight){
2.     //#### PRVA KAMERA
3.     if((xPos <= 640 && yPos <= 360) && ((xPos + objWidth) <= 700 && (yPos + objHeight)
4.         <= 420)){
5.         return 1;
6.     }
7.     //#### DRUGA KAMERA
8.     else if((xPos <= 1280 && xPos > 580 && yPos <= 360) && ((xPos + objWidth) <= 1280
9.         && (xPos + objWidth) > 640 && (yPos + objHeight) <= 420) ){
10.        return 2;
11.    }
12.    //#### TRECA KAMERA
13.    else if((xPos <= 640 && yPos <= 720 && yPos > 300) && ((xPos + objWidth) <= 700 &&
14.        (yPos + objHeight) <= 720 && (yPos + objHeight) > 360 )){
15.        return 3;
16.    }
17.    //#### CETVRTA KAMERA
18.    else if((xPos <= 1280 && xPos > 580 && yPos <= 720 && yPos > 300) && ((xPos + objW
19.        idth) <= 1280 && (xPos + objWidth) > 640 && (yPos + objHeight) <= 720 && (yPos + objHe
20.        ight) > 360)){
21.        return 4;
22.    }
23.    else{
24.        return 0;
25.    }
26. }
```

Sl. 3.9. Kod funkcije za određivanje na kojoj slici (s koje kamere) je pronađen pješak

Za mapiranje pojedinog pješaka potrebne su informacije o prethodno proračunatoj udaljenosti, na slici s koje kamere je detektiran pješak, vrijednost horizontalnog kuta i pozicija graničnog okvira na ekranu. Dobivanje 2D pozicije detektiranih pješaka započinje tako da, ovisno na kojoj kameri je pješak detektiran, primjenjuje se određeni način mapiranja. Prvo što se računa je kut za lokaciju na kojoj je pješak detektiran.

Na slici 3.10 može se vidjeti kut pod kojim se pješak nalazi od kamere.



Sl. 3.10. Kut pod kojim se pješak nalazi u odnosu na kameru

Proračun kuta položaja pješaka se računa tako da se  $x$  koordinata sredine graničnog okvira po širini izražene u elementima slike podijeli sa širinom područja kojeg zauzima kamera u čitavom prikazu, a u ovom slučaju je to polovina od 1280, tj. 640, te se dobije broj manji 1. Nakon toga se određuje kut tako da se omjer pomnoži s vrijednošću horizontalnog kuta. Formula 3.4 prikazuje proračun kuta položaja pješaka.

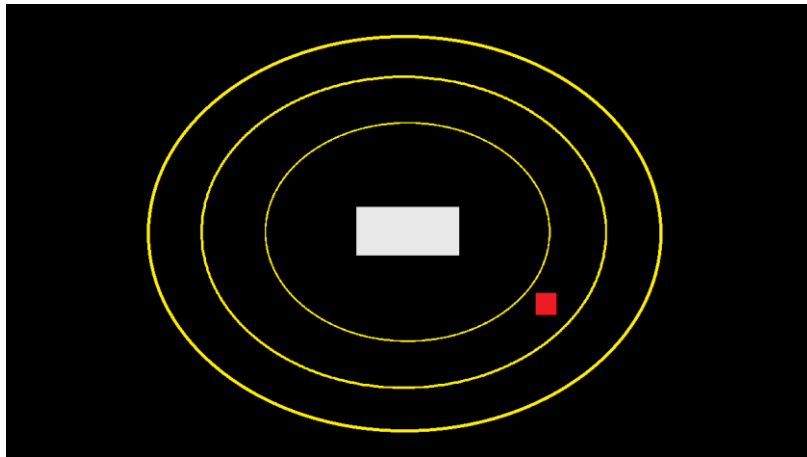
$$\alpha = \frac{X_{pos}}{Sw} * \beta \quad (3.4.)$$

$\alpha$  predstavlja kut položaja pješaka.  $X_{pos}$  predstavlja poziciju sredine graničnog okvira detektiranog pješaka po širini izraženu u elementima slike.  $Sw$  je širina slike kamere na ekranu, izražena u elementima slike, dok  $\beta$  predstavlja horizontalni kut širokokutne kamere, koji iznosi  $190^\circ$ .

Idući korak je proračun 2D  $x$  i  $y$  koordinata položaja pojedinog pješaka u pogledu odozgor. 2D mapa koncipirana je tako da se model autića nalazi na sredini ekrana i da mu prednja strana gleda od sredine ekrana prema desno. Zadnja strana vozila gleda prema lijevo, lijeva strana prema gore, a desna prema dole. Udaljenost od jednog metra predstavljena je sa 100 elemenata slike. Ova vrijednost je proizvoljno izabrana, te se može promijeniti u bilo kojem trenutku, što će samo utjecati na konačni prikaz 2D mape (veće ili manje koncentrične elipse). S obzirom na različitu dužinu i širinu vozila, formirane su koncentrične elipse umjesto koncentričnih kružnica, tako da su iscrtane na udaljenosti od 100, 200 i 300 elemenata slike od pojedine kamere na vozilu. Na osnovu informacije o tome na kojoj je kameri pješak detektiran, na osnovu proračunate udaljenosti i kuta, u tom smjeru se na udaljenosti, prebačenoj u elemente slike, od sredine ekrana iscrtava kvadratić.

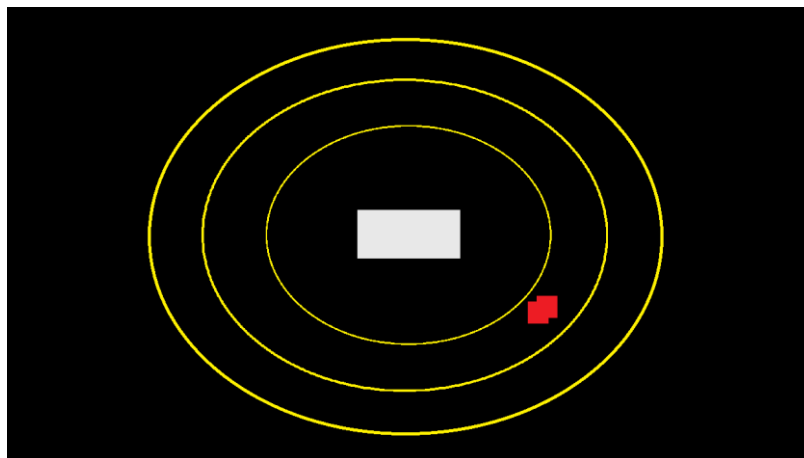


Iduća slika 3.11 prikazuje mapu s iscrtanim autićem i pješakom, koji se nalazi na udaljenosti od 1.5 m detektiran prednjom i desnom kamerom.



Sl. 3.11. Mapa s iscrtanim autićem i pješakom

S obzirom na to da su kamere širokokutne, područje prikaza svake kamere ulazi u područje susjednih lijevih i desnih kamera. To znači da je određeni dio okoline vozila uvijek prikazan na slikama s dviju susjednih kamera. Tu se javlja problem jer pješak često bude detektiran na slikama s dviju susjednih kamera, što rezultira iscrtavanjem dvaju kvadratića jedan neposredno blizu drugog, od kojih oba pripadaju istom pješaku. To dovodi do netočnih rezultata, jer u tom slučaju mapa prikazuje više pješaka nego što ih zapravo ima. Na slici 3.12 nalazi se 2D mapa, na kojoj se nalaze iscrtana dva kvadratića jednog detektiranog pješaka, koji se nalazi na udaljenosti od 1.5 m detektiran prednjom i desnom kamerom, a radi se o pješaku, koji je nakon obrade informacija prikazanih na slici 3.12 prikazan na slici 3.11.



Sl. 3.12. 2D mapa na kojoj se nalaze iscrtana dva kvadratića jednog detektiranog pješaka

Na slici 3.13 se može vidjeti dio kod funkcije za određivanje 2D pozicije položaja detektiranog pješaka na prednjoj i desnoj kameri.

```
1. void xyPos2D (int whichCamera, float *xPos2D, float *yPos2D, float distance,
2. int xPos, int objWidth, char *color){
3.     float xScale;
4.     int param = 100;
5.     distance2D = distance * param;
6.     float fi;
7.     float alfa;
8.
9.     switch(whichCamera){
10.        case 1:
11.            xScale = (float)((xPos + (float)(objWidth/2))/640);
12.            fi = xScale * 190;
13.            if (fi <= 90){
14.                alfa = 90 - fi;
15.                *yPos2D = 720./2 - (float)(sin(alfa * 3.14 / 180) * distance2
D) - (0.135 * param);
16.            }else{
17.                alfa = fi - 90;
18.                *yPos2D = 720./2 + (float)(sin(alfa * 3.14 / 180) * distance2
D) + (0.135 * param);
19.            }
20.            *xPos2D = 1280./2 + (float)(cos(alfa * 3.14 / 180) * distance2D)
+ (0.25 * param);
21.            *color = 'r';
22.            break;
23.        case 2:
24.            xScale = (float)((xPos - 640 + (float)(objWidth/2))/640);
25.            fi = xScale * 190;
26.            if (fi <= 90){
27.                alfa = 90 - fi;
28.                *xPos2D = 1280./2 + (float)(sin(alfa * 3.14 / 180) * distance
2D) + (0.25 * param);
29.            }else{
30.                alfa = fi - 90;
31.                *xPos2D = 1280./2 - (float)(sin(alfa * 3.14 / 180) * distance
2D) - (0.25 * param);
32.            }
33.            *yPos2D = 720./2 + (float)(cos(alfa * 3.14 / 180) * distance2D) +
(0.135 * param);
34.            *color = 'g';
35.            break;
36.        ...
```

Sl. 3.13. Dio koda funkcije za određivanje 2D pozicije položaja detektiranog pješaka na prednjoj i desnoj kameri

Rješenje ovog problema leži u pisanju dodatnog algoritma, koji će prilikom detekcije pješaka na slikama s dviju susjednih kamera, prepoznati da se radi o istom pješaku i iscrtati samo jedan kvadratić. Kada je pješak prepoznat na slikama s dviju susjednih kamera, na jednoj kameri on bude prepoznat na sredini, a na drugoj kameri u jednom od uglova, ili na obje kamere bude prepoznat otprilike na istom mjestu. Kako su kamere širokokutne, prostor u krajevima kamera je iskrivljen, a time i pješak koji je detektiran. Iskrivljenost pješaka dovodi do loše detekcije (manjeg

graničnog okvira oko pješaka nego što bi on trebao biti da je slika ispravljena ili da kamera nije širokokutna), a time i do loše procjene udaljenosti, što na kraju rezultira lošim mapiranjem. U slučaju kada je pješak prepoznat na slikama s dviju susjednih kamera tako da se na prvoj nalazi na sredini, a da se na drugoj nalazi pri krajevima, gdje se prostor iskrivljuje, točnost procijenjene udaljenosti je puno veća na prvoj kameri, nego točnost procijenjene udaljenosti na drugoj kameri. Zato se u ovakvom primjeru uzima u obzir samo procijenjena udaljenost s prve kamere.

Prikaz s kamere podijeljen je u 6 regija, područja od interesa (engl. *region of interest* – *ROI*), s lijeva na desno, kojima su pridijeljene određene težine za računanje procjene udaljenosti pješaka. Područja kamera su podijeljena tako da je širina prikaza pojedine kamere, u ovom slučaju 640 elemenata slike, podijeljena na 5 područja širine 106 elemenata slike i zadnje područje širine 110 elemenata slike. Područjima 1 i 6 sa slike 3.14 pridijeljena je težina 0, područjima 2 i 5 težina 0.5, a područja 3 i 4 težina 1.



Sl. 3.14. Područja kamera s pripadajućim težinama

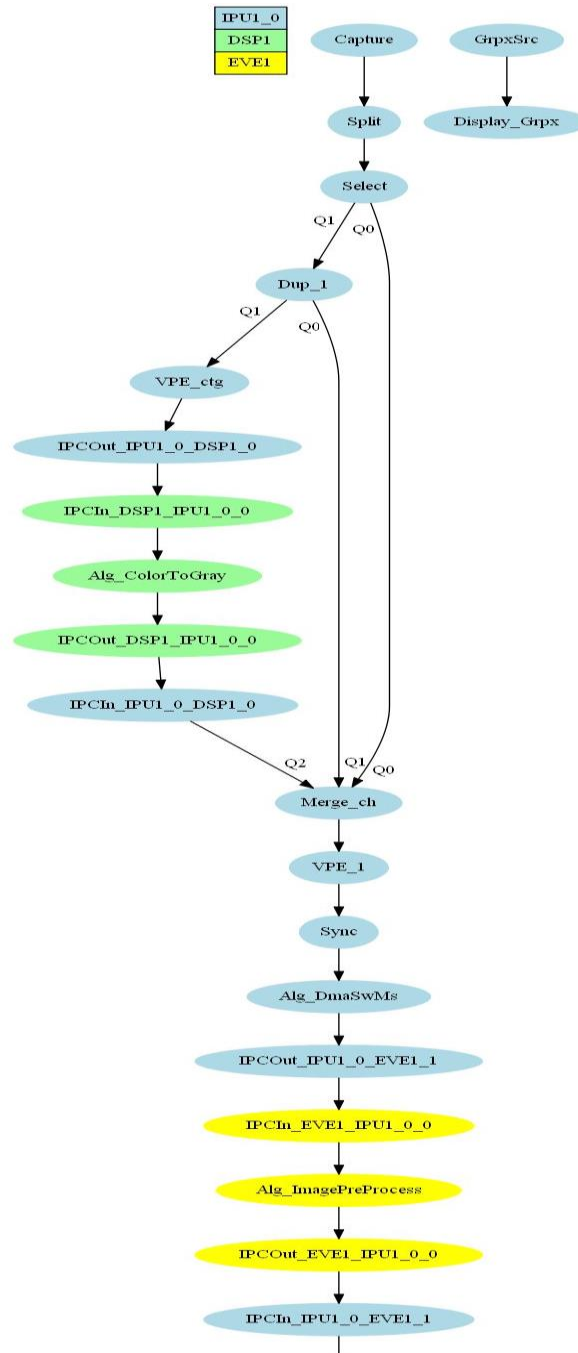
To znači da ako je osoba prepoznata na slikama s dviju susjednih kamera, a na prvoj se nalazi u području veće težine, dok je na drugoj u području manje težine, procijenjena udaljenost pješaka s prve kamere je točnija nego procijenjena udaljenost pješaka s druge kamere, te se za mapiranje uzima u obzir samo procijenjena udaljenost s prve kamere. U slučaju da se na slikama s dviju susjednih kamera pješak nađe u područjima od interesa iste težine, za mapiranje se računa srednja vrijednost procijenjenih udaljenosti s obje kamere.

Algoritam za rješavanje gore navedenih problema sastoji se od raznih funkcija kojima se dolazi do točnog mapiranja. Prvo je potrebno napraviti polja struktura prepoznatih pješaka sa svake kamere pojedinačno. To znači da svaka kamera ima svoje polje struktura, u kojem se nalaze informacije o svim pješacima detektiranim na toj kameri. Struktura svakog detektiranog pješaka sadrži  $x$  koordinatu gornjeg lijevog ugla graničnog okvira detektiranog pješaka, zatim 2D  $x$  i  $y$  koordinate potrebne za mapiranje i težinu područja od interesa, koja je pridijeljena detektiranom pješaku na osnovu  $x$  koordinate gornjeg lijevog ugla graničnog okvira. Nakon proračuna 2D  $x$  i  $y$  koordinate pješaka, određuje se težina područja od interesa detektiranog pješaka i ovisno na kojoj kameri je pješak detektiran, u polje te kamere se spremaju informacije o njemu, potrebne za daljnju obradu. Nakon obrađenih svih detektiranih pješaka s jednog okvira, pozivaju se funkcije za sortiranje detektiranih pješaka na pojedinoj kameri prema  $x$  koordinati graničnog okvira detektiranog pješaka. Sortiranje je potrebno da bi se redom mogli mapirati detektirani pješaci. Nakon što je okvir obrađen, pješaci raspoređeni po kamerama i sortirani prema  $x$  koordinatama, slijedi mapiranje svih pješaka prema težini područja na kojem je pješak detektiran. Kod konačnog rješenja za iscrtavanje 2D mape položaja pješaka u okolini vozila upotrebom 4 kamere, može se vidjeti u elektroničkom prilogu ovom radu P.3.1.

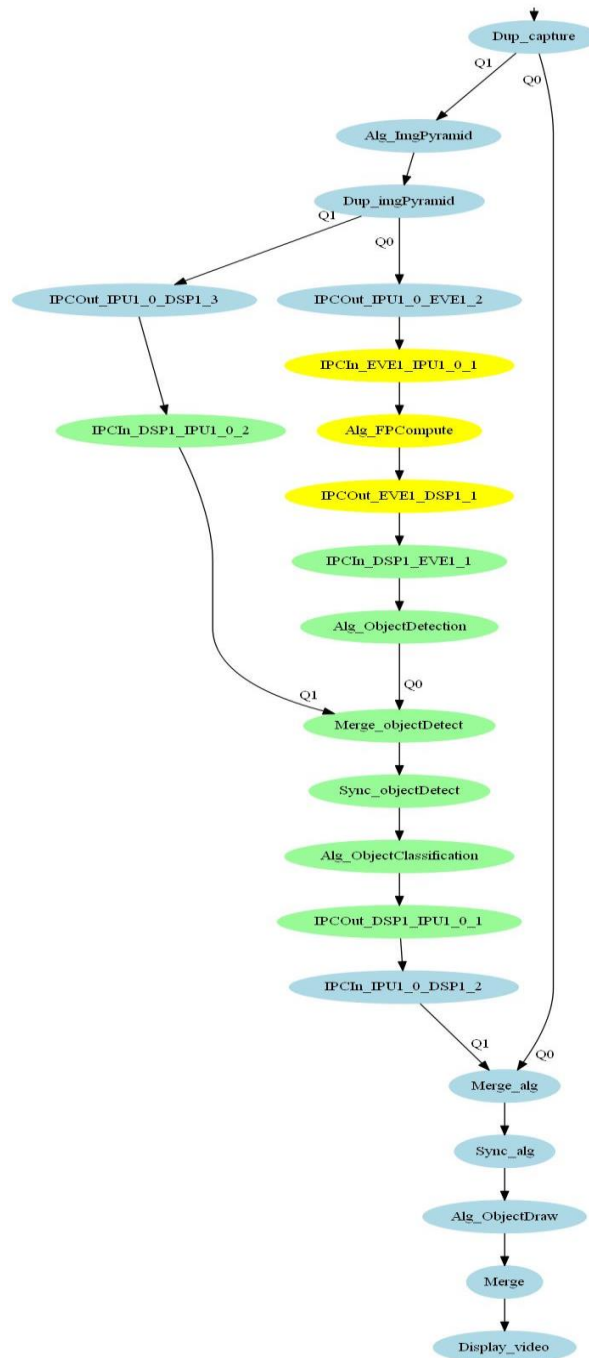
### **3.3. Programsko rješenje za iscrtavanje 2D mape položaja pješaka u okolini vozila koristeći dvije kamere**

Drugi slučaj upotrebe razvijen je kao programsko rješenje za iscrtavanje 2D mape položaja pješaka u okolini vozila koristeći dvije kamere. Postoje dvije kombinacije kamere ovog slučaja upotrebe, a to je kombinacija prednje i desne kamere i kombinacija prednje i lijeve kamere.

Na slici 3.15 može se vidjeti slika stvorenog slučaja upotrebe za iscrtavanje 2D mape položaja pješaka u okolini vozila koristeći dvije kamera u vozilu.



...



Sl. 3.15. Slika stvorenog slučaja upotrebe za iscrtavanje 2D mape položaja pješaka u okolini vozila koristeći dvije kamere u vozilu

Ovaj slučaj upotrebe napravljen je radi ispitivanja da li algoritam za detekciju pješaka radi bolje na većoj rezoluciji, odnosno da bi se pokazalo kako se mijenja točnost detekcije i

procjene udaljenosti s povećanjem rezolucije S obzirom na to da kod slučaja upotrebe konačnog rješenja, algoritam detekcije objekata radi na mozaičkom prikazu rezolucije 1280x720 elemenata slike, od koje svaka kamera zauzima područje 640x360 elemenata slike, proračun udaljenosti detektiranog pješaka se računa na rezoluciji koja je dva puta manja po širini i dva puta manja po visini od HD rezolucije, te radi do maksimalno 3 metra. Radi pokušaja dobivanja proračuna udaljenosti većih od 3 metra, razvijen je slučaj upotrebe, kod kojeg će se proračun udaljenosti računati na rezoluciji 960x540 elemenata slike. Prvo se pred-odabirom odabire kombinacija kamera kojima će se hvatati određeno područje okoline vozila. Postoje dvije mogućnosti, a to su područje vozila koje hvataju prednja i lijeva kamera i područje vozila koje hvataju prednja i desna kamera. Svi algoritmi detekcije, proračuna udaljenosti i iscrtavanja 2D mape položaja pješaka, u ovom slučaju upotrebe rade na isti princip kao što je objašnjeno u prethodnom potpoglavlju, u kojem se detaljno opisuje slučaj upotrebe konačnog rješenja, samo s različitim parametrima.

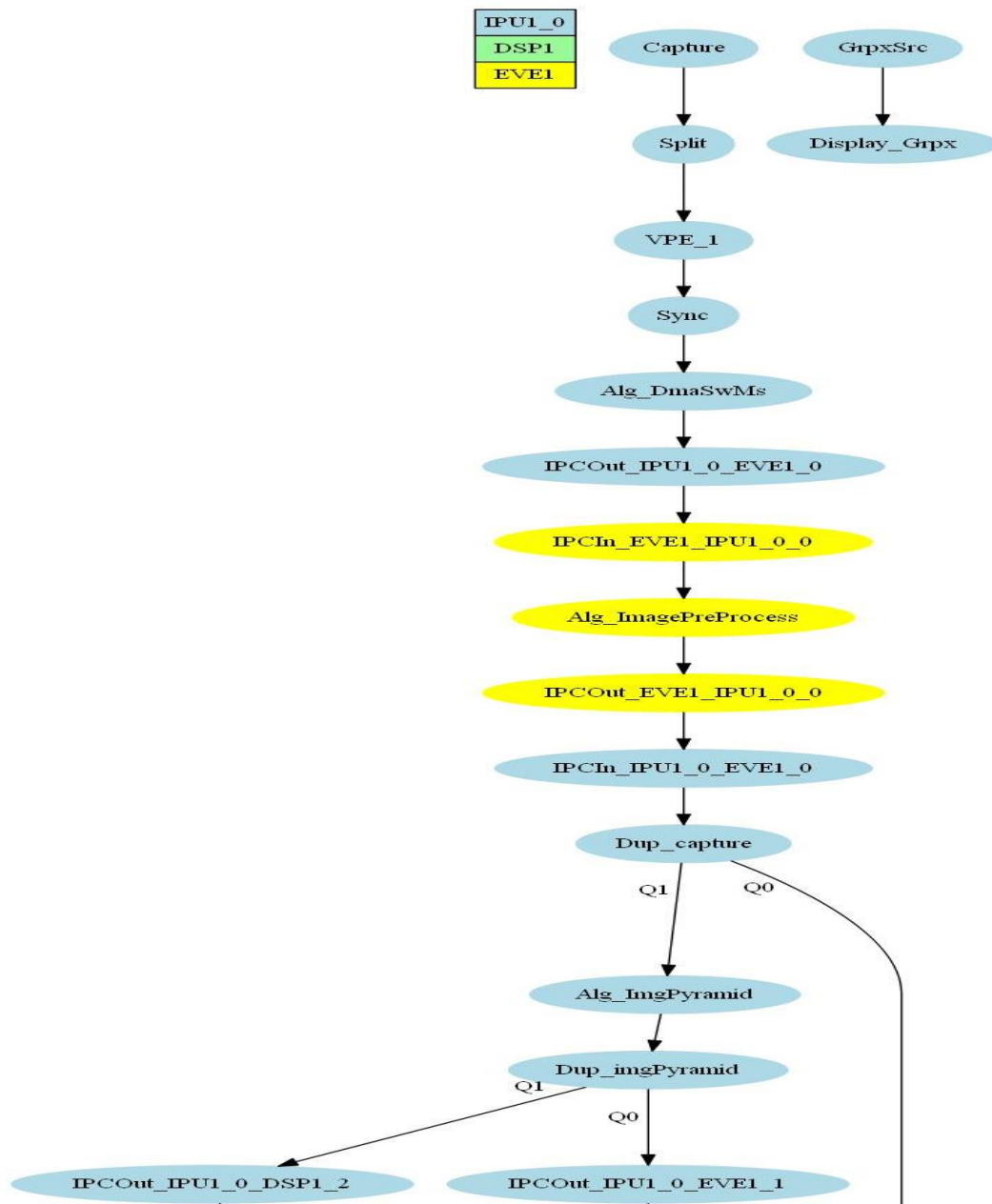
Na slici 3.16 se može vidjeti prikaz dobiven na ekranu kad se pokrene slučaj upotrebe iscrtavanja 2D mape položaja pješaka u okolini vozila koristeći 2 kamere (prednja i desna) u vozilu.



Sl. 3.16. Prikaz na ekranu dobiven kad se pokrene slučaj upotrebe iscrtavanja 2D mape položaja pješaka u okolini vozila koristeći 2 kamere u vozilu

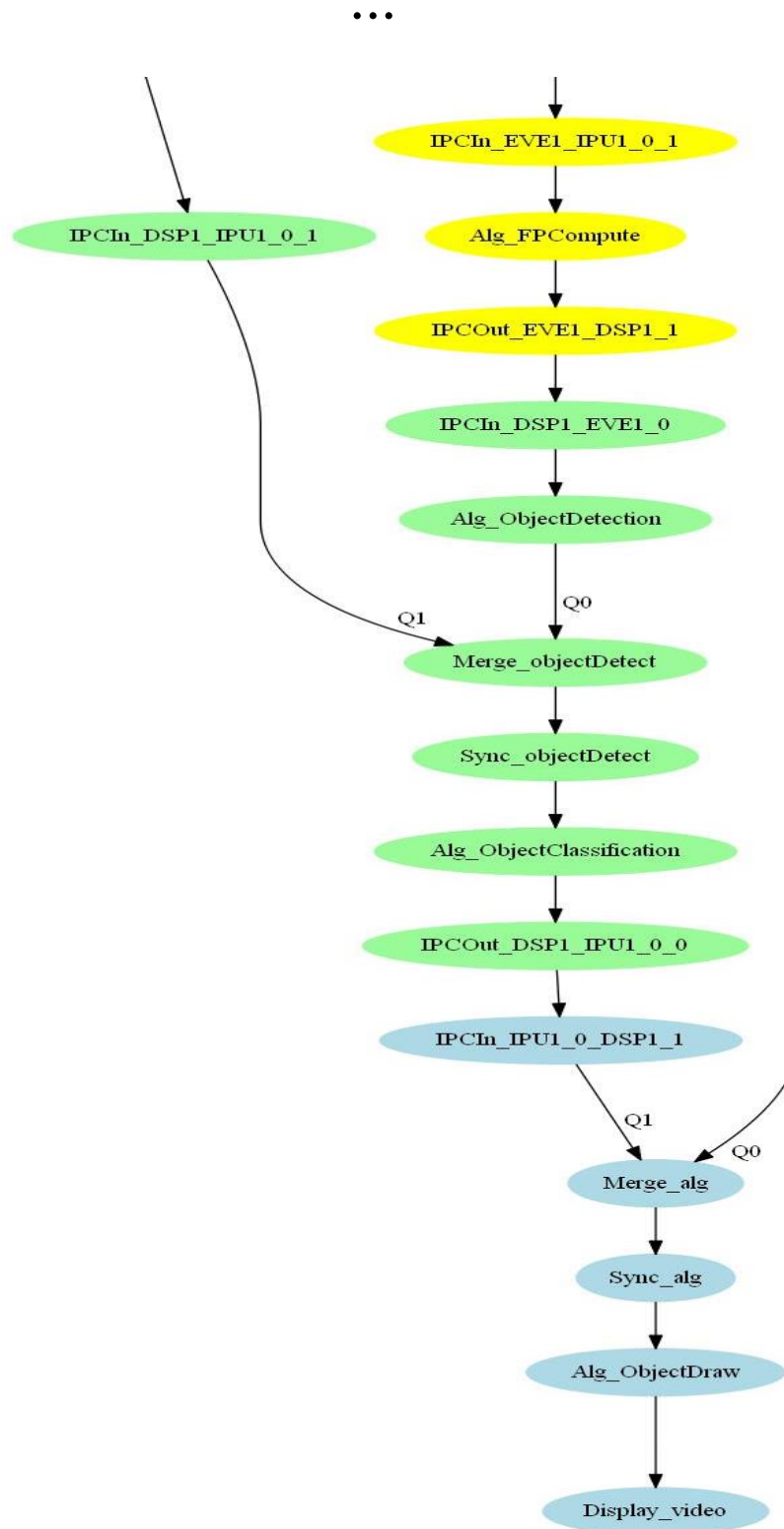
### 3.4. Programsko rješenje za procjenu udaljenosti pješaka koristeći jednu kameru na vozilu

Programsko rješenje za proračun udaljenosti detektiranog pješaka koristeći jednu kameru na vozilu predstavljeno je dijagramom toka slučaja upotrebe koji se može vidjeti na slici 3.17.



...





Sl. 3.17. Slika stvorenog slučaja upotrebe za procjenu udaljenosti pješaka upotrebom jedne kamere

Treći slučaj upotrebe, ujedno i posljednji, napravljen je također radi ispitivanja da li algoritam za detekciju pješaka radi bolje na većoj rezoluciji, odnosno da bi se pokazalo kako se mijenja točnost detekcije i procjene udaljenosti s povećanjem rezolucije. Prethodna dva objašnjena slučaja upotrebe detekciju i proračun udaljenosti rade na manjim rezolucijama, zbog čega se pješaci u tim slučajevima upotrebe ne mogu detektirati na udaljenostima većim od 5 metara. Zato se u ovom slučaju upotrebe detekcija i proračun pješaka izvršavaju na prikazu jedne kamere rezolucije 1280x720 elemenata slike. Svi algoritmi detekcije kao i algoritam proračuna udaljenosti rade na isti princip kao što rade i kod prethodna dva objašnjena slučaja upotrebe, samo s različitim parametrima. Jedino što nedostaje u ovom slučaju upotrebe, u odnosu na prethodna dva objašnjena, je iscrtavanje 2D položaja detektiranih pješaka u okolini vozila. Na slici 3.18 se može vidjeti prikaz dobiven na ekranu kad se pokrene slučaj upotrebe za procjenu udaljenosti pješaka koristeći jednu kameru na vozilu.



Sl. 3.18. Prikaz na ekranu dobiven kad se pokrene slučaj upotrebe za procjenu udaljenosti pješaka koristeći jednu kameru na vozilu

### **3.5. Način pokretanja vlastitog rješenja za iscrtavanje 2D mape položaja pješaka u okolini vozila**

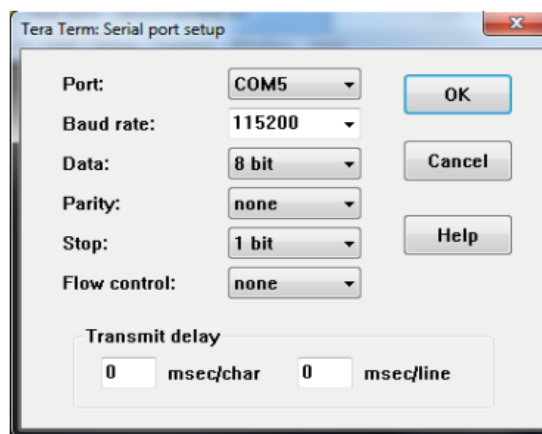
Za pokretanje vlastitog rješenja na ADAS ploči, prije svega je potrebno instalirati VisionSDK okruženje na računalo, te nakon izgradnje slučaja upotrebe treba napraviti nekoliko koraka:

- U komandnoj liniji treba se postaviti u sljedeću mapu  
*C:\VISION\_SDK\_02\_12\_01\_00\vision\_sdk*
- Idući korak je izgradnja niže razine VisionSDK, što se radi sljedećom naredbom  
*gmake -s -j depend*
- Izgradnja *bootloader* – a naredbom:  
*gmake -s -j sbl\_sd*
- Izgradnja VisionSDK dodataka i slučaja upotrebe naredbom:  
*gmake -s -j*
- Izgradnja *image* datoteke naredbom:  
*gmake -s appimage*

Nakon pokrenutih naredbi, potrebno je formatirati SD karticu prema uputama koje su dio VisionSDK okruženja, jer se rješenje na ADAS ploči pokreće preko SD kartice. Idući korak je kopiranje dviju datoteku, MLO i AppImage, na SD karticu, radi pokretanja slučaja upotrebe na ploči. One se nalaze u mapama:

*C:\VISION\_SDK\_02\_12\_01\_00\vision\_sdk\build\scripts\sd\_tda2xx-evm*

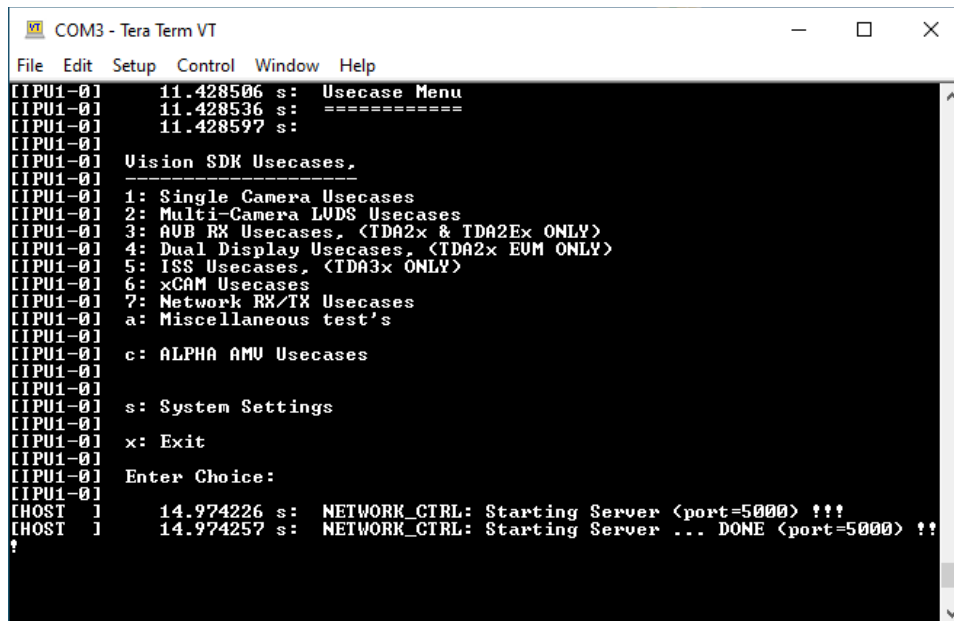
*C:\VISION\_SDK\_02\_12\_01\_00\vision\_sdk\binaries\tda2xx\_evm\_bios\_all\vision\_sdk\bin\tda2xx-evm\sbl\_boot*Za pristup korisničkom sučelju aplikacije koristi se UART – USB pretvarač. Potrebno je instalirati program *TeraTerm* te ga pokrenuti nakon spajanja ploče i računala i podesiti postavke kao na slici 3.19.



Sl. 3.19. Postavke TeraTerm programa

Nakon postavljanja TeraTerm programa, potrebno je ubaciti SD karticu u određeni utor i uključiti napajanje.

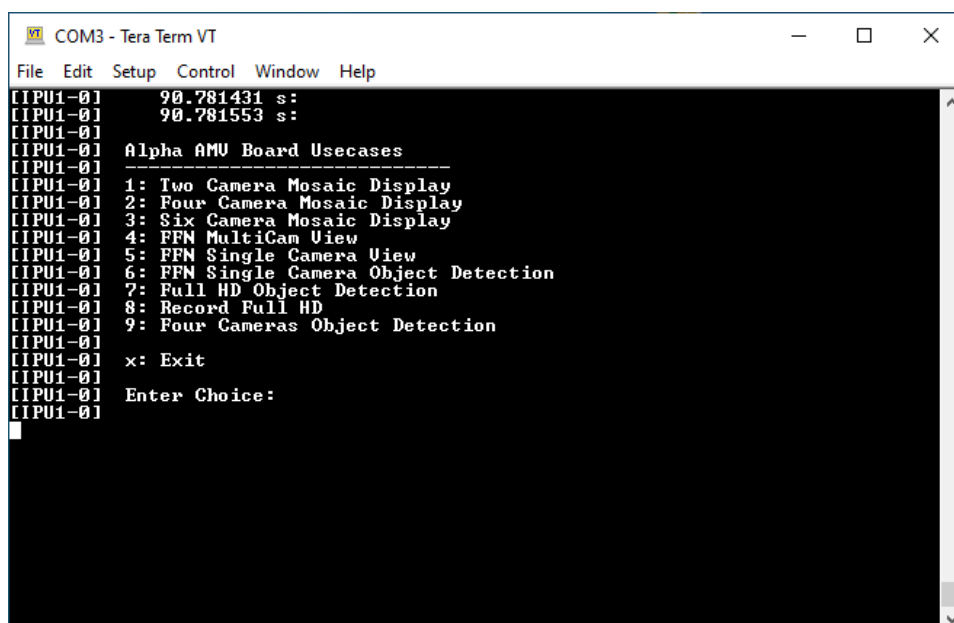
Kada se u TeraTerm – u pojavi izbornik kao na slici 3.20 potrebno je odabrati „ALPHA AMV Usecases“ pritiskom tipke „c“.



```
COM3 - Tera Term VT
File Edit Setup Control Window Help
[IPU1-0] 11.428506 s: Usecase Menu
[IPU1-0] 11.428536 s: =====
[IPU1-0] 11.428597 s:
[IPU1-0]
[IPU1-0] Vision SDK Usecases,
[IPU1-0] -----
[IPU1-0] 1: Single Camera Usecases
[IPU1-0] 2: Multi-Camera LUDS Usecases
[IPU1-0] 3: AUB RX Usecases, <TDA2x & TDA2Ex ONLY>
[IPU1-0] 4: Dual Display Usecases, <TDA2x EUM ONLY>
[IPU1-0] 5: ISS Usecases, <TDA3x ONLY>
[IPU1-0] 6: xCAM Usecases
[IPU1-0] 7: Network RX/TX Usecases
[IPU1-0] a: Miscellaneous test's
[IPU1-0]
[IPU1-0] c: ALPHA AMU Usecases
[IPU1-0]
[IPU1-0] s: System Settings
[IPU1-0]
[IPU1-0] x: Exit
[IPU1-0]
[IPU1-0] Enter Choice:
[IPU1-0]
[HOST 1] 14.974226 s: NETWORK_CTRL: Starting Server <port=5000> !!!
[HOST 1] 14.974257 s: NETWORK_CTRL: Starting Server ... DONE <port=5000> !!!
?
```

Sl. 3.20. Početni izbornik nakon uključivanja TeraTerm – a

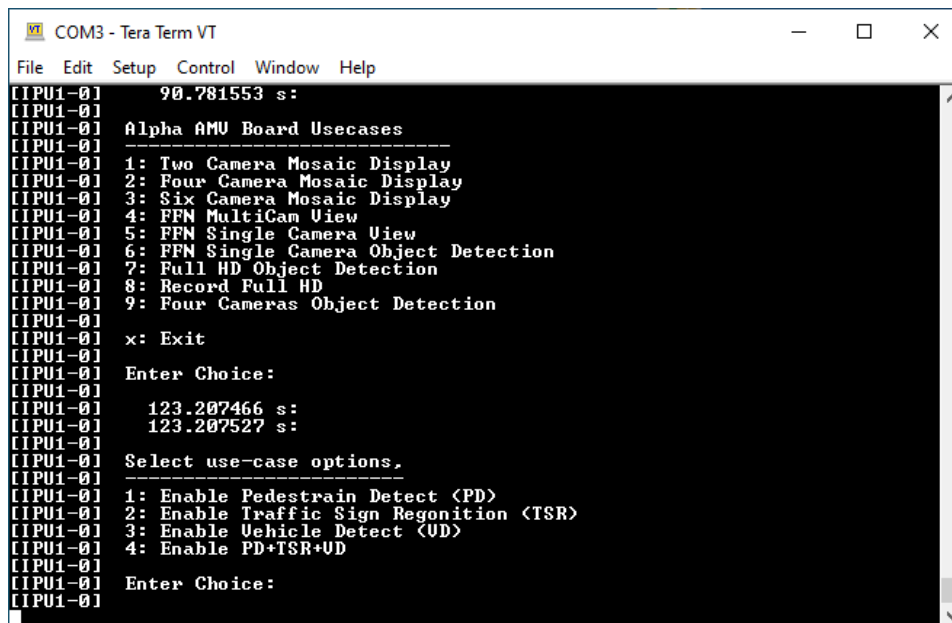
Nakon pritiskanja tipke „c“ otvorit će se drugi izbornik u kojem će se moći pristupiti slučaju upotrebe za iscrtavanje 2D mape položaja pješaka u okolini vozila. Slika 3.21 prikazuje izbornik na kojem se nalazi popis slučaja upotrebe među kojima treba odabrati „Four Cameras Object Detection“ pritiskom tipke „9“.



```
COM3 - Tera Term VT
File Edit Setup Control Window Help
[IPU1-0] 90.781431 s:
[IPU1-0] 90.781553 s:
[IPU1-0]
[IPU1-0] Alpha AMU Board Usecases
[IPU1-0] -----
[IPU1-0] 1: Two Camera Mosaic Display
[IPU1-0] 2: Four Camera Mosaic Display
[IPU1-0] 3: Six Camera Mosaic Display
[IPU1-0] 4: FFM MultiCam View
[IPU1-0] 5: FFM Single Camera View
[IPU1-0] 6: FFM Single Camera Object Detection
[IPU1-0] 7: Full HD Object Detection
[IPU1-0] 8: Record Full HD
[IPU1-0] 9: Four Cameras Object Detection
[IPU1-0]
[IPU1-0] x: Exit
[IPU1-0]
[IPU1-0] Enter Choice:
[IPU1-0]
```

Sl. 3.21. Izbornik među kojima se nalazi razvijeni slučaj upotrebe pod rednim brojem 9

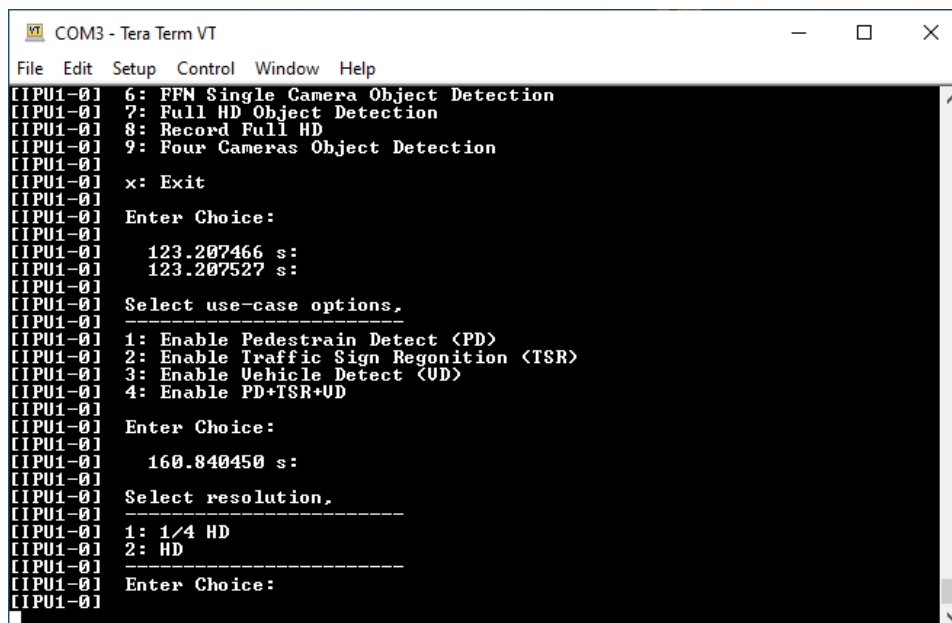
Pritiskom tipke „9“ otvara se izbornik kao na slici 3.22 u kojem se treba odabrati vrsta objekata koje algoritam detektira, pješake, vozila ili prometne znakove. U ovom slučaju treba se odabrati „Enable Pedestrian Detect <PD>“ pritiskom na tipku „1“.



```
COM3 - Tera Term VT
File Edit Setup Control Window Help
[IPU1-0] 90.781553 s:
[IPU1-0] Alpha AMU Board Usecases
[IPU1-0] -----
[IPU1-0] 1: Two Camera Mosaic Display
[IPU1-0] 2: Four Camera Mosaic Display
[IPU1-0] 3: Six Camera Mosaic Display
[IPU1-0] 4: FFN MultiCam View
[IPU1-0] 5: FFN Single Camera View
[IPU1-0] 6: FFN Single Camera Object Detection
[IPU1-0] 7: Full HD Object Detection
[IPU1-0] 8: Record Full HD
[IPU1-0] 9: Four Cameras Object Detection
[IPU1-0] x: Exit
[IPU1-0] Enter Choice:
[IPU1-0] 123.207466 s:
[IPU1-0] 123.207527 s:
[IPU1-0] Select use-case options,
[IPU1-0] -----
[IPU1-0] 1: Enable Pedestrian Detect <PD>
[IPU1-0] 2: Enable Traffic Sign Recognition <TSR>
[IPU1-0] 3: Enable Vehicle Detect <UD>
[IPU1-0] 4: Enable PD+TSR+UD
[IPU1-0] Enter Choice:
```

Sl. 3.22. Izbornik u kojem se odabiru pješaci kao kategorija objekata za detekciju pritiskom tipke „1“

Idući izbornik nudi odabir rezolucije kamere kao na slici 3.23.

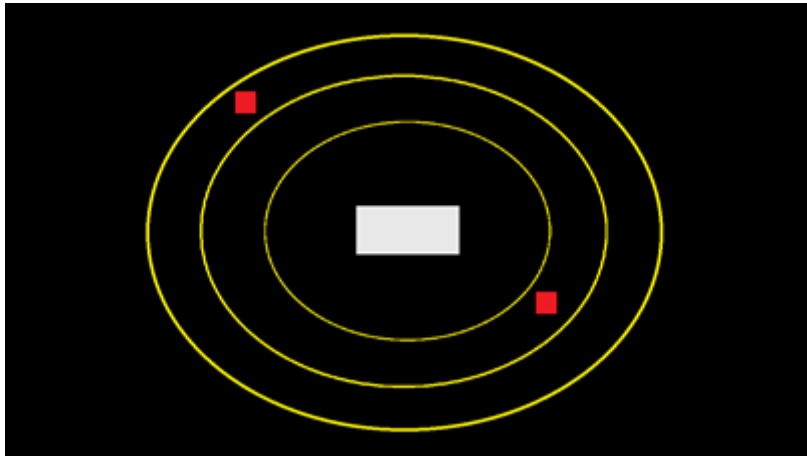


```
COM3 - Tera Term VT
File Edit Setup Control Window Help
[IPU1-0] 6: FFN Single Camera Object Detection
[IPU1-0] 7: Full HD Object Detection
[IPU1-0] 8: Record Full HD
[IPU1-0] 9: Four Cameras Object Detection
[IPU1-0] x: Exit
[IPU1-0] Enter Choice:
[IPU1-0] 123.207466 s:
[IPU1-0] 123.207527 s:
[IPU1-0] Select use-case options,
[IPU1-0] -----
[IPU1-0] 1: Enable Pedestrian Detect <PD>
[IPU1-0] 2: Enable Traffic Sign Recognition <TSR>
[IPU1-0] 3: Enable Vehicle Detect <UD>
[IPU1-0] 4: Enable PD+TSR+UD
[IPU1-0] Enter Choice:
[IPU1-0] 160.840450 s:
[IPU1-0] Select resolution,
[IPU1-0] -----
[IPU1-0] 1: 1/4 HD
[IPU1-0] 2: HD
[IPU1-0] Enter Choice:
```

Sl. 3.23. Izbornik u kojem se pritisca tipka „1“ za detekciju pješaka na prikazu s četiriju kamera

Pritiskom na tipku „2“ dobit će se prikaz samo jedne kamere na ekranu, dok za konačno rješenje slučaja upotrebe treba pritisnuti tipku „1“, kako bi se detekcija pješaka izvršavala na prikazu s četiriju kamera, te se iscrtala 2D mapa s položajem pješaka u okolini vozila.

Na kraju se dobije 2D mapa kao na slici 3.24, gdje je u ovom slučaju jedan pješak detektiran prednjom i desnom kamerom na udaljenosti od 1.5 m, a drugi je detektiran zadnjom i lijevom kamerom na udaljenosti od 2.5 m.



Sl. 3.24. 2D mapa slučaja upotrebe za iscrtavanje položaja pješaka u okolini vozila

## **4. TESTIRANJE RADA PREDLOŽENOG RJEŠENJA ZA ISCRTAVANJE 2D MAPE POLOŽAJA PJEŠAKA U OKOLINI VOZILA KORISTEĆI VIŠE KAMERA**

U ovom poglavlju detaljno je opisano testiranje ispravnosti rada predloženog rješenja za iscrtavanje 2D mape položaja pješaka u okolini vozila koristeći četiri kamera. S obzirom na to da predloženo rješenje podrazumijeva izvršavanje detekcije, procjene udaljenosti i iscrtavanje 2D mape položaja pješaka na osnovu obrade okvira rezolucije 640x360 elemenata slike s četiriju pojedinačnih kamera, načinjena je evaluacija dobivenih rezultata za dodatna dva slučaja upotrebe, kod kojih se obrada izvršava na okvirima većih rezolucija nego na okvirima predloženog konačnog rješenja.

Prije evaluacije predloženog rješenja za iscrtavanje 2D mape položaja pješaka u okolini vozila, testirani su odziv i preciznost dostupnog algoritma za detekciju pješaka, koji je dio VisionSDK razvojnog okruženja, i kao takav nije bio dio samog zadatka ovog rada (tj. korišteno je gotovo dostupno rješenje za detekciju pješaka). Testiranja preciznosti i odziva algoritma detekcije pješaka provedena su korištenjem Kitti [27] baze video signala snimanih u stvarnom vremenu. Baza videa, koja je preuzeta s interneta, sadrži velik broj video signala vožnje po gradskim ulicama i to vozilima u kojima je kamera bila postavljena te je sniman gradski promet u kojima se nalaze druga vozila i pješaci.

Nakon analize odziva i preciznosti algoritma za detekciju pješaka, izvršena je prvo evaluacija procjene udaljenosti detektiranih pješaka na različitim udaljenostima od kamere pokretanjem različitih slučajeva upotrebe, koji se prvenstveno razlikuju po rezoluciji okvira video signala s kamera, koji trebaju biti obrađeni.

Na kraju, zadnje testiranje je izvršeno na algoritmu za iscrtavanje 2D mape položaja pješaka, koji se nalaze u okolini vozila. Tu se testirao prelazak pješaka iz područja kojeg hvata jedna kamera u područje koje hvata druga kamera. Evaluacije je načinjena upotrebom dvaju slučajeva upotrebe, koji se razlikuju po rezoluciji okvira video signala s kamera, koji trebaju biti obrađeni.

### **4.1. Testiranje performansi algoritma za detekciju pješaka**

Prva testiranja koja su izvršena bila su testiranja gotovog (dobivenog, koji je dio VisionSDK okruženja) algoritma za detekciju pješaka, na kojem se zasniva konačno rješenje ovog

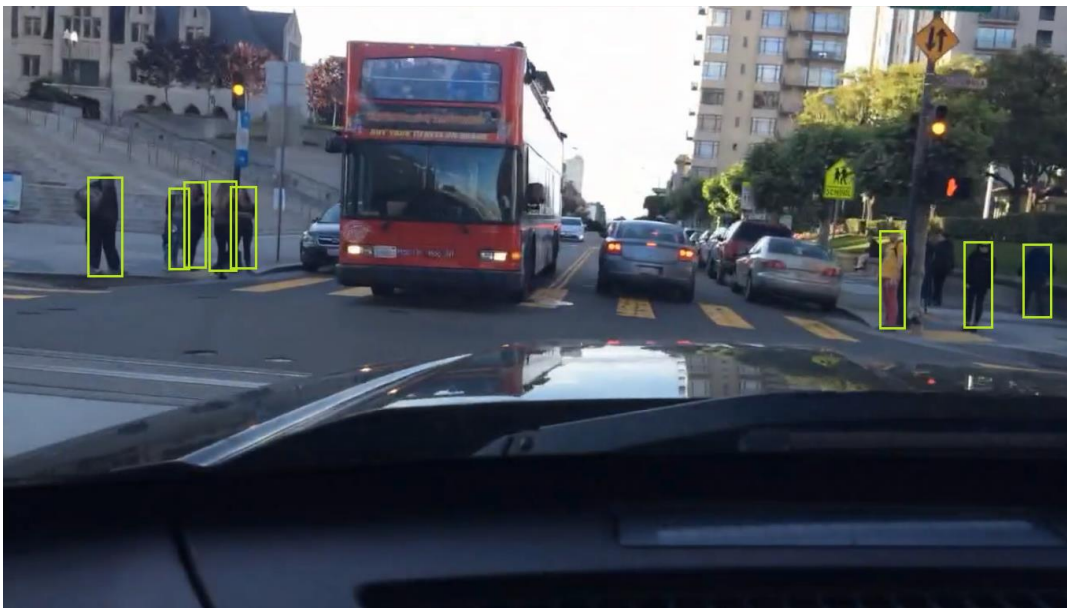
rada. Preciznost procjene udaljenosti i iscrtavanje 2D mape položaja pješaka u okolini vozila izravno ovise o ispravnosti algoritma za detekciju pješaka. Za testiranje ispravnosti algoritma detekcije pješaka korištena je Kitti [27] baza video signala snimanih u stvarnom vremenu . Video signali sadrže snimke vožnje po gradskim ulicama vozilima u kojima je kamera bila postavljena te je sniman gradski promet u kojima se nalaze druga vozila i pješaci. Odabrano je 16 videa u kojima se nalaze pješaci koji šetaju ulicama gradova. Svaki od testnih videa ima rezoluciju 1280x720 elemenata slike, uz 30 okvira po sekundi. Cilj testiranja je provjeriti performanse algoritma za detekciju pješaka u vidu odziva i preciznosti. U tablici 4.1 mogu se vidjeti rezultati odziva i preciznosti algoritma za detekciju pješaka za 16 korištenih video signala.

Tablica 4.1. Rezultati testiranja ispravnosti algoritma za detekciju pješaka

	BROJ PJEŠAKA	TOČNA DETEKCIJA (TP)	BROJ NEDETEKTIRANIH PJEŠAKA (FN)	POGREŠNA DETEKCIJA (FP)	ODZIV	PRECIZNOST
Video1	12	11	1	4	91,67%	73,33%
Video2	11	9	2	1	81,82%	90,00%
Video3	5	4	1	12	80,00%	25,00%
Video4	3	1	2	4	33,33%	20,00%
Video5	1	1	0	10	100,00%	9,09%
Video6	2	1	1	2	50,00%	33,33%
Video7	6	6	0	9	100,00%	40,00%
Video8	1	1	0	7	100,00%	12,50%
Video9	14	12	2	13	85,71%	48,00%
Video10	6	5	1	6	83,33%	45,45%
Video11	7	5	2	3	71,43%	62,50%
Video12	7	5	2	8	71,43%	38,46%
Video13	6	5	1	9	83,33%	35,71%
Video14	6	6	0	5	100,00%	54,55%
Video15	3	2	1	0	66,67%	100,00%
Video16	3	2	1	8	66,67%	20,00%
<b>Ukupno</b>	<b>93</b>	<b>76</b>	<b>17</b>	<b>101</b>	<b>79,09%</b>	<b>44,25%</b>



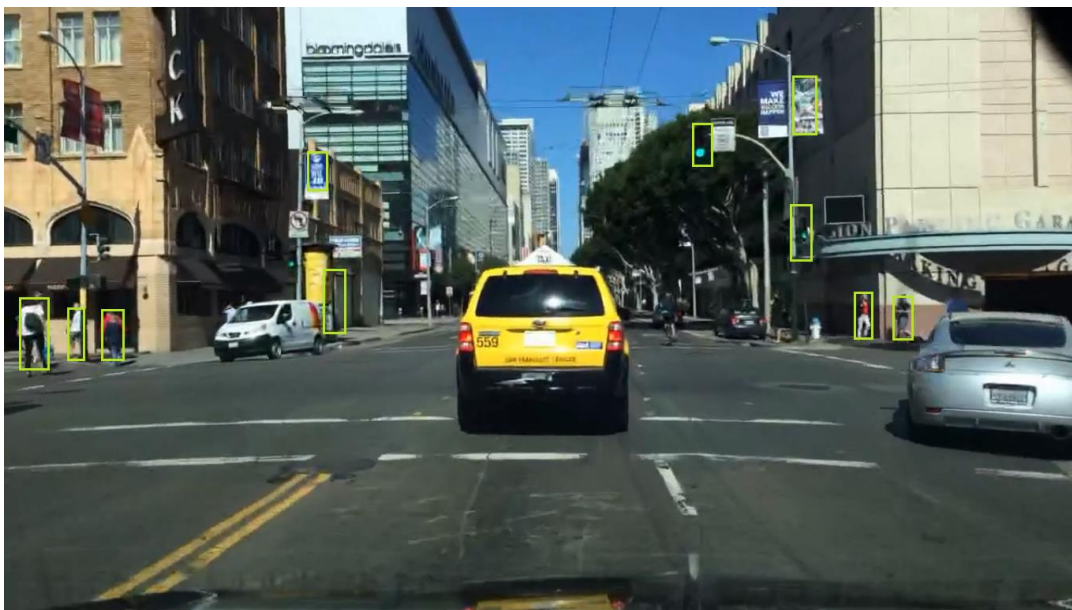
Iz rezultata se može vidjeti da prosječan postotak odziva algoritma za detekciju pješaka iznosi 79.09%. Međutim, u nekim videima postotak odziva je jako nizak i iznosi oko 30%. To se događa kada je osvjetljenje u video signalu loše pa je detektoru teško prepoznati pješaka. Većina video signala sadrži velik broj pješaka, koji trebaju biti detektirani, a vozilo unutar kojeg se nalazi kamera koja snima gradski promet, ponekad se kreće većom brzinom, tako da je često i čovjeku teško uočiti sve pješake, koji se nalaze u video signalu. S obzirom na loša osvjetljenja u određenim video signalima i dinamične uvjete unutar kojih algoritam treba izvršavati detekciju, prosječan postotak odziva algoritma za detekciju je zadovoljavajući. Na slici 4.1 može se vidjeti okvir kada algoritam za detekciju pješaka ima zadovoljavajući odziv.



Sl. 4.1. Okvir videa na kojemu je odziv algoritma za detekciju pješaka zadovoljavajući

Što se tiče preciznosti, iz tablice se može vidjeti da je postotak prosječne preciznosti prilično nizak te da iznosi 44.25%. Gotovo je u svim video signalima postotak preciznosti nizak, osim u dva, od kojih u jednom postotak preciznosti dostiže 73.33%, a u drugom čak i 90%. Preciznost se odnosi na omjer broja točno detektiranih pješaka i ukupnog broja detekcija, što uključuje bilo koju detekciju. Iz podataka o preciznosti algoritma za detekciju pješaka može se zaključiti da algoritam prilično loše radi u smislu da ima priličan broj lažno pozitivnih detekcija, što će kasnije dovesti do poteškoća prilikom iscrtavanja 2D mape položaja pješaka u okolini vozila. U videima se primijeti da algoritam, pored pješaka u videu, često detektira bandere, drveća, semafore, oblake itd. kao pješake. To kasnije dovodi do pojave grešaka prilikom iscrtavanja 2D mape položaja pješaka. Na primjer, može se dogoditi da se na slikama s dvije susjedne kamere nalazi isti pješak u područjima isith težina na određenoj udaljenosti, na primjer 1 m, ali pješak

bude detektiran samo na jednoj slici s kamere, a na drugoj slici s kamere umjesto pješaka bude detektirano drvo koje se nalazi u pozadini na procijenjenoj udaljenosti na primjer 3 m. Ovakva neprecizna detekcija drveta kao pješaka, unijet će pogrešku u krajnje iscrtavanje 2D položaja pješaka. U ovom primjeru, na 2D mapi će se iscrtati položaj pješaka na udaljenosti od 2 m (aritmetička sredina dva detektirana objekta na slikama sa susjednih kamera u područjima istih težina). Algoritam za detekciju pješaka nije robustan na loše osvjetljenje i dinamične uvjete, u kojima se detekcija izvršava, što dovodi do jako niskog postotka preciznosti. Na slici 4.2 može se vidjeti okvir video signala, u kojem su pored detektiranih pješaka detektirani i objekti koji nisu pješaci.



Sl. 4.2. Okvir videa u kojem su pored detektiranih pješaka detektirani i objekti koji nisu pješaci

Ovim se testiranjem željelo ispitati kakve su performanse algoritma za detekciju pješaka, čiji rezultati izravno utječu na sve ono što dolazi nakon tog koraka u konačnom rješenju za iscrtavanje položaja pješaka na 2D mapi. Sami video signali korišteni u ovom dijelu testiranja nisu korišteni u ostalim testiranjima opisanim ispod, jer nisu istovremeno snimani s dvije kamere smještene na različitim mjestima (stranama) u automobilu. Cilj je ovog testiranja samo bio utvrditi performanse rada algoritma za detekciju objekata na njemu dotad nepoznatom skupu podataka.

## 4.2. Testiranje ispravnosti rada algoritma za procjenu udaljenosti

Nakon evaluacije rezultata odziva i preciznosti algoritma za detekciju pješaka na ranije spomenutom skupu podataka, uslijedilo je testiranje algoritma za procjenu udaljenosti, zasnovanog na prethodno opisanom algoritmu za detekciju pješaka. Što je granični okvir oko samog detektiranog pješaka precizniji, to je procijenjena udaljenosti pješaka točnija. S obzirom na to da loše osvjetljenje nepovoljno djeluje na detekciju, procjena udaljenosti detektiranog pješaka često odstupa od stvarnih vrijednosti. U nastavku potpoglavlja bit će opisano proračunavanje udaljenosti detektiranog pješaka unutar različitih slučajeva upotrebe, koji se prvenstveno razlikuju po rezoluciji okvira video signala snimanih u stvarnom vremenu nad kojim se vrši detekcija i post obradi, koji uključuje procjenu udaljenosti detektiranog pješaka. Svi video signali, neovisno o rezoluciji sadrže 30 okvira po sekundi. Procjena udaljenosti računata je upotrebom dviju metoda, koje su objašnjene u trećem poglavlju, u dijelovima 3.2.3. i 3.2.4.. Za obje metode bila je potrebna informacija o prosječnoj visini čovjeka, koja u Europi iznosi 1.77 m [26]. S obzirom na to da pješaci nisu iste visine, testiranje ispravnosti rada algoritma za procjenu udaljenosti je izvršeno na pet osoba različitih visina. U tablici 4.2 se mogu vidjeti visine osoba, koje su sudjelovale u testiranju algoritma za procjenu udaljenosti detektiranog pješaka.

Tablica 4.2. Visine osoba, koje su učestvovalе u testiranju algoritma za procjenu udaljenosti

	Visina osobe [m]
Osoba 1	1.63
Osoba 2	1.74
Osoba 3	1.79
Osoba 4	1.90
Osoba 5	1.96

Testiranja su vršena u sunčanim vremenskim uvjetima, što znači da je bilo puno osvjetljenja i područja izmjene sunca i hlada, što je dodatno nepovoljno utjecalo na rezultate procjene udaljenosti detektiranih pješaka. Kredom su ocrtane pozicije udaljenosti od kamere. Prva pozicija koja je ocrтана je 1 metar udaljena od kamere. Iduće su udaljenije po pola metra i tako sve do 10 metara. Rezultati rada algoritma za procjenu udaljenosti detektiranih pješaka (tj. njegova pogreška), izraženi su kao korijeni srednje kvadratne pogreške (engl. *Root – mean – square – error – RMSE*). Video materijali snimki nastalih prilikom testiranja algoritma za procjenu udaljenosti nalaze se u elektroničkom prilogu ovom radu P.4.1..

#### 4.2.1. Procjena udaljenosti detektiranih pješaka unutar okvira rezolucije 640x360 elemenata slike

Prvo testiranje algoritma za procjenu udaljenosti detektiranih pješaka, izvršeno je pokretanjem prvog slučaja upotrebe opisanog u potpoglavlju 3.2., odnosno slučaja upotrebe konačnog rješenja, tj. nad video signalima snimanim u stvarnom vremenu okvira rezolucije 640x360 elemenata slike i uz 30 okvira u sekundi. Okviri video signala nad kojima je vršena detekcija male su rezolucije, pa zbog toga postoji mogućnost manje uspješne detekcije i proračuna udaljenosti na velikim udaljenostima od vozila. U tablici 4.3 mogu se vidjeti vrijednosti RMSE pri procjeni udaljenosti u video signalima rezolucije 640x360 elemenata slike za svaku od metoda (metoda 1 opisana u dijelu 3.2.3., a metoda 2 opisana u dijelu 3.2.4.) na različitim udaljenostima.

Tablica 4.3. RMSE algoritma za procjenu udaljenosti detektiranih pješaka u video signalima rezolucije 640x360 elemenata slike za svaku od metoda procjene udaljenosti na različitim udaljenostima

	Udaljenost 1m		Udaljenost 1.5m		Udaljenost 2m		Udaljenost 2.5m	
	Metoda 1	Metoda 2	Metoda 1	Metoda 2	Metoda 1	Metoda 2	Metoda 1	Metoda 2
	[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]
Osoba 1	1.12	1.2	1.57	1.68	1.85	1.97	2.05	2.19
Osoba 2	1.06	1.14	1.58	1.69	2.0	2.14	2.02	2.16
Osoba 3	1.01	1.09	1.48	1.58	1.90	2.04	2.52	2.70
Osoba 4	0.99	1.06	1.45	1.56	1.9	2.04	2.23	2.39
Osoba 5	0.97	1.04	1.55	1.66	1.85	1.99	2.05	2.19
RMSE	6.18%	12.07%	5.78%	14.43%	11.40%	7.01%	37.65%	26.83%

Iz dobivenih rezultata može se vidjeti da metoda 1 ima manji RMSE na manjim udaljenostima od metode 2. Na udaljenostima preko 2 metra, metoda 2 ima manji RMSE. Međutim, na udaljenostima iznad 2 metra, ni jedna od metoda nema zadovoljavajuće rezultate procjena udaljenosti. Razlog tome je niska rezolucija okvira video signala na kojima se izvršava detekcija objekata, uz svakako loše osvjetljenje, na koje detektor nije robustan. Na udaljenostima iznad 2.5 metra, na ovoj rezoluciji detektor ne prepoznaje pješake pa stoga u tablici nisu ni dani rezultati procjene udaljenosti na udaljenostima većim od 2.5 metara. Rezultati pokazuju da procijenjene udaljenosti kod osobe 3, koja je visoka 1.79 metara, su najbliže stvarnim udaljenostima, što je očekivano jer parametar prosječne visine osobe iznosi 1.77 metara [26], te je najbliži visini osobe 3, u odnosu na visine ostalih osoba. S obzirom na to da algoritam za detekciju pješaka i sami rezultati procijenjenih udaljenosti nisu zadovoljavajući nakon obrade video signala rezolucije 640x360 elemenata slike, provedena su dodatna testiranja na većim rezolucijama.

Na slici 4.3 se mogu vidjeti neke od procijenjenih udaljenosti na okvirima rezolucije 640x360 elemenata slike.



a)

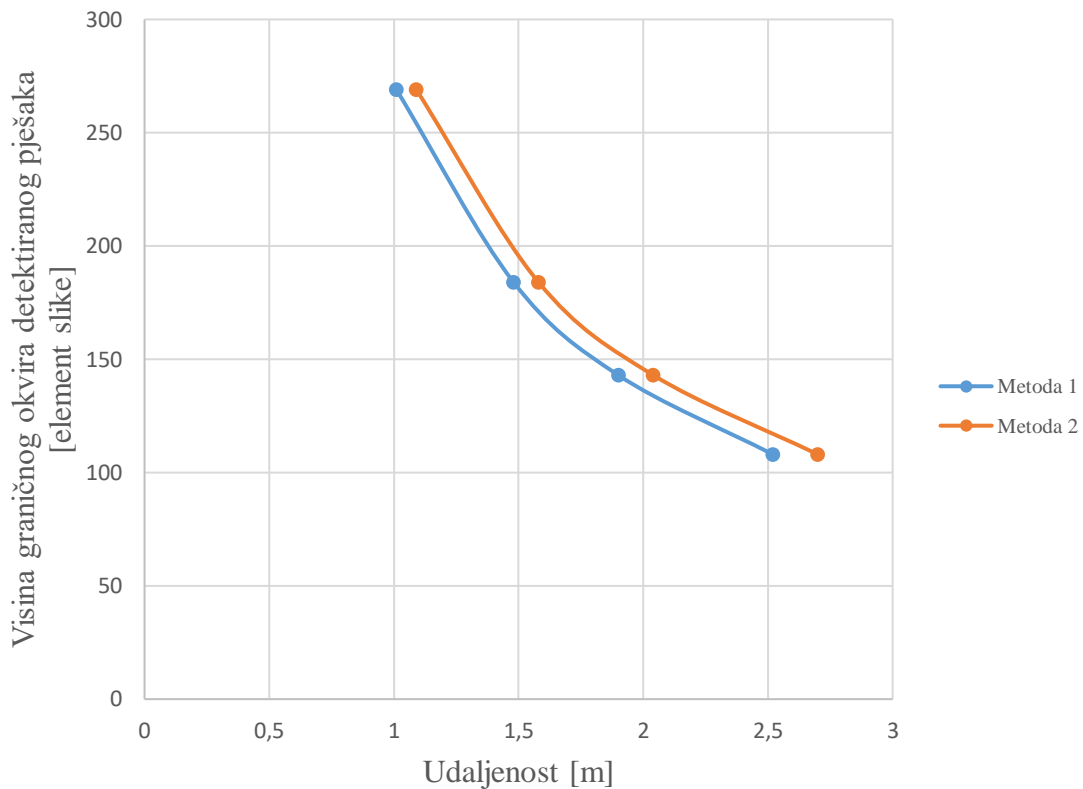


b)

Sl. 4.3. Procijenjene udaljenosti na okvirima 640x360 elemenata slike na (a) 1 metar i na (b) 2 metra stvarne udaljenosti za osobu 3

Iz tablice 4.3. se može vidjeti da s povećanjem udaljenosti rezultati procjene udaljenosti imaju veću pogrešku. To je zato što pogreška od jednog elementa slike u samom graničnom okviru

unosu veću pogrešku u procjenu udaljenosti. Na slici 4.4 može se vidjeti graf odnosa visine graničnog okvira detektiranog pješaka i procijenjene udaljenosti za obje metode za osobu 3.



Sl. 4.4. Graf odnosa visine graničnog okvira detektiranog pješaka i procijenjene udaljenosti za okvir rezolucije 640x360 elemenata slike za osobu 3

Na video signalu koji se nalazi u elektroničkom prilogu ovom radu P.4.1.1. može se vidjeti kako radi algoritam za procjenu udaljenosti na okvirima rezolucije 640x360 elemenata slike.

#### 4.2.1. Procjena udaljenosti detektiranih pješaka unutar okvira rezolucije 960x540 elemenata slike

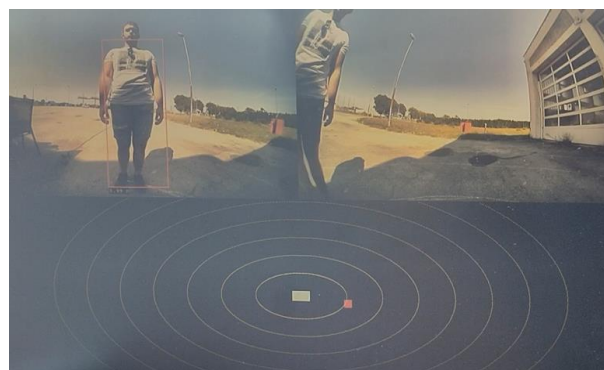
Nakon izvršenog testiranja procjene udaljenosti pješaka na video signalima rezolucije 640x360 elemenata slike, testiranje je ponovljeno nad drugim slučajem upotrebe, opisanim u potpoglavlju 3.3., u istim uvjetima i s istim osobama, samo s video signalima snimanim u stvarnom vremenu rezolucije 960x540 elemenata slike uz 30 okvira u sekundi.

U tablici 4.4 mogu se vidjeti vrijednosti RMSE procjena udaljenosti detektiranih pješaka u video signalima rezolucije 960x540 elemenata slike za svaku od metoda na različitim udaljenostima.

Tablica 4.4. RMSE algoritma za procjenu udaljenosti detektiranih pješaka u video signalima rezolucije 960x540 elemenata slike za svaku od metoda procjene udaljenosti na različitim udaljenostima

	Udaljenost 1m		Udaljenost 1.5m		Udaljenost 2m		Udaljenost 2.5m	
	Metoda	Metoda	Metoda	Metoda	Metoda	Metoda	Metoda	Metoda
	1 [m]	2 [m]	1 [m]	2 [m]	1 [m]	2 [m]	1 [m]	2 [m]
Osoba 1	1.11	1.19	1.49	1.59	2.36	2.53	2.52	2.7
Osoba 2	1.05	1.12	1.51	1.62	2.03	2.18	2.49	2.67
Osoba 3	1.05	1.12	1.55	1.66	2.0	2.15	2.54	2.72
Osoba 4	1.04	1.11	1.43	1.53	1.86	1.99	2.23	2.39
Osoba 5	1.19	1.27	1.52	1.63	2.0	2.15	2.58	2.72
RMSE	10.47%	17.31%	4.00%	11.48%	17.33%	26.77%	12.76%	20.11%
	Udaljenost 3m		Udaljenost 3.5m		Udaljenost 4m		Udaljenost 4.5m	
	Metoda	Metoda	Metoda	Metoda	Metoda	Metoda	Metoda	Metoda
	1 [m]	2 [m]	1 [m]	2 [m]	1 [m]	2 [m]	1 [m]	2 [m]
Osoba 1	3.07	3.29	3.49	3.74	4.0	4.29	4.35	4.66
Osoba 2	3.05	3.26	3.52	3.77	4.04	4.33	4.34	4.66
Osoba 3	3.03	3.25	3.49	3.74	4.01	4.29	4.30	4.61
Osoba 4	2.86	3.06	3.35	3.59	4.0	4.29	4.09	4.38
Osoba 5	3.07	3.29	3.43	3.68	3.85	4.13	4.39	4.71
RMSE	8.10%	24.57%	7.48%	21.38%	6.97%	27.50%	23.16%	15.61%
	Udaljenost 5m							
	Metoda	Metoda						
	1 [m]	2 [m]						
Osoba 1	4.35	4.66						
Osoba 2	4.34	4.66						
Osoba 3	4.49	4.81						
Osoba 4	4.21	4.51						
Osoba 5	4.35	4.66						
RMSE	65.80%	35.30%						

Iz dobivenih rezultata može se vidjeti da do 4 metra udaljenosti od kamere na vozilu, metoda 1 ima manji RMSE od metode 2. Na udaljenostima većim od 4 metra RMSE je velik i za jednu i za drugu metodu jer granični okvir detektiranih pješaka nije statičan i precizan. U tablici se može vidjeti da su rezultati procijenjenih udaljenosti najbliži stvarnima kod druge (visoka je 1.74 metra) i treće (visoka je 1.79 metara) osobe, što se i očekivalo jer su visine tih osoba najbliže prosječnoj visini ljudi u Europi, koja iznosi 1.77 metara [26]. U usporedbi s prethodnim testiranjem, gdje je okvir video signala snimanog u stvarnom vremenu bio rezolucije 640x360 elemenata slike i gdje je detektor uspješno detektirao pješake na slici s kamere koji su udaljeni do 2.5 metara od kamere, rezultati testiranja gdje je okvir rezolucije 960x540 elemenata slike mogli su se analizirati do 5 metara udaljenosti. Detektor na rezoluciji 960x540 elemenata slike ne detektira pješake na slici s kamere na udaljenostima većim od 5 metara od kamere na vozilu. Ali bez obzira na mogućnost detektiranja pješaka na slici s kamere, koji se nalaze na udaljenostima do 5 metara, RMSE procijenjenih udaljenosti do 2.5 metara su manji kod testiranja provedenih na video signalima rezolucije 640x360 elemenata slike nego kod testiranja provedenih na video signalima 960x540 elemenata slike, te se iz toga da zaključiti da rezolucija na malim udaljenostima nema značajan utjecaj na procjenu udaljenosti. Na slici 4.5 se mogu vidjeti neke od procijenjenih udaljenosti osobe 3 na okvirima rezolucije 960x540 elemenata slike.



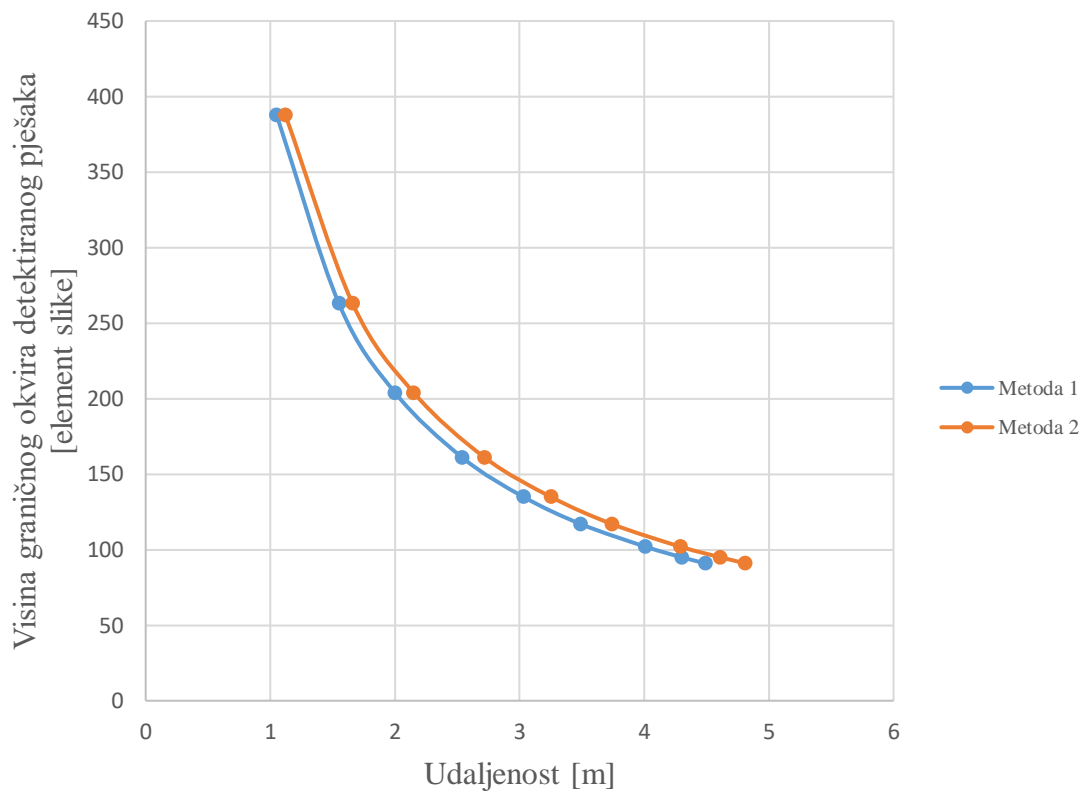
a)



b)



Slika 4.5 prikazuje graf odnosa visine graničnog okvira detektiranog pješaka i procijenjene udaljenosti za obje metode za osobu 3.



Sl. 4.6. Graf odnosa visine graničnog okvira detektiranog pješaka i procijenjene udaljenosti za okvir rezolucije 960x540 elemenata slike za osobu 3

Iz grafa se može vidjeti kako je promjena visine graničnog okvira detektiranog pješaka između malih udaljenosti veća, dok je promjena visine graničnog okvira detektiranog pješaka između većih udaljenosti manja, što znači da je pogreška prilikom procjene udaljenosti pješaka, koji se nalazi na većim udaljenostima, veća jer jedan element slike, kod procjene većih udaljenosti, unosi veću pogrešku. Na video signalu koji se nalazi u elektroničkom prilogu ovom radu P.4.1.2. može se vidjeti kako radi algoritam za procjenu udaljenosti na okvirima rezolucije 960x540 elemenata slike.

#### 4.2.2. Procjena udaljenosti detektiranih pješaka unutar okvira rezolucije 1280x720 elemenata slike

Posljednje testiranje algoritma za procjenu udaljenosti detektiranih pješaka na slikama s kamere izvršeno je pokretanjem trećeg slučaja upotrebe, opisanog u potpoglavlju 3.4. Testirano je u istim uvjetima s istim osobama, samo s video signalima snimanim u stvarnom vremenu rezolucije 1280x720 elemenata slike uz 30 okvira u sekundi. U tablici 4.5 mogu se vidjeti RMSE procjene udaljenosti detektiranih pješaka u video signalima rezolucije 1280x720 elemenata slike za svaku od metoda na različitim udaljenostima.

Tablica 4.5. RMSE algoritma za procjenu udaljenosti detektiranih pješaka u video signalima rezolucije 1280x720 elemenata slike za svaku od metoda procjene udaljenosti na različitim udaljenostima

	Udaljenost 1m		Udaljenost 1.5m		Udaljenost 2m		Udaljenost 2.5m	
	Metoda	Metoda	Metoda	Metoda	Metoda	Metoda	Metoda	Metoda
	1 [m]	2 [m]	1 [m]	2 [m]	1 [m]	2 [m]	1 [m]	2 [m]
Osoba 1	1.05	1.12	1.61	1.73	2.03	2.18	2.49	2.67
Osoba 2	1.07	1.15	1.50	1.61	1.85	1.99	2.23	2.28
Osoba 3	1.03	1.10	1.52	1.63	2.0	2.15	2.52	2.70
Osoba 4	1.01.	1.09	1.51	1.62	1.85	1.98	2.37	2.55
Osoba 5	1.01	1.08	1.49	1.60	1.89	2.03	2.49	2.68
RMSE	4.12%	11.08%	5.04%	14.58%	8.45%	10.61%	17.57%	17.45%
	Udaljenost 3m		Udaljenost 3.5m		Udaljenost 4m		Udaljenost 4.5m	
	Metoda	Metoda	Metoda	Metoda	Metoda	Metoda	Metoda	Metoda
	1 [m]	2 [m]	1 [m]	2 [m]	1 [m]	2 [m]	1 [m]	2 [m]
Osoba 1	3.34	3.58	3.58	3.84	4.03	4.33	4.46	4.78
Osoba 2	2.92	3.14	3.51	3.76	4.0	4.29	4.57	4.90
Osoba 3	3.08	3.3	3.49	3.74	4.09	4.39	4.46	4.78
Osoba 4	2.82	3.02	3.42	3.87	4.09	4.39	4.32	4.63
Osoba 5	3.07	3.30	3.51	3.76	4.07	4.36	4.5	4.82
RMSE	18.20%	32.75%	5.12%	29.84%	6.63%	35.41%	11.36%	29.53%
	Udaljenost 5m		Udaljenost 5.5m		Udaljenost 6m		Udaljenost 6.5m	
	Metoda	Metoda	Metoda	Metoda	Metoda	Metoda	Metoda	Metoda
	1 [m]	2 [m]	1 [m]	2 [m]	1 [m]	2 [m]	1 [m]	2 [m]
Osoba 1	4.95	5.31	5.55	5.96	6.05	6.49	6.48	6.95
Osoba 2	4.9	5.26	5.55	5.96	5.98	6.41	6.56	7.04
Osoba 3	5.09	5.46	5.45	5.84	5.99	6.42	6.56	7.03
Osoba 4	4.69	5.0	5.44	5.84	6.05	6.49	6.56	7.03
Osoba 5	4.99	5.36	5.61	6.02	6.05	6.49	6.48	6.95
RMSE	15.28%	31.78%	6.81%	43.01%	4.00%	46.15%	4.82%	50.17%

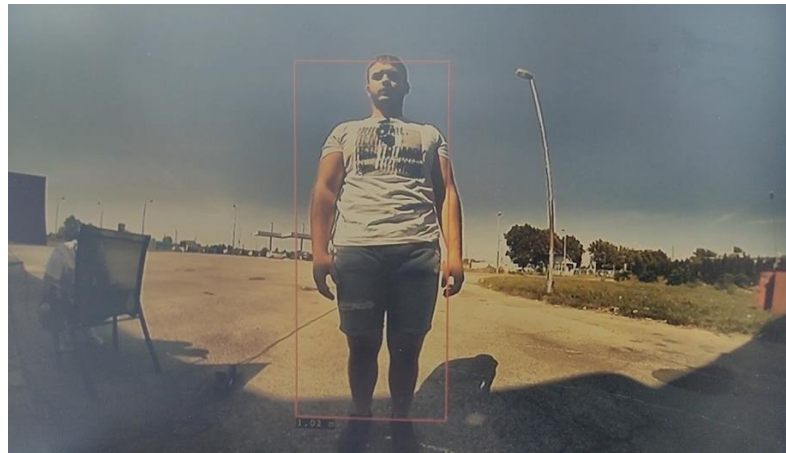
...

	Udaljenost 7m		Udaljenost 7.5m		Udaljenost 8m		Udaljenost 8.5m	
	Metoda 1	Metoda 2	Metoda 1	Metoda 2	Metoda 1	Metoda 2	Metoda 1	Metoda 2
	[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]
Osoba 1	7.07	7.58	7.46	8.0	8.01	8.59	8.65	9.27
Osoba 2	6.89	7.39	7.56	8.11	8.01	8.58	8.38	8.98
Osoba 3	6.90	7.39	7.57	8.11	7.90	8.46	8.12	8.70
Osoba 4	6.98	7.48	7.36	7.89	7.78	8.34	7.78	8.34
Osoba 5	6.81	7.30	7.37	7.89	7.89	8.46	8.12	8.71
RMSE	11.27%	43.84%	9.65%	50.96%	11.89%	49.46%	37.79%	43.20%
	Udaljenost 9m		Udaljenost 9.5m					
	Metoda 1	Metoda 2	Metoda 1	Metoda 2				
	[m]	[m]	[m]	[m]				
Osoba 1	8.93	9.58	9.08	9.74				
Osoba 2	8.92	9.57	9.23	9.90				
Osoba 3	8.25	8.85	8.38	8.98				
Osoba 4	8.25	8.85	8.25	8.85				
Osoba 5	9.08	9.73	9.23	9.90				
RMSE	47.81%	49.33%	79.24%	46.27%				

Iz dobivenih rezultata se može vidjeti da su pješaci uspješno detektirani na slikama s kamere i kada su se nalazili na udaljenostima većim od 9 metara. RMSE procijenjenih udaljenosti za metodu 1 su manji u odnosu na RMSE metode 2 na gotovo svim udaljenostima osim u posljednjem mjerenju, na 9.5 metara udaljenosti od kamere, kada metoda 2 ima manji RMSE od metode 1. Međutim RMSE je za obje metode velik iznad udaljenosti od 8 metara. Detekcija pješaka je uspješna i na udaljenostima oko 10 metara, ali se točnost procijenjene udaljenosti iznad 8 metara znatno smanjuje.

RMSE za ovu rezoluciju do 2.5 m su manji u odnosu na RMSE za prethodne dvije rezolucije. Iako su manji, ne razlikuju se puno jedni od drugih, iz čega se može zaključiti da rezolucija na malim udaljenostima nema značajan utjecaj na procjenu udaljenosti. Od 2.5 m do 4 m RMSE su manji za rezoluciju 960x540 elemenata slike u odnosu na ovu rezoluciju 1280x720 elemenata slike, dok za udaljenosti od 4 m do 5 m RMSE su manji za ovu rezoluciju u odnosu na prethodnu rezoluciju 960x540 elemenata slike. Iz toga se može zaključiti da na većim udaljenostima rezolucija ima utjecaj na procjenu udaljenosti.

Na slici 4.6 se mogu vidjeti neke od procijenjenih udaljenosti na okvirima rezolucije 1280x720 elemenata slike.



a)



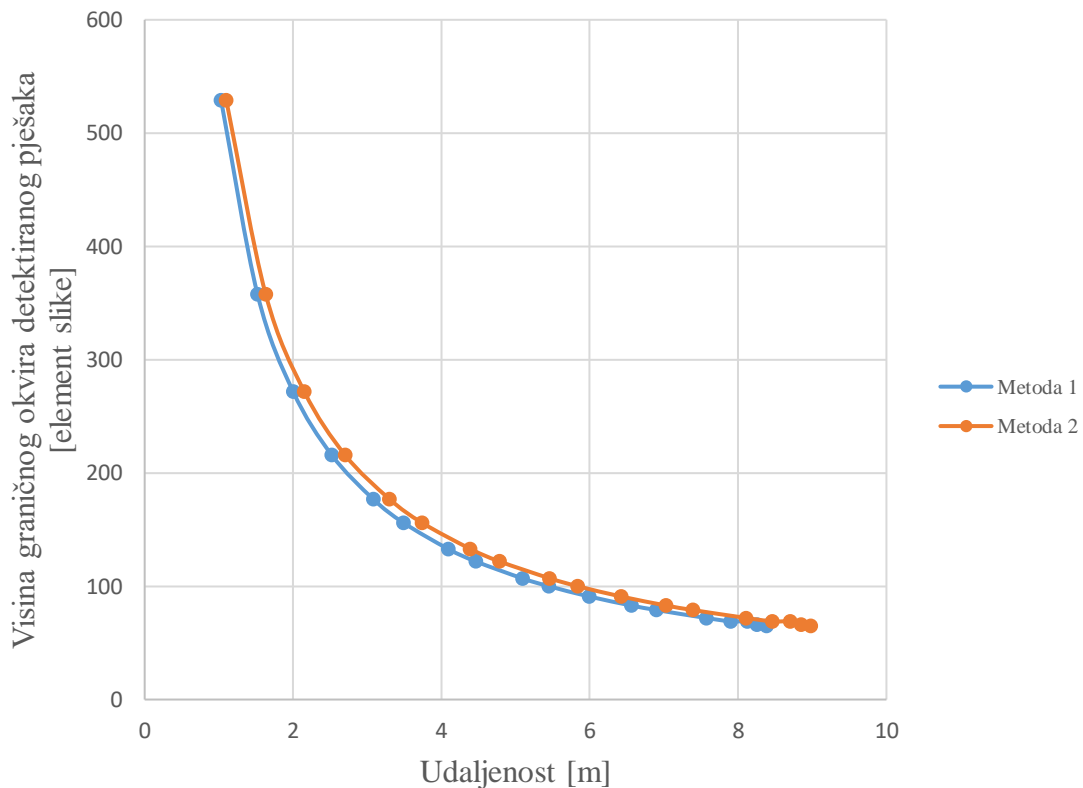
b)



c)

Sl. 4.7. Procijenjene udaljenosti na okvirima 1280x720 elemenata slike na (a) 1 metar, (b) 4 metra, (c) 9 metara stvarne udaljenosti za osobu 3

Slika 4.7 prikazuje graf odnosa visine graničnog okvira detektiranog pješaka i procijenjene udaljenosti za obje metode za osobu 3.



Sl. 4.8. Graf odnosa visine graničnog okvira detektiranog pješaka i procijenjene udaljenosti za okvir rezolucije 1280x720 elemenata slike za osobu 3

Iz grafa posljednjeg testiranja procjene udaljenosti se najbolje može vidjeti razlika promjene visine graničnog okvira detektiranog pješaka na malim udaljenostima u odnosu na promjenu visine graničnog okvira detektiranog pješaka na većim udaljenostima. Iznad 6 metara procijenjene udaljenosti, malom promjenom visine graničnog okvira se procijenjena udaljenost jako povećava. To znači da promjena visine od jednog elementa slike na malim udaljenostima ima malu promjenu u procjeni udaljenosti, dok kod velikih udaljenosti, promjena od jednog elementa slike znatno mijenja procjenu udaljenosti, a time i unosi veću pogrešku u krajnji rezultat. Visina graničnog okvira detektiranog pješaka na nekoj fiksnoj udaljenosti od kamere je različita za tri različite rezolucije. To znači da se udaljenost može preciznije procjenjivati na većim rezolucijama i da promjena visine od jednog elementa slike kod većih rezolucija unosi manju pogrešku u procjenu udaljenosti. Na video signalu koji se nalazi u elektroničkom prilogu ovom radu P.4.1.3. može se vidjeti kako radi algoritam za procjenu udaljenosti na okvirima rezolucije 1280x720 elemenata slike.

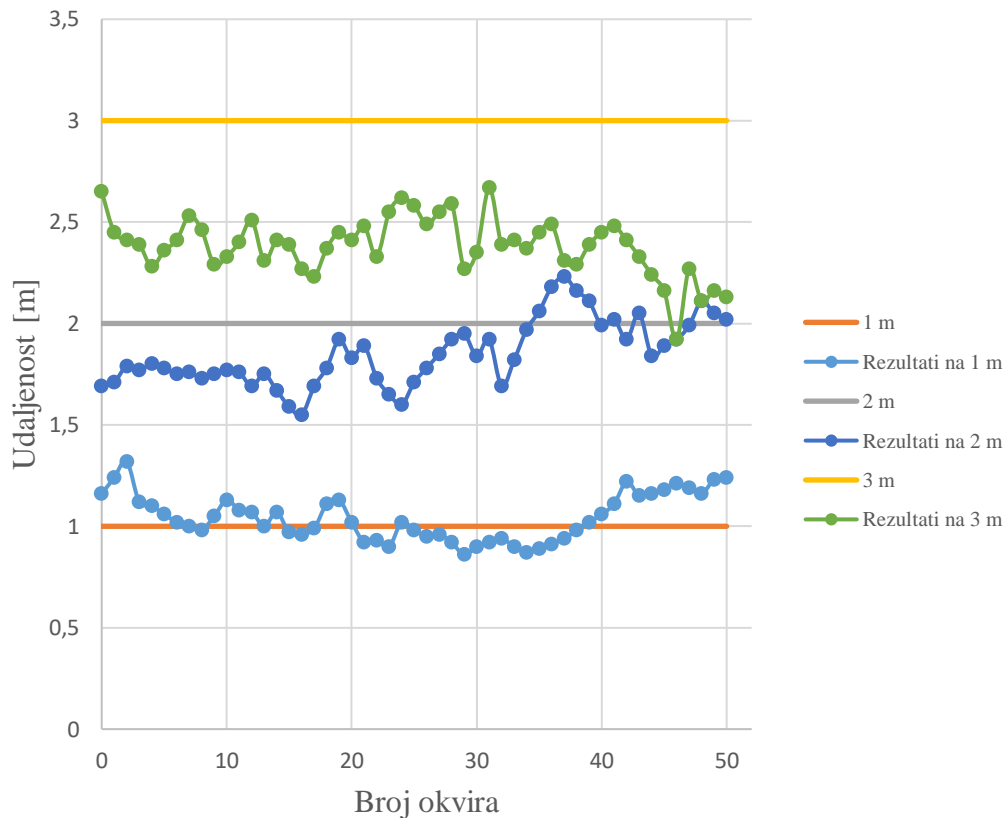
### **4.3. Testiranje ispravnosti rada algoritma za iscrtavanje 2D mape položaja pješaka u okolini vozila koristeći više kamera**

Posljednje testiranje je bilo testiranje algoritma za iscrtavanje 2D mape položaja pješaka u okolini vozila upotrebom više kamera u vozilu. Algoritam za iscrtavanje 2D mape položaja pješaka zasniva se na algoritmu detekcije pješaka i algoritmu za procjenu udaljenosti pješaka. Na osnovu ispravne detekcije, točne procjene udaljenosti, informacije na slici s koje kamere je pješak detektiran i pod kojim kutom, lako se položaj detektiranog pješaka iscrtava na 2D mapu. Tu se javlja jedan problem jer pješak često bude detektiran na slikama s dviju susjednih kamera. To se događa zato što, kada je pješak detektiran na dvije slike s dviju susjednih kamera, on je od jedne kamere udaljen na primjer 1 m, a od druge 1,2 m, čemu još nepovoljno doprinosi iskrivljenje prostora u krajevima područja kojeg hvataju širokokutne kamere, što rezultira iscrtavanjem dvaju kvadratića/markera, jedan neposredno blizu drugog, od kojih oba pripadaju istom pješaku. To dovodi do netočnih rezultata, jer u tom slučaju mapa prikazuje više pješaka nego što ih zapravo ima u okolini vozila. Zato je algoritam za iscrtavanje 2D mape položaja pješaka u okolini vozila dodatno proširen, čime se riješio problem iscrtavanja dvaju kvadratića koji pripadaju istom pješaku. Ono što je testirano kod ovog algoritma je iscrtavanje položaja pješaka na 2D mapu u trenucima prelaska pješaka iz područja kojeg hvata jedna kamera, preko područja kojeg hvataju dvije susjedne kamere u područje kojeg hvata druga kamera. Testiranja su provedena u sunčanim vremenskim uvjetima s jakim osvjetljenjem i područjima promjene osunčanog dijela okoline i hladovine. Testirano je tako da se vozilo nalazi u sredini, a oko njega su ocrtane koncentrične elipse na udaljenostima 1, 2, 3, 4 i 5 metara od pojedine kamere. Pješak se kretao po elipsama oko vozila, a rezultati iscrtavanja 2D mape položaja pješaka koji se kretao oko vozila, a koji su dobiveni na ekranu za prikaz, snimljeni su mobilnim telefonom. Video materijali snimki nastalih prilikom testiranja algoritma za iscrtavanje 2D mape položaja pješaka, koji se kreće oko vozila na određenim udaljenostima prelaskom iz područja koje hvata jedna kamera u područje koje hvata druga kamera, za okvire različitih rezolucija nalaze se u elektroničkom prilogu ovom radu P.4.2.

Testiranje je sprovedeno na dvama slučajevima upotrebe. Prvi slučaj upotrebe na kojem je izvršeno testiranje je slučaj upotrebe konačnog rješenja, koji podrazumijeva iscrtavanje 2D mape položaja pješaka u okolini vozila upotrebom četiriju kamera. Kod slučaja upotrebe konačnog rješenja, okviri video signala su rezolucije 640x360 elemenata slike. To znači da, prema ranije provedenim testiranjima, detekcija i proračun udaljenosti funkcioniraju do 3 metra. Testiranje je izvedeno tako da se osoba 3 kretala po elipsama, koje su od kamere vozila na svakoj strani vozila udaljene 1 m, 2 m i 3 m, tako da je osoba 3 šetala iz područja desne kamere preko područja prednje

kamere, u područje lijeve kamere. Iz snimljenih videa izdvojeno je po 50 okvira za svako od tri kretanja osobe, odabranih tako da pokrivaju područje koje hvataju prednja, desna i lijeva kamera.

Na slici 4.8 se može vidjeti graf rezultata procjene udaljenosti kretanja osobe 3 po elipsama na određenim udaljenostima oko vozila.

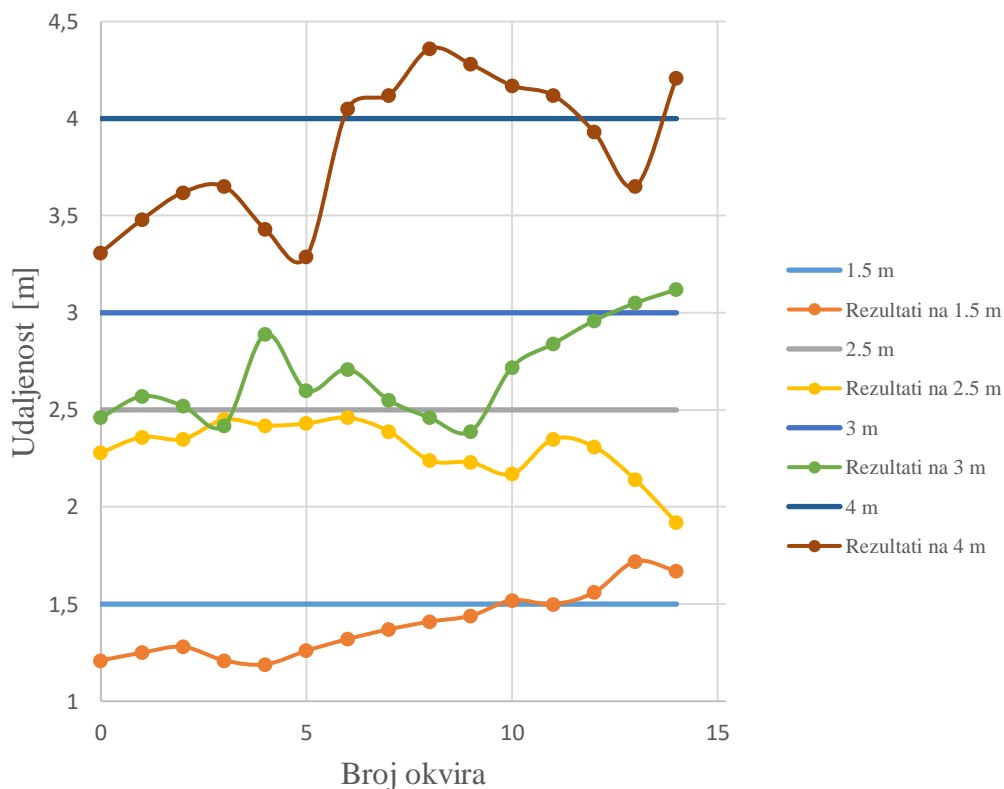


Sl. 4.9. Graf rezultata procjene udaljenosti kretanja osobe po elipsama na određenim udaljenostima oko vozila u slučaju upotrebe, koji podrazumijeva iscrtavanje 2D mape položaja pješaka u okolini vozila upotrebom četiriju kamera

Iz dobivenih rezultata se može vidjeti da proračun udaljenosti osobe 3, koja se kreće oko vozila, varira radi nestabilnog graničnog okvira oko detektiranog pješaka. U prvom slučaju, kada se osoba kreće po elipsi koja je udaljena 1 metar od vozila, procjene udaljenosti su blizu stvarnoj vrijednosti od 1 metar. U drugom slučaju, kada osoba hoda po elipsi koja je udaljena 2 metra od vozila, pojavljuju se veća odstupanja pri procjeni udaljenosti, dok kod trećeg slučaja kada osoba hoda po elipsi, koja je udaljena 3 metra od vozila, odstupanja su primjetna jer je procjena udaljenosti pješaka na okviru video signala rezolucije 640x360 elemenata slike uspješna do 2.5 metara. Problem prelaska osobe iz područja kojeg hvata jedna kamera, preko područja kojeg hvataju dvije susjedne kamere u područje kojeg hvata druga kamera, uspješno je riješen algoritmom za iscrtavanje 2D mape. Na video signalima, koji se nalaze u elektroničkom prilogu

ovom radu P.4.2.1. može se vidjeti kako radi iscertavanje 2D mape položaja pješaka koji se kreće oko vozila na određenim udaljenostima prelaskom iz područja koje hvata jedna kamera u područje koje hvata druga kamera, za okvire rezolucije 640x360 elemenata slike.

Drugi slučaj upotrebe na kojem je izvršeno testiranje je slučaj upotrebe kod kojeg su se za iscertavanje 2D mape položaja pješaka u okolini vozila koriste kombinacije dviju kamera, prednja – lijeva ili prednja – desna. Rezolucije okvira pojedine slike s kamere su 960x540 elemenata slike. Prema ranije provedenim testiranjima, detekcija i proračun udaljenosti rade do 5 metara udaljenosti pješaka od kamere na vozilu. Testiralo se na sličan način ponovo s osobom 3. U ovom slučaju osoba je hodala po elipsama koje su udaljene 1.5, 2.5, 3 i 4 metra. Na slici 4.9 se može vidjeti graf rezultata procjene udaljenosti kretanja osobe po elipsama na određenim udaljenostima oko vozila.



Sl. 4.10. Graf rezultata procjene udaljenosti kretanja osobe po elipsama na određenim udaljenostima oko vozila u slučaju upotrebe, koji podrazumijeva iscertavanje 2D mape položaja pješaka u okolini vozila upotrebom dvije kamere

Iz dobivenih rezultata se može vidjeti da, kao i u prethodnom slučaju, vrijednosti proračuna udaljenosti osobe 3, koja se kretala po elipsama oko vozila, variraju zbog nestabilnosti graničnog



okvira oko detektiranog pješaka. Na udaljenostima od 1.5 i 2.5 metara od kamere, vrijednosti ne odstupaju puno od stvarnih, dok kod udaljenosti većih od 3 metra postoje znatna odstupanja od stvarnih vrijednosti. Na video signalima, koji se nalaze u elektroničkom prilogu ovom radu P.4.2.2. može se vidjeti kako radi iscertavanje 2D mape položaja pješaka koji se kreće oko vozila na određenim udaljenostima prelaskom iz područja koje hvata jedna kamera u područje koje hvata druga kamera za okvire rezolucije 960x540 elemenata slike.

#### **4.4. Osvrt na dobivene rezultate**

Nakon provedenog testiranja i evaluacije predloženog rješenja, može se zaključiti da je glavni uzrok većine pogrešaka algoritam za detekciju pješaka, koji je dio VisionSDK okruženja i na čiju ispravnost se nije moglo dodatno utjecati. S obzirom na to da je odziv algoritma za detekciju zadovoljavajući i iznosi 79,09%, ono što je jako loše, je preciznost algoritma koja iznosi 44.25%. Često se oblaci, lišće, bandere i druge stvari, koje nemaju niti jednu zajedničku značajku s ljudima detektiraju kao pješaci. Zbog detekcije drugih objekata kao pješaka, loše se iscertavaju na 2D mapi položaji pješaka, koji su detektirani u okolini vozila. Osim preciznosti koja je niska, na iscertavanje 2D mape položaja pješaka u okolini vozila utječe i neprecizno iscertavanje graničnog okvira oko detektiranog pješaka, odnosno niski IoU. Kako su korištene širokokutne kamere, o kojima nema informacija na internetu, parametri istih su se morali ručno proračunati. Zbog toga se dodatno zasigurno unijela pogreška u krajnje rezultate.

Rezultati proračuna udaljenosti variraju u maloj mjeri od osobe do osobe, koje su testirane. Prosječna visina čovjeka u Europi i iznosi 1.77 m [26], a visine pješaka su različite, stoga su se morala provesti testiranja s više osoba različitih visina. Iz dobivenih rezultata se vidi da se na okvirima veće rezolucije pješaci mogu detektirati na većim udaljenostima. RMSE za pojedinu udaljenost je za većinu rezultata manji kod metode 1 u odnosu na metodu 2. S obzirom na to da su se prilikom testiranja, koristile različite rezolucije okvira video signala, iz grafova odnosa visine graničnog okvira detektiranih pješaka i procijenjenih udaljenosti, može se zaključiti da visina jednog elementa slike kod proračuna većih udaljenosti, unosi veću pogrešku, nego kod proračuna manjih udaljenosti. Iz toga se da zaključiti da su pogreške proračuna većih udaljenosti veće kod okvira manjih rezolucija, nego kod okvira većih rezolucija.

Ispravnost algoritma za iscertavanje 2D mape položaja pješaka u okolini vozila ovisi o ispravnosti algoritma za detekciju pješaka i o točnosti algoritma za procjenu udaljenosti. Parametri koji su potrebni algoritmu za iscertavanje 2D mape položaja pješaka su što točnija procjena

udaljenosti, kut i informacija o tome na kojoj je kameri pješak detektiran. Iz dobivenih rezultata provedenog testiranja kretanja pješaka po elipsi, može se vidjeti da pri manjim udaljenostima od kamere procjena udaljenosti i iscrtavanje položaja pješaka na 2D mapu ne odstupa puno od stvarne udaljenosti. Čim se osoba počne kretati po elipsama koje su udaljenije od kamere, visina graničnog okvira oko detektiranog pješaka počinje varirati, što unosi pogrešku u proračun udaljenosti, a time i u krajnje iscrtavanje položaja pješaka na 2D mapu. Prelazak iz područja kojeg hvata jedna kamera, preko područja kojeg hvataju dvije susjedne kamere, u područje druge kamere izvodi se fluidno bez pojave dvaju kvadratića koji predstavljaju jednog pješaka, što se može vidjeti na video signalima koji se nalaze u elektroničkom prilogu ovom radu P.4.2.

S obzirom na to da su kamere širokokutne, a kalibracija istih se nije izvršila, pogreške u proračunu udaljenosti detektiranih pješaka najveće su u krajnjim područjima kamera, gdje je prostor iskrivljen i kada se pješak nađe u tim područjima, na slici izgleda iskrivljeno, odnosno ako ga je detektor uspio detektirati iskrivljenjem mu je smanjena visina, a time i procjena udaljenosti. Ovaj problem se riješio tako što se područje koje hvataju kamere podijelilo na 6 područja te im se pridijelile odgovarajuće težine. Područja što bliže sredini su imala veću težinu, dok su područja bliža krajevima dobila manju težinu. U slučaju da se pješak našao u području kojeg hvataju dvije susjedne kamere na osnovu težina područja u kojima se isti našao, izvršavalo se iscrtavanje 2D mape položaja istog u okolini vozila.

## 5. ZAKLJUČAK

Potpuna autonomna vožnja jedan je od ciljeva koji se žele ostvariti u automobilskoj industriji u skorijoj budućnosti. Da bi to bilo moguće, ADAS moraju obavljati sve funkcije s visokom preciznošću i točnošću, ne ostavljajući ni malo prostora greškama. Jedna od glavnih funkcija, koju autonomna vozila moraju obavljati, je detekcija objekata u okolini vozila, u kojoj se isto kreće. Tema ovog rada je iscrtaavanje 2D mape položaja pješaka u okolini vozila. Zadatak je bio razviti vlastiti slučaj upotrebe, unutar kojeg će se stvoriti vlastiti algoritam za procjenu udaljenosti detektiranih pješaka i algoritam za iscrtaavanje 2D mape položaja pješaka u okolini vozila. Navedene algoritme bilo je potrebno povezati s algoritmom za detekciju pješaka, koji je dio VisionSDK okruženja. Nakon povezivanja svih potrebnih veza u funkcionalni slučaj upotrebe, započeto je razvijanje potrebnih algoritama u C programskom jeziku. Algoritam za procjenu udaljenosti razvijen je upotrebom dviju metoda, čiji su rezultati u radu uspoređeni. Nakon razvijenog algoritma za procjenu udaljenosti, razvijen je algoritam za iscrtaavanje 2D položaja pješaka, koji se nalaze u okolini vozila. S obzirom na ograničenje algoritma za detekciju pješaka, kod kojeg se proces detekcije izvršava na okvirima maksimalne rezolucije 1280x720 elemenata slike, video signalima rezolucije 1280x720 elemenata slike s četiriju kamera, koje snimaju okolinu vozila, morala se smanjiti rezolucija na 640x360 elemenata slike, kako bi se sva četiri okvira pojedinog video signala mogla rasporediti u okvir mozaičkog prikaza ukupne rezolucije 1280x720 elemenata slike, na kojem se izvršavala detekcija. Nakon razvijenog slučaja upotrebe predloženog rješenja, radi testiranja i evaluacije rješenja razvijena su dodatna dva slučaja upotrebe, unutar kojih se algoritam za detekciju pješaka izvršavao na video signalima s većom rezolucijom u odnosu na onu korištenu u konačnom rješenju, samo iz razloga kako bi se ispitalo dolazi li do veće točnosti detekcije i procjene udaljenosti za veće udaljenosti kod okvira većih rezolucija.

Testiranja su prvo provedene na algoritmu za detekciju objekata. Utvrđeno je da algoritam za detekciju objekata ima zadovoljavajuće rezultate odziva, ali da mu je preciznost jako loša. Osim niske preciznosti, problem je i neprecizno iscrtaavanje graničnog okvira oko detektiranog pješaka, odnosno niski IoU. Nakon prvog testiranja provedena su testiranja na algoritmu za procjenu udaljenosti i algoritmu za iscrtaavanje 2D položaja pješaka, koji se kreće po elipsi u okolini vozila. Testiranja vlastitih algoritama su sprovedena na više slučajeva upotrebe za različite rezolucije videa. Zaključeno je da u slučajevima upotrebe, u kojima se obrađuju okviri manjih rezolucija, detekcija pješaka je moguća na manjoj udaljenosti od kamere na vozilu, dok kod slučaja upotrebe, u kojima se obrađuju okviri većih rezolucija detekcije pješaka su moguće i do 10 metara udaljenosti od kamere. S obzirom na to da se vlastiti algoritmi oslanjaju na algoritam za detekciju

pješaka i ovise o njemu, rezultati dobiveni testiranjem vlastitih algoritama su zadovoljavajući na udaljenostima bližim kameri. Upotreba nekalibriranih širokokutnih kamera, ručni proračun parametara kamere te lošiji algoritam za detekciju pješaka najviše unose pogreške u rezultate. Dodatnim razvojem vlastitih algoritama te kalibracijom kamera, kao i modifikacijom algoritma za detekciju pješaka mogli bi se postići bolji rezultati konačnog rješenja, a kasnije možda i upotreba u stvarnom svijetu.

## LITERATURA

- [1] M. Vranješ, D. Vajak, »Predložci s laboratorijskih vježbi iz kolegija „Digitalna obrada slike i videa za autonomna vozila“, vježbe „Upoznavanje s ADAS razvojnom pločom“ i „Izrada vlastitog use-case-a za ADAS razvojnu ploču“«.
- [2] »The 6 Levels of Vehicle Autonomy Explained,« [Mrežno]. Available: <https://www.synopsys.com/automotive/autonomous-driving-levels.html>.
- [3] C. Hall, »Self-driving cars: Autonomous driving levels explained,« 2020. [Mrežno]. Available: <https://www.pocket-lint.com/cars/news/143955-sae-autonomous-driving-levels-explained>.
- [4] N. Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G.V. Hernandez, L. Krpalkova, D. Riordan, J. Walsh, »Deep Learning vs. Traditional Computer Vision,« 2019.
- [5] Y. Sakai, T. Oda, M. Ikeda, L. Barolli, »An Object Tracking System Based on SIFT and SURF Feature Extraction Methods,« 2015.
- [6] D. Lowe, »Distinctive Image Features from Scale-Invariant Keypoints,« 2004.
- [7] H. Bay, T. Tuytelaars, L.V. Gool, »SURF: Speeded Up Robust Features,« 2006.
- [8] M. A. Rashidan, Y. M. Mustafah, Z. Z. Abidin, N. A. Zainuddin, N. N. A. Aziz , »Analysis of Artificial Neural Network and Viola-Jones Algorithm Based Moving Object Detection,« 2014.
- [9] P. Viola, M. Jones, »Robust Real - Time Object Detection,« 2011.
- [10] K.B. Pranav, J. Manikandan, »Design and Evaluation of a Real-time Pedestrian,« 2020.
- [11] »INRIA Person Dataset,« [Mrežno]. Available: <http://pascal.inrialpes.fr/data/human/>.
- [12] Y. Deng, P. Luo, C.C. Loy, X. Tang, »Pedestrian Attribute Recognition At Far Distance,« [Mrežno]. Available: <http://mmlab.ie.cuhk.edu.hk/projects/PETA.html>.
- [13] D. Bao, P. Wang, »Vehicle Distance Detection Based on Monocular,« 2016.
- [14] A.G. Hapsani, D. Syauqy, F. Utamingrum, P.P. Adikara, S. Adinugroho, »Onward Movement Detection and Distance Estimation of Object Using Disparity Map on Stereo Vision,« 2017.
- [15] W. Lan, J. Dang, Y. Wang, S. Wang, »Pedestrian Detection Based on YOLO Network Model,« 2018.
- [16] »RT-RK - Automotive,« [Mrežno]. Available: <https://www.rt-rk.com/services/automotive>.
- [17] »VISION SDK, Links Framework - Deep Dive«.
- [18] L. Umiljanović, »Diplomski rad, "KORISNIČKA APLIKACIJA UREĐAJA ZA SNIMANJE I REPRODUKCIJU VIDEO SADRŽAJA ZA VERIFIKACIJU ADAS ALGORITAMA,« 2018.
- [19] I. Jamal, M.U. Akram, A. Tariq, »Retinal Image Preprocessing: Background and Noise«.
- [20] E. H. Adelson, C. H. Anderson, R. Bergen, P. J. Burt, J. M. Ogden, »Pyramid methods in image processing«.
- [21] »Image Pyramid,« [Mrežno]. Available: [https://en.wikipedia.org/wiki/Pyramid\\_\(image\\_processing\)#/media/File:Image\\_pyramid.svg](https://en.wikipedia.org/wiki/Pyramid_(image_processing)#/media/File:Image_pyramid.svg).
- [22] »About OpenCV,« [Mrežno]. Available: <https://opencv.org/about/>.
- [23] »Install OpenCV-Python in Windows,« [Mrežno]. Available: [https://docs.opencv.org/3.4.1/d5/de5/tutorial\\_py\\_setup\\_in\\_windows.html](https://docs.opencv.org/3.4.1/d5/de5/tutorial_py_setup_in_windows.html).
- [24] »NumPY,« [Mrežno]. Available: <https://numpy.org/doc/>.

- [25] »Installing NumPY,« [Mrežno]. Available: <https://numpy.org/install/>.
- [26] I. Langtree, »Height Chart of Men and Women in Different Countries,« 2017. [Mrežno]. Available: <https://www.disabled-world.com/calculators-charts/height-chart.php>.
- [27] A. Geige, P. Lenz, C. Stiller, R. Urtasun, »Kitti Raw Dataset,« [Mrežno]. Available: [http://www.cvlibs.net/datasets/kitti/raw\\_data.php?type=city](http://www.cvlibs.net/datasets/kitti/raw_data.php?type=city).

## SAŽETAK

U ovom radu opisane su postojeće metode detekcije objekata u slikama. Da bi se razvila potpuna autonomna vožnja kojoj automobilska industrija stremi, ADAS moraju obavljati funkcije s visokom točnošću i preciznošću. Nakon opisa postojećih metoda za detekciju objekata, u sklopu ovog rada razvijeno je vlastito rješenje za iscrtavanje 2D mape položaja pješaka u okolini vozila. Prvo je razvijen vlastiti slučaj upotrebe unutar VisionSDK razvojnog okruženja, koji se zasnivao na algoritmu za detekciju objekata, koji je dio VisionSDK razvojnog okruženja. Nakon razvoja vlastitog slučaja upotrebe, započet je razvoj vlastitih algoritama za procjenu udaljenosti i za iscrtavanje 2D mape položaja pješaka u okolini vozila. Vlastiti algoritmi su izrađeni u C programskom jeziku, a konačno rješenje je implementirano na ADAS ALPHA ploči. Evaluacijom rješenja je utvrđeno da je algoritam za detekciju pješaka nedovoljno pouzdan, ali da ima zadovoljavajuću razinu odziva. Rezultati procjene udaljenosti su pokazali da detekcija i procjena udaljenosti uvelike ovise o rezoluciji okvira video signala na kojima se detekcija pješaka izvršava. U konačnici, krajnji rezultat iscrtavanja 2D mape položaja pješaka u okolini vozila postiže zadovoljavajuće rezultate na manjim udaljenostima od kamere. Kalibracijom kamera i modificiranjem algoritma za detekciju kao i vlastitih algoritama, ispravnost rada konačnog rješenja mogla bi se povećati.

**Ključne riječi:** *detekcija pješaka, ADAS, VisionSDK, slučaj upotrebe, 2D mapa položaja pješaka*

# PEDESTRIAN DETECTION USING MULTIPLE CAMERAS LOCATED IN THE VEHICLE

## ABSTRACT

This master's thesis describes existing object detection methods in images. In order to develop the fully autonomous driving that the automotive industry strives for, ADAS must perform functions with high accuracy and precision. After the description of the existing methods for object detection, as part of this master's thesis, own solution for drawing a 2D map of the position of pedestrians in the environment of the vehicle was developed. First, own use case within the VisionSDK development environment, which was based on an object detection algorithm, which is part of the VisionSDK development environment, was developed. After developing own use case, own algorithms for estimating the distance and for drawing a 2D map of the position of pedestrians in the vicinity of the vehicle were developed. Own algorithms were developed in C programming language, and the final solution was implemented on the ADAS ALPHA board. The evaluation of the solution revealed that the algorithm for pedestrian detection is insufficiently reliable, but that it has a satisfactory level of response. The results of the distance estimation have shown that the detection and estimation of the distance largely depend on the resolution of the video signal frames on which the pedestrian detection is performed. Ultimately, the end result of drawing a 2D map of pedestrian positions around the vehicle achieves satisfactory results at shorter distances from the camera. By calibrating the cameras and modifying the detection algorithm as well as our own algorithms, the correctness of the operation of the final solution could be increased.

**Keywords:** *pedestrian detection, ADAS, VisionSDK, use case, 2D pedestrian position map*



## ŽIVOTOPIS

Nemanja Avramović rođen je u Vukovaru 06.11.1996. Pohađao je Osnovnu školu „Borovo“. Nakon završene osnovne škole upisuje Gimnaziju Vukovar, smjer Opća gimnazija. Gimnaziju završava 2015. godine, nakon čega iste godine upisuje Preddiplomski sveučilišni studij elektrotehnike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, tadašnji Elektrotehnički fakultet, na Sveučilištu J.J. Strossmayera. Nakon završetka preddiplomskog studija elektrotehnike, 2018. godine stječe akademski naziv sveučilišni prvostupnik (lat. *baccalaureus*) inženjer elektrotehnike. Iste godine upisuje diplomski sveučilišni studij Elektrotehnike, smjer komunikacije i informatika, izborni blok Mrežne tehnologije u Osijeku.

---

Potpis autora

## **PRILOZI**

P.3.1. – Mapa u kojoj se nalazi kod, koji sadrži algoritam za iscrtavanje 2D mape položaja pješaka u okolini vozila (elektronički prilog)

P.4.1. – Mapa koja sadrži video signale za testiranje algoritma za procjenu udaljenosti (elektronički prilog)

P.4.1.1. – Mapa koja sadrži video signal za testiranje algoritma za procjenu udaljenosti detektiranog pješaka na okvirima rezolucije 640x360 elemenata slike (elektronički prilog)

P.4.1.2. – Mapa koja sadrži video signal za testiranje algoritma za procjenu udaljenosti detektiranog pješaka na okvirima rezolucije 960x540 elemenata slike (elektronički prilog)

P.4.1.3. – Mapa koja sadrži video signal za testiranje algoritma za procjenu udaljenosti detektiranog pješaka na okvirima rezolucije 1280x720 elemenata slike (elektronički prilog)

P.4.2. – Mapa koja sadrži video signale za testiranje algoritma za iscrtavanje položaja pješaka na 2D mapu prilikom prelaska iz područja jedne kamere u područje druge kamere (elektronički prilog)

P.4.2.1. – Mapa koja sadrži video signale za testiranje algoritma za iscrtavanje položaja pješaka na 2D mapu prilikom prelaska iz područja jedne kamere u područje druge kamere na okvirima rezolucije 640x360 (elektronički prilog)

P.4.2.2. – Mapa koja sadrži video signale za testiranje algoritma za iscrtavanje položaja pješaka na 2D mapu prilikom prelaska iz područja jedne kamere u područje druge kamere na okvirima rezolucije 960x540 (elektronički prilog)