Lokalizacija i praćenje objekata na nizu sekvencijalnih slika korištenjem dubokih neuronskih mreža

Komušar, Hrvoje

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 122

LOKALIZACIJA I PRAĆENJE OBJEKATA NA NIZU SEKVENCIJALNIH SLIKA KORIŠTENJEM DUBOKIH NEURONSKIH MREŽA

Hrvoje Komušar

Zagreb, veljača 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 122

LOKALIZACIJA I PRAĆENJE OBJEKATA NA NIZU SEKVENCIJALNIH SLIKA KORIŠTENJEM DUBOKIH NEURONSKIH MREŽA

Hrvoje Komušar

Zagreb, veljača 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 3. listopada 2024.

DIPLOMSKI ZADATAK br. 122

Pristupnik:	**Hrvoje Komušar (0036523919)**
Studij:	Informacijska i komunikacijska tehnologija
Profil:	Automatika i robotika
Mentor:	izv. prof. dr. sc. Vinko Lešić

Zadatak: **Lokalizacija i praćenje objekata na nizu sekvencijalnih slika korištenjem dubokih neuronskih mreža**

Opis zadatka:

Geometrijskom projekcijom moguće je ostvariti detekciju trodimenzionalnih objekata iz monokularnih zapisa dvodimenzionalnih slika i djelomično poznatih dimenzija objekata. U sklopu rada potrebno je razviti algoritam za praćenje položaja i brzine objekata tijekom vremena iz niza sekvencijalnih slika preuzetih iz javno dostupnog skupa podataka i primjene u autonomnoj vožnji u cestovnom prijevozu. Pri tome je potrebno odabrati pogodnu arhitekturu duboke neuronske mreže i primijeniti ju za ekstrakciju informacija iz slika te ostvariti procjenu brzine gibanja objekta. Nesavršenosti u detekciji i gubitak informacija potrebno je nadomjestiti primjenom estimacijskih algoritama kao što je Kalmanov filter. Potrebno je dodatno načiniti analizu utjecaja precizne procjene brzine objekta na točnost njegove lokalizacije.

Rok za predaju rada: 14. veljače 2025.

Sadržaj

Uvod

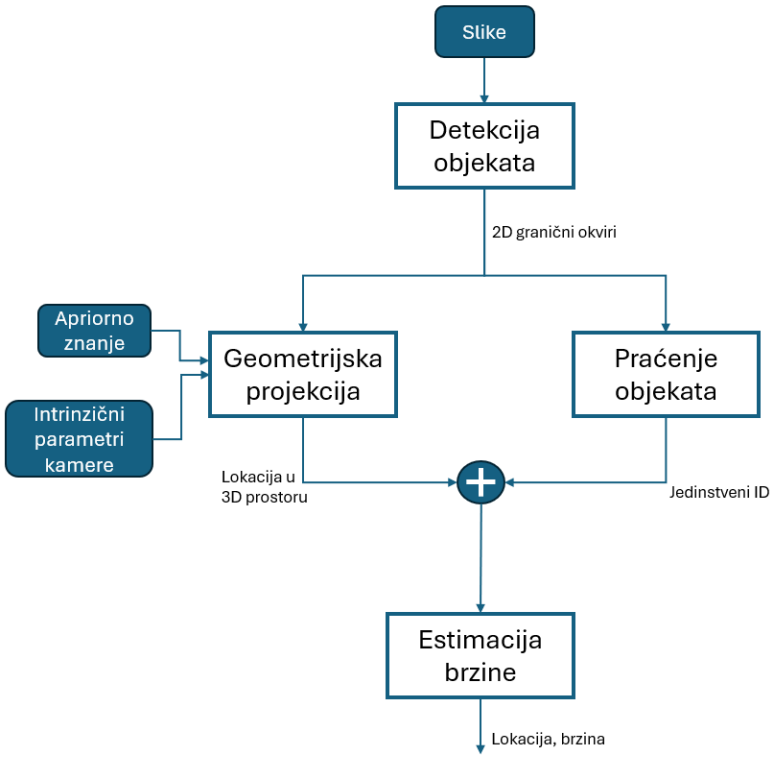
Estimacija pozicije i brzine objekata je veliki izazov u robotici, autonomnoj vožnji i nadzoru prometa. Tradicionalne se metode oslanjaju na podatke dobivene od raznih senzora - radara, lidara, višestrukih setova kamera, kako bi se dobili robusni podatci potrebni za točnu estimaciju. Svaki dio sustava nudi određenu prednost. Radar se ističe u mjerenju brzina objekata zahvaljujući svom principu rada temeljenom na Dopplerovom efektu [1], lidar laserskim pulsevima mapira svoje okruženje [2], dok stereo kamere omogućuju bolju percepciju dubine i šire vidno polje [3]. Takav pristup, iako vrlo efektivan, može biti skup, računalno zahtjevan i nepraktičan za slučajave s ograničenim resursima.

Stoga se nameće pitanje - koliko je moguće pojednostaviti proces prikupljanja podataka te samim time i njihovu obradu, a da estimacija i dalje bude bliska stvarnim vrijednostima? U ovom radu istražit će se izvedivost korištenja samo niza 2D slika, prikupljenih monokularnom kamerom, za estimaciju pozicija i brzina objekata. Takav način ne koristi kompleksnije tehnike poput optičkog toka koje analiziraju pomak piksela na slikama za estimaciju gibanja [4]. Umjesto toga, integriraju se model za detekciju objekata, 3D geometrijska projekcija, algoritam za praćenje objekata i estimacija brzine uz korištenje Kalmanovog filtra.

Prednost ove minimalističke metode je da ne zahtijeva dodatne ulazne podatke. U radu će biti evaluiran predloženi model u različitim scenarijima, što uključuje njegovu mogućnost da precizno lokalizira i prati objekte, te da estimira njihovu brzinu. Istraživanjem potencijala sustava samo s jednom kamerom pridonosimo razvoju ekonomičnijih rješenja u primjeni računalnog vida.

Da bi se došlo od sekvence slika do estimiranih brzina potrebno je postupak razložiti na nekoliko koraka. Proces počinje detekcijom objekata na nizu slika pomoću modela

YOLOv5.[5] Detektiranim objektima se dodjeljuju 2D granični okviri, odnosno koordinate centra u pikselima te širina i visina okvira. Ti podatci, zajedno s intrinzičnim parametrima kamere i apriornim znanjem o visini objekta, koriste se za procjenu dubine, iz čega proizlazi pozicija objekta u 3D prostoru.



Slika 1. Grafički prikaz tijeka rada

Paralelno s geometrijskom projekcijom objektima se dodjeljuju jedinstveni ID-jevi koji omogućuju praćenje objekta u svrhu računanja brzine. Spajanjem lociranih objekata s njihovim identifikatorima sustav može računati promjenu pozicije objekta u vremenu. Zbog nesavršenosti u detekciji i lokalizaciji objekata, integriran je Kalmanov filtar kako bi se povećala robusnost i preciznost sustava. Kalmanov filtar zaglađuje zašumljena mjerenja te popunjava rupe u slučaju privremenog nestanka mjerenja. Time se osigurava da se kretanje objekta precizno prati unatoč nedostatcima u prikupljanju podataka.

1. Podatci za praćenje objekata u prometu

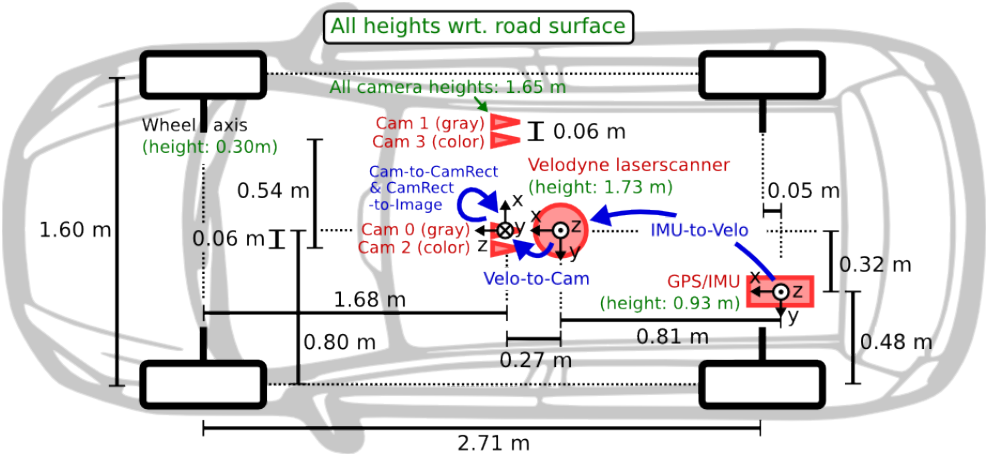
KITTI (*Karlsruhe Institute of Technology and Toyota Technological Institute*) *dataset* je sveobuhvatan paket mjerila računalnog vida u stvarnom svijetu, često korištenih u istraživanju autonomne vožnje i mobilne robotike. Razvijen je od strane Tehnološkog Instituta Karlsruhe i Toyotinog tehnološkog instrituta u Chicagu. Podatci su podijeljeni u nekoliko skupova ovisno o njihovoj primjeni u stvarnom svijetu - stereo vizija, optički tok, vizualna odometrija, trodimenzionalna detekcija, praćenje objekata. Svi podatci su skupljeni vožnjom po Karlsruheu u Njemačkoj. Pokriveni su različiti scenariji: vožnja gradom, ruralnim cestama te autocestom. Svaka slika može sadržavati do 15 automobila i 30 pješaka čime se nude raznoliki podatci za razvoj algoritma.[6]

Za prikupljanje podataka, automobil Volkswagen Passat B6 opremljen je sljedećim mjernim uređajima:[7]

- Kamere
  - Dvije kamere visoke rezolucije u boji 1.4 MP (FL2-14S3C-C)
  - Dvije crno-bijele kamere 1.4 MP (FL2-14S3M-C)
- Lidar senzor (Velodyne HDL-64E)
- Interni navigacijski sustav: GPS/IMU (*Internal Measurement Unit*) - praćenje lokacije i podataka o gibanju vozila (OXTS RT 3003)
- Četiri varifokalne leće 4-8 mm (Edmuns Optics NT59-917)

Laserski skener se vrti brzinom 10 sličica po sekundi. Vertikalna rezolucija lasera je 64, odnosno uređaj se sastoji od 64 emitera koji vertikalno skeniraju svoje okruženje. Kamere, postavljene paralelno s ravninom tla, daju slike veličine 1382x512 piksela, no nakon rektifikacije (uklanjanja geometrijskih deformacija) veličina se smanji na 1242x375

piksela. Laserski skener služi kao okidač za kamere koje uzimaju slike istom frekvencijom od 10 sličica u sekundi uz dinamičko određivanje vremena ekspozicije ovisno o trenutnom osvjetljenju.[7]



Slika 1.1. Opremljeno vozilo, shema s gornje strane[7]

Slika 1.2. Opremljeno vozilo, pogled izvana[7]

1.1. Opis podataka

KITTI sadržava skupove podataka namijenjene različitim problemima poput detekcije objekata, estimacije dubine, stereo vizije, semantičke segmentacije te praćenja objekata. Svaki skup sastoji se od sekvenci slika te njihovih anotacija. U ovom radu, korišten je skup podataka specifično dizajniran za razvoj algoritama za praćenje objekata u pokretu. Podatke čini nekoliko komponenti:

1. Sekvenca slika - lijeva i desna kamera

   Sastoji se od podataka za treniranje koji sadržavaju 20 različitih sekvenci, te od podataka za testiranje koji sadržavaju 28 sekvenci slika. Svaka sekvenca je novi scenarij na cesti, snimljen na različitom mjestu s različitim prometnim uvjetima. Koristit će se samo slike dobivene lijevom kamerom jer želimo sve informacije o dubini dobiti geometrijskom projekcijom, a ne pomoću dodatnih informacija koje pruža druga kamera.

(a) Scenarij 3 - cesta s medijanom

(b) Scenarij 7 - ulica u predgrađu

Slika 1.3. Primjer četiri različita scenarija među podatcima za treniranje (1/2)[8]

(c) Scenarij 19 - pješačka zona

(d) Scenarij 20 - autocesta

Slika 1.3. Primjer četiri različita scenarija među podatcima za treniranje (2/2)[8]

2. *3D Point Clouds*

   Podatci lidara. Svaki zapis odgovara jednoj slici. Nisu direktno korišteni u radu, ali su prethodno upotrijebljeni od KITTI tima za stvaranje *ground truth* podataka.

3. Anotacije slika za treniranje

   Informacije o objektima koji se nalaze na slikama. Po jedna tekstualna datoteka za svaku scenu. Sadrži podatke o rednom broju slike, klasi objekta, graničnom okviru, lokaciji objekta, udaljenosti od kamere te njegovoj veličini. Objektu može biti dodijeljena jedna od 8 klasa: automobil, kamion, kombi, pješak, osoba (koja sjedi), biciklist, tramvaj te ostalo. [6]

   Anotacije podataka za 3D lokalizaciju napravio je tim zaposlen specifično za te potrebe. Korišten je alat koji integrira slike i podatke lidar senzora. Manualno su dodani 3D granični okviri tako da precizno odgovaraju podatcima lidara, dok je kamera davala vizualni kontekst koristan u slučaju zaklonjenosti objekta. [9] Time je osigurana kvaliteta podataka koja ne bi bila zagarantirana da su anotacije dobivene putem *crowdsourcinga*.

Atribut	Iznos	Opis
Broj slike	2	Redni broj slike u sekvenci
ID	0	Jedinstven ID dodijeljen objektu za potrebe praćenja
Objekt	Car	Klasa detektiranog objekta
τ	0	Nalazi li se objekt unutar granica okvira slike 0 = potpuno vidljiv 1 = potpuno izvan okvira
Ω	0	Zaklonjenost ostalim objektima na slici 0 = potpuno vidljiv 1 = djelomično zaklonjen 2 = značajno zaklonjen 3 = nepoznato
α	-2.12	Kut promatranja objekta naprema smjera gledanja kamere [rad]
x_{min}	866.58	2D granični okvir Koordinate nasuprotnih vrhova okvira
y_{min}	189.73	
x_{max}	1241.00	
y_{max}	374.00	
H	1.38	Stvarna dimenzija objekta (visina, širina, dužina)[m]
W	1.51	
L	4.10	
X	3.37	Stvarna lokacija u odnosu na kameru [m]
Y	1.54	
Z	5.33	
ϕ	-1.58	Rotacija objekta oko vertikalne osi [rad]

Tablica 1.1. Podatci u zapisu jedne detekcije

4. Kalibracijski podatci

   Omogućuju precizno mapiranje podataka skupljenih iz različitih senzora. Među njima se nalaze intrinzični parametri kamere koji su potrebni za projekciju podataka iz 3D svijeta na 2D ravninu slike i obrnuto. Svaka sekvenca slika u tracking podatcima ima svoju odgovarajuću teksualnu datoteku koja sadrži sljedeće komponente:

   - Intrinzični i ekstrinzični parametri kamere (P_0, P_1, P_2, P_3)

     Predstavljaju projekcijske matrice za četiri kamere (dva stereo para kamera). U matricama se nalaze:

     - f_x, f_y - fokalna duljina duž x i y osi

       Opisuju mapiranje 3D točaka u 2D prostor. Ovise o rezoluciji kamere i svojstvima leće.

– c_x, c_y - pomaci glavne točke
Koordinate (u pikselima) točke gdje se optička os kamere križa s ravninom slike.[10]

– t_x, t_y, t_z - translacijske komponente za ekstrinzičnu kalibraciju senzora

Projekcijsku matricu se može rastaviti na njezin intrinzični dio **K** i ekstrinzični dio **T**:

$$\mathbf{P} = \left[\begin{array}{ccc:c} f_x & 0 & c_x & t_x \\ 0 & f_y & c_y & t_y \\ 0 & 0 & 1 & t_z \end{array}\right] = \left[\begin{array}{c:c} \mathbf{K} & \mathbf{T} \end{array}\right].$$

- Rektifikacijska matrica
Matrica dimenzija 3x3 koja rektificira sliku radi poravnavanja za stereo spajanje slika.

- Transformacijske matrice
Matrice za transformaciju iz IMU (Internal Measurement Unit) sustava u koordinatni sustav lidara, te za transformaciju iz lidar sustava u koordinatni sustav kamere.

U daljnjem radu korištena je matrica P_2 koja odgovara lijevoj kameri u boji. Budući da se u radu ne koriste podatci lidar senzora, niti odgovarajući stereo par kamere, nisu korištene matrice za rektifikaciju i transformaciju.

5. GPS/IMU (Internal Measurement Unit) podatci
Pokazuju stanje vozila i mjernih instrumenata u svakom vremenskom koraku. GPS/IMU podatci se snimaju frekvencijom 100 Hz (za razliku od kamerinih 10 Hz). Stoga je već napravljeno mapiranje KITTI slika s očitanjima GPS/IMU sustava na način da su uzete informacije koje imaju najbliži *timestamp* toj slici. U najgorem slučaju dolazi do vremenske razlike od 5 ms. Pri svakom očitanju spremljeno je 30 različitih vrijednosti: geografska širina i dužina, nadmorska visina, orijentacija vozila, brzina i akceleracija, te podatci o preciznosti i radu satelita.[6]

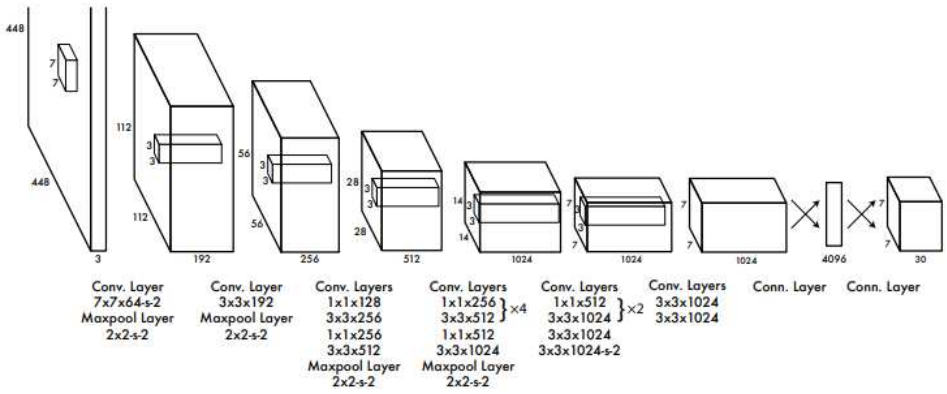
2. Estimacija pozicije i brzine iz monokularnih zapisa

2.1. Detekcija objekata

YOLO (en. *You only look once*) je algoritam za detekciju objekata koji je značajno pridonio području računalnog vida. Tradicionalne metode (Faster R-CNN ili SSD) odjeljuju proces detekcije objekata u nekoliko koraka što ih čini računalno zahtjevnima.[11] Prvi korak je predlaganje regija interesa. Često se za to koriste sidreni okviri ili specijalizirani algoritmi (Selective Search). Nakon što su prijedlozi doneseni, neuronska mreža dorađuje njihovu lokaciju i veličinu. Svaka redefinirana regija se potom klasificira u jednu od diskretno odabranih kategorija. Za razliku od tradicionalnih metoda, YOLO pojednostavljuje proces na način da sve tretira kao regresijski problem na način da predviđa kontinuirane numeričke vrijednosti za koordinate graničnih okvira, razinu pouzdanosti i vjerojatnost pripadanja određenoj klasi. Dakle, umjesto klasifikacije u jednu grupu, računa se vjerojatnost da objekt pripada svakoj od prisutnih klasa.[12]

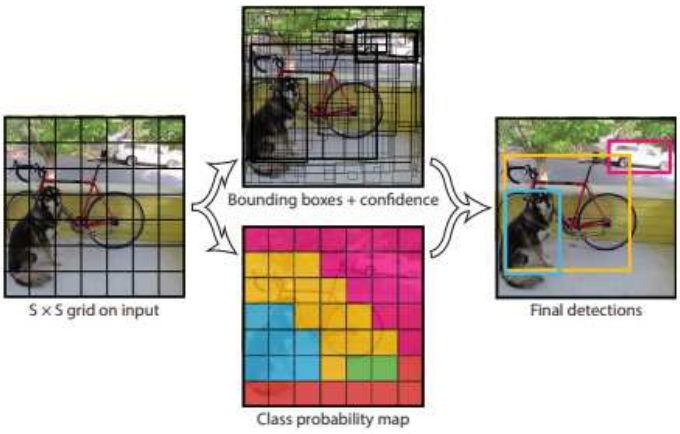
Arhitektura se bazira na konvolucijskoj neuronskoj mreži (CNN). Sadrži 24 konvolucijska sloja za ekstrakciju značajki, 4 *max-pooling* sloja za reduciranje prostornih dimenzija i 2 potpuno spojena sloja za konačne predikcije.[12]

Prije ulaza u konvolucijsku mrežu, slika se transformira na veličinu 448x448. YOLO algoritam dijeli sliku na rešetku dimenzija S x S. Ako centar objekta leži unutar ćelije u rešetci, onda je ta ćelija zadužena za detekciju tog objekta. Verzije modela od YOLOv4 nadalje nemaju jednoznačno postavljen iznos za S, nego koriste nekoliko veličina rešetke kako bi bolje detektirali objekte različitih veličina. Svaka ćelija predviđa B sidrenih okvira i razine pouzdanosti za te okvire. Većina YOLO modela, kao i onaj korišten u ovom radu, koristi 3 sidrena okvira po ćeliji. Razina pouzdanosti se računa tako da se uzme u



Slika 2.1. Arhitektura mreže[12]

obzir sadrži li okvir taj specifični objekt te koliko se predviđeni okvir preklapa s istinitim. $Pr(Object) \cdot IOU_{truth}^{prediction}$. Uz to, svaka ćelija će jednom predvidjeti uvjetne klasne vjerojatnosti $Pr(Class_i|Object)$. Njih će biti onoliko koliko ima različitih klasa, a rezultat se koristi u cijeloj ćeliji za sve sidrene okvire. Uvjetne vjerojatnosti se potom množe s razinom pouzdanosti kako bi se dobile nove razine pouzdanosti specifične za određenu klasu. Prema tom rezultatu se rangiraju predikcije te se donosi konačna odluka o pripadnosti.[12]



Slika 2.2. Podjela na rešetku dimenzija SxS, predikcija sidrenih okvira i klasnih vjerojatnosti[12]

U ovom zadatku korištena je već istrenirana verzija YOLOv5 modela za detekciju objekata[5]. Nisu rađene nikakve modifikacije modela niti fine-tuning. Treniranje je odrađeno na velikom skupu podataka COCO, koji sadrži razne kategorije objekata u različitim okruženjima, ne nužno povezanima sa scenarijima u prometu[13]. Skinute su težine istreniranog modela koje su direktno iskorištene za lokalizaciju objekata na KITTI skupu podataka. Budući da su rezultati lokalizacije bili dovoljno dobri, preskočen je ra-

čunalno i vremenski zahtjevan korak *fine-tuninga*.

2.2. Lokalizacija i praćenje objekata

Lokalizacija se odnosi na proces određivanja pozicije objekta na slici. Sami podatci stekнuti 2D lokalizacijom, odnosno pozicija objekta u koordinatama slike, uglavnom nisu dovoljni za primjenu u autonomnoj vožnji, robotici i sl. Zbog toga se ulazi u proces 3D lokalizacije kojim se premošćuje jaz između prepoznavanja objekta i razumijevanja prostora. Cilj je dobiti informaciju lokaciji objekta u stvarnom prostoru kako bi budući sustav mogao biti u interakciji sa svojim okruženjem.
Pri korištenju samo monokularnih slika, najveća prepreka je procjena dubine. Obično se ta informacija dobije drugim senzorima poput lidara.[2] Još jedan način je koristiti stereo par kamera koji istu sliku uzima iz malo različitih kuteva. Kasnije se usporedbom značajki i triangulacijom dobije informacija o dubini.[14]
U nedostatku tih podataka jedan od načina kako doći do željenih informacja je korištenjem geometrijskih ograničenja i apriornog znanja.[15]

2.2.1. Geometrijska projekcija

Uvođenje informacija o geometrijskom apriornom znanju prvi je korak u projekciji iz 2D svijeta u 3D svijet. Apriorno znanje može imati razne oblike, od ograničenja scene, pretpostavki o geometriji objekata do statističke distribucije veličina objekata. Jedna od metoda, koja je korištena i u ovom radu, je estimacija visine 2D i 3D graničnog okvira i potom računanje dubine preko projekcijske formule:

$$dubina = \frac{h_{3d} \cdot f}{h_{2d}}, \tag{2.1}$$

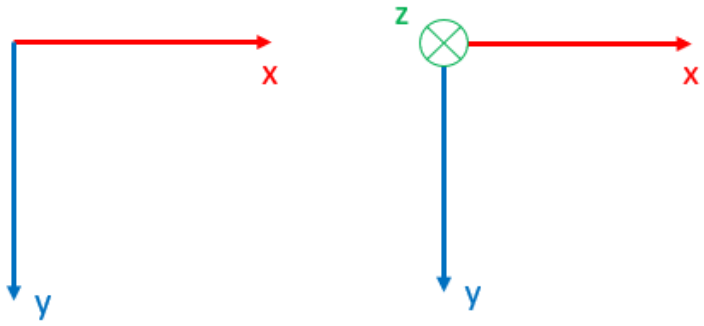
gdje je f fokalna duljina kamere[15]. Visina 2D graničnog okvira je već prisutna kao izlaz iz YOLO modela. Visinu 3D graničnog okvira možemo promatrati kao visinu objeka u stvarnom svijetu. Najjednostavniji pristup je uvesti pretpostavku za visinu objekta. Budući da dubina visoko korelira s pretpostavljenim visinama, znači da će sve potencijalne greške biti reflektirane u procjeni dubine.[15] Na slici 3.1. prikazane su odabrane

klase za koje će se raditi lokalizacija. Za svaku klasu uzeta je vrijednost za prosječnu visinu. Prema toj vrijednosti će se za sve pripadnike te klase raditi proračun dubine. Visina kako automobila, tako i ljudi i drugih objekata odstupa od prosječnih vrijednosti, tako da se očekuju greške u lokalizaciji. S obzirom da su sve slike snimljene u Karlsruheu u Njemačkoj, za prosječne visine automobila[16] i ljudi[17] uzete su prosječne visine iz te zemlje. Za visine autobusa[18] i kamiona[19] su uzete visine jednog od modela marke Mercedes.

Klasa	**Uzeta prosječna visina [m]**
Automobil	1.550
Autobus	3.095
Kamion	3.510
Osoba	1.730

Tablica 2.1. Odabrane vrijednosti za visine klasa

Projekcijske formule su bazirane na modelu kamere s točkastim otvorom (en. *Pinhole Camera Model*). To je pojednostavljeni prikaz kamere gdje svjetlo iz 3D scene prolazi kroz jednu točku te radi projekciju na 2D ravninu. Parametri kamere, fokalna dužina i pomaci glavne točke, povezuju 2D koordinate u pikselima s koordinatama u 3D svijetu u metrima. Pri projekciji potrebno je voditi računa o orijentaciji koordinatnih sustava slike i kamere. Orijentacija x i y osi se podudara (slika 2.3.), tako da nije potrebno vršiti rotacije prilikom projekcije.



(a) Koordinatni sustav slike **(b)** Koordinatni sustav kamere

Slika 2.3. Usporedba koordinatnih sustava slike i kamere

Koristeći homogeni zapis koordinata[20], koordinate točke na slici (x_{slika}, y_{slika}) se

prebacuju u oblik (x, y, w):

$$x_{slika} = \frac{x}{w}, \qquad y_{slika} = \frac{y}{w}. \tag{2.2}$$

Temelj lokalizacijskog postupka je projekcijska formula kamere:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \mathbf{K} \cdot \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}, \tag{2.3}$$

gdje je $\begin{bmatrix} X_c & Y_c & Z_c \end{bmatrix}^T$ vektor stvarnih koordinata objekta u koordinatnom sustavu kamere, a K je intrinzična matrica kamere, čiji su parametri prisutni u kalibracijskim podatcima KITTI *dataseta*[6]:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.4}$$

Jednostavnim matričnim množenjem dobiju se sljedeće relacije:

$$x = f_x \cdot X_c + c_x \cdot Z_c, \qquad y = f_y \cdot Y_c + c_y \cdot Z_c. \tag{2.5}$$

Supstitucijom pomoću izraza 2.2 dobije se direktan odnos koordinata slike u pikselima i koordinata kamere u metrima:

$$x_{slika} \cdot Z_c = f_x \cdot X_c + c_x \cdot Z_c, \qquad y_{slika} \cdot Z_c = f_y \cdot Y_c + c_y \cdot Z_c. \tag{2.6}$$

Dubina Z_c već je poznata iz relacije 2.1 tako da imamo formulu za izračun pozicije objekta na x i y osima:

$$X_c = Z_c \cdot \frac{x_{slika} - c_x}{f_x}, \qquad Y_c = Z_c \cdot \frac{y_{slika} - c_y}{f_y}. \tag{2.7}$$

2.2.2. Dodjeljivanje identifikatora objektima

Estimacija brzine objekta zahtijeva ne samo identifikaciju objekta u individualnim slikama nego i određivanje njegovog pomaka kroz vrijeme. Algoritmi praćenja to postižu asociranjem detektiranih objekata kroz uzastopne okvire. Time objekt dobiva neprekidnu trajektoriju koja pomaže u mjerenju brzine. S obzirom da YOLO ne daje informaciju o praćenju objekta kroz okvire, nužno je upotrijebiti algoritam specijaliziran za taj posao. U ovom radu, za praćenje objekata upotrijebljen je algoritam SORT (en. *Simple Online and Realtime Tracking*). SORT koristi Kalmanov filtar za predikciju stanja te Mađarski algoritam[21] i presjek preko unije[22] za povezivanje detektiranih objekata kroz slijed okvira. Kalmanov filtar estimira poziciju objekta i brzinu prema prethodnom stanju. Time je moguće predviđanje lokacije u trenutcima kada je objekt zaklonjen ili nedostaje detekcija. Loš efekt Kalmanovog filtra je što je moguća mala promjena graničnih okvira u obliku pomaka centra ili proširivanja i sužavanja. Računanjem presjeka preko unije (IOU) dobije se metrika za asocijaciju objekata. Mađarski algoritam[21] onda radi asocijaciju na način na minimizira grešku, odnosno u ovom slučaju maksimizira presjek. Velika prednost SORT-a je njegova jednostavnost i brzina, što ga čini pogodnim za sustave s ograničenim resursima. Njegova najveća limitacija je upravo vrlo jednostavan način asociranja objekata koji u obzir uzima samo granične okvire. Bez informacija koje nam pruža slika (boja, orijentacija, oblik) moguće su krive asocijacije objekata, posebice u slučajevima kada dolazi do preklapanja na slici ili nepredvidivog kretanja. [23]

2.3. Estimacija brzine

2.3.1. Kalmanov filtar

Kalmanov filtar je algoritam za procjenu stanja u dinamičkim sustavima. Posebno je koristan kada su mjerenja zašumljena ili je dinamika sustava podložna izmjenama. Estimacija stanja sustava radi se rekurzivnim algoritmom te je moguća čak i kada su mjerenja neprecizna ili nepotpuna. Stanje sustava čini matematički prikaz sustava u datom trenutku. U kontekstu estimacije brzine, može ga činiti npr. lokacija, brzina ili akcele-

racija. Model sustava i mjerenja prikazuju se dvjema linearnim jednadžbama:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \quad \mathbf{w}_{k-1} \sim \mathcal{N}(0, \mathbf{Q}_{k-1}) \tag{2.8}$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k) \tag{2.9}$$

gdje

- $\mathbf{x}_k \in \mathbb{R}^n$ je stanje sustava, a $\mathbf{z}_k \in \mathbb{R}^m$ je mjerenje
- $\mathbf{w}_{k-1}$ i $\mathbf{v}_k$ su šumovi procesa i mjerenja
- $\mathbf{u}_{k-1}$ je poznata ulazna/kontrolna varijabla
- **B** je matrica kontrolnog ulaza
- **A** je prijelazna matrica modela sustava
- **H** je matrica modela mjerenja.

Filtar se sastoji od dva koraka:

1. **Predikcija** - Prema poznatoj dinamici sustava predviđa se stanje u sljedećem vremenskom koraku. Model predstavlja tranzicijska matrica u kojoj je opisano kako se stanje sustava mijenja kroz vrijeme.

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1}^+ + \mathbf{B}\mathbf{u}_{k-1}, \tag{2.10}$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}^+\mathbf{A}^\top + \mathbf{Q}_{k-1}. \tag{2.11}$$

   Oznaka minusa iznad varijable znači da se radi o predviđenim estimacijama, dakle prije nego postane dostupno stvarno mjerenje. Matrica $\mathbf{P} \in \mathbb{R}^{n\times n}$, predstavlja kovarijancu greške estimacije, a njezina vrijednost se mijenja svakim ažuriranjem. Matrice $\mathbf{Q} \in \mathbb{R}^{n\times n}$ i $\mathbf{R} \in \mathbb{R}^{m\times m}$, označavaju kovarijancu šuma procesa odnosno mjerenja. Mijenjanjem vrijednosti u njima određuje se koliko će se vjerovati procesu, a koliko dobivenim mjerenjima. Obično se inicijaliziraju na početku te se smatra da se ne mijenjaju s vremenom.

2. **Korekcija** - Filtar koristi nova mjerenja dobivena od senzora za korekciju svoje predikcije. Prije same korekcije, pomoću izraza 2.12 računa se Kalmanovo poja-

čanje. Ono određuje koliko estimacija $\hat{\mathbf{x}}_k^-$ iz koraka predikcije treba biti korigirana mjerenjima $\mathbf{z}_k$:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^\top \left(\mathbf{H} \mathbf{P}_k^- \mathbf{H}^\top + \mathbf{R}_k \right)^{-1}, \tag{2.12}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_k^- \right), \tag{2.13}$$

$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H} \right) \mathbf{P}_k^-. \tag{2.14}$$

Kada je model siguran u predviđeno stanje $\hat{\mathbf{x}}_k^-$, matrica $\mathbf{P}_k^-$ će imati manje vrijednosti u sebi. To znači da će Kalmanovo pojačanje biti relativno malo, odnosno da će se prema relaciji 2.13 manju važnost dati mjerenju $\mathbf{z}_k$. U suprotnom slučaju, kada model nije siguran u svoju estimaciju, Kalmanovo pojačanje je veće te će veću težinu imati korekcija $\left(\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_k^- \right)$. Matrica kovarijance stanja se također ažurira nakon dobivanja mjerenja, kako bi reflektirala promjenu nesigurnosti.

Tim ciklusom Kalmanov filtar ažurira svoje estimacije dolaskom novih mjerenja. To ga čini iznimno prikladnim za aplikaciju u stvarnom vremenu. Primijetimo da filtar može raditi i u odsutstvu mjerenja. To je izuzetno bitno u računalnom vidu u trenutcima kada objekt ne bude detektiran zbog zaklonjenosti ili nesavršenosti modela detekcije. U tim uvjetima bitno je da sustav i dalje može predvidjeti gdje će se objekt naći sve dok mjerenja ne postanu ponovno dostupna.

2.3.2. Matematički model

Za potrebe filtra, vektor mjerenja $\mathbf{z}$, koji sadrži samo mjerenja pozicije nastala geometrijskom projekcijom, proširen je "mjerenjima" brzine. Brzina objekta u svakom koraku izračunata je kao pomak na dvjema slikama kroz vrijeme između dvije slike $\Delta t = 0.1s$:

$$v_{x,k} = \frac{x_k - x_{k-1}}{\Delta t}, \qquad v_{y,k} = \frac{y_k - y_{k-1}}{\Delta t}, \qquad v_{z,k} = \frac{z_k - z_{k-1}}{\Delta t}. \tag{2.15}$$

U vektor stanja sustava uključeni su pozicija i brzina objekta, te je prema relaciji 2.10

određena tranzicijska matrica **A**.

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix}, \qquad \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Ovime se na temelju prošlog stanja radi predikcija. Pretpostavka modela pri predikciji je da je brzina objekta konstantna kroz vrijeme. Kako su jedina prava mjerenja koja postoje mjerenja pozicije, nema dovoljno informacija za modeliranje promjene brzine. Za to bismo trebali imati, primjerice, podatke o akceleraciji objekta koje bismo tretirali kao ulaznu varijablu $\mathbf{u}_{k-1}$.

Matrica mjerenja je poveznica predviđenog stanja i mjerenja:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Iako se za brzinu smatra da je konstantna tijekom predikcije, ona se i dalje može promijeniti dolaskom mjerenja. Korekcija se ne radi samo nad pozicijama nego i nad brzinama. Bilo kakva promjena brzine nastaje isključivo zbog odstupanja mjerenja od predviđenog stanja. U slučaju nedostatka mjerenja, sustav treba nastaviti predviđati stanje na temelju dinamike određene matricom **A**. U stvarnosti postoje dva razloga nestanka mjerenja. Prvi je da se dogodila greška u detekciji ili praćenju objekta. U tom slučaju želimo da sustav nastavi predviđati stanje kako bi se prilikom ponovnog dolaska mjerenja objekt lakše uklopio u estimirano stanje. Drugi slučaj je kada objekt izađe iz kadra te ne bude ponovno detektiran. Tada nemamo koristi od praćenja njegovog stanja. Kako ne bi-

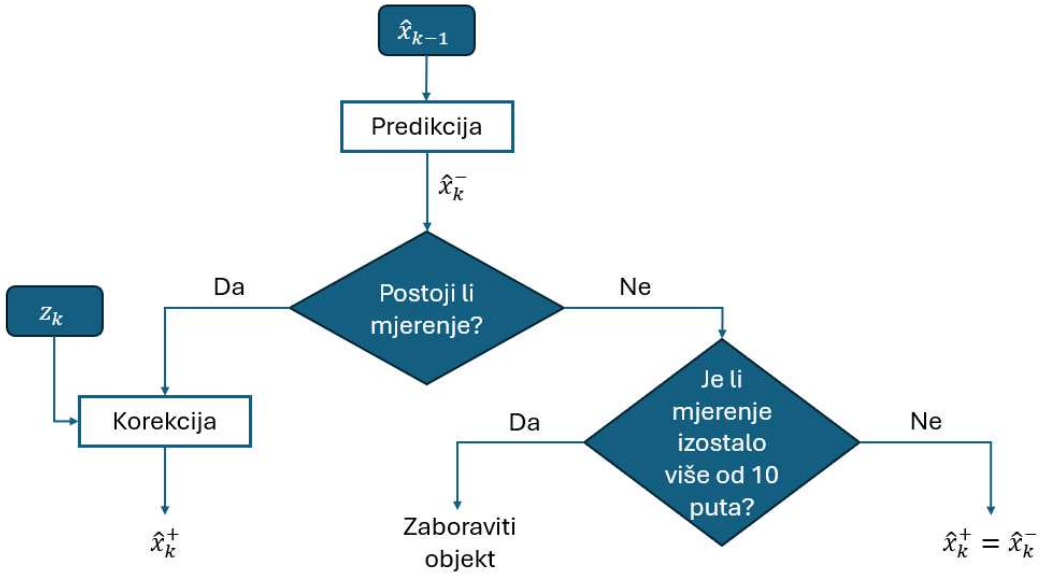
smo preopteretili sustav predikcijama koje u odsutstvu mjerenja postaju sve nesigurnije, postavljen je prag od 10 okvira. Ako objekt ne bude reidentificiran unutar 10 slika od nestanka, sustav briše njegovo stanje iz memorije.

Matricom **Q** definira se koliko ćemo vjerovati modelu, odnosno predikcijama nastalim pomoću tranzicijske matrice. Njezini elementi predstavljaju kovarijancu među različitim stanjima. Ukoliko su dva stanja međusobno neovisna, njihova korelacija iznosi 0.

$$\mathbf{Q} = \begin{bmatrix} q_{xx} & q_{xy} & q_{xz} & q_{xv_x} & q_{xv_y} & q_{xv_z} \\ q_{xy} & q_{yy} & q_{yz} & q_{xv_x} & q_{xv_y} & q_{xv_z} \\ q_{xz} & q_{yz} & q_{zz} & q_{xv_x} & q_{xv_y} & q_{xv_z} \\ q_{xv_x} & q_{xv_x} & q_{xv_x} & q_{v_xv_x} & q_{v_xv_y} & q_{v_xv_z} \\ q_{xv_y} & q_{xv_y} & q_{xv_y} & q_{v_xv_y} & q_{v_yv_y} & q_{v_yv_z} \\ q_{xv_z} & q_{xv_z} & q_{xv_z} & q_{v_xv_z} & q_{v_yv_z} & q_{v_zv_z} \end{bmatrix} = \begin{bmatrix} 10^{-7} & 0 & 0 & 10^{-6} & 0 & 0 \\ 0 & 10^{-7} & 0 & 0 & 10^{-6} & 0 \\ 0 & 0 & 10^{-7} & 0 & 0 & 10^{-6} \\ 10^{-6} & 0 & 0 & 10^{-4} & 0 & 0 \\ 0 & 10^{-6} & 0 & 0 & 10^{-4} & 0 \\ 0 & 0 & 10^{-6} & 0 & 0 & 10^{-4} \end{bmatrix}.$$

Elementi na dijagonali predstavljaju nesigurnost vezanu uz varijable stanja. Nesigurnost pozicija (q_{xx}, q_{yy}, q_{zz}) postavljena je na vrijednost 10^{-7}, što znači da filtar jako vjeruje modelu u estimaciji koordinata x, y i z. Nešto veća vrijednost od 10^{-4} odabrana je za nesigurnost estimacije brzine $(q_{v_xv_x}, q_{v_yv_y}, q_{v_zv_z})$. To je napravljeno iz razloga što model pretpostavlja konstantnu brzinu što ne odgovara stvarnosti. Time filtru dajemo više fleksibilnosti za ažuriranje brzine kada pristignu mjerenja. Članovi van dijagonale predstavljaju korelaciju dviju varijabli stanja. Faktori korelacije pozicija i njima po osi odgovarajućih brzina $(q_{xv_x}, q_{yv_y}, q_{zv_z})$ su postavljeni u 10^{-6}, dok se za sve ostale pretpostavlja da ne koreliraju te zbog toga iznose 0.

$$\mathbf{R} = \begin{bmatrix} 10^{-3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{-3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^{-2} \end{bmatrix}.$$



Slika 2.4. Shematski prikaz rada sustava za estimaciju u trenutku k

Matrica **R** daje informaciju o nesigurnosti mjerenja. Nesigurnost mjerenja pozicije iznosi 10^{-3}. Budući da je brzina dobivena pomoću mjerenja pozicije, propagira se i greška u izračunu pozicije. Zato su nesigurnosti mjerenja brzina uvećane za faktor 10.

Prilikom inicijalizacije filtra potrebno je odrediti početne uvjete za vektor stanja, odnosno za brzinu i poziciju objekta. Početni uvjeti postavljeni su kao:

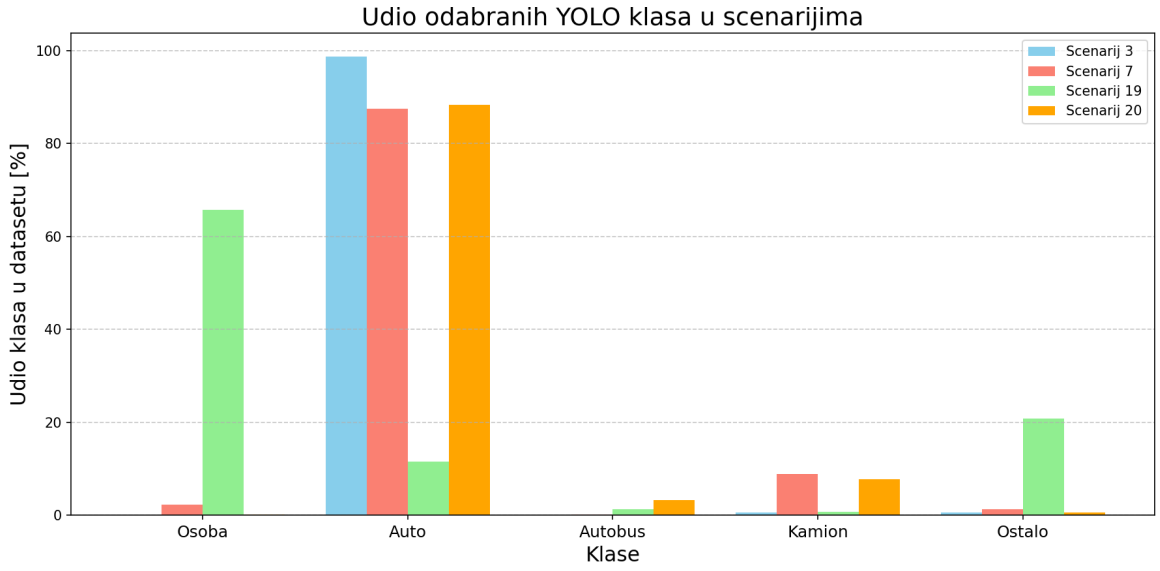
$$\mathbf{x_0} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ v_{x_0} \\ v_{y_0} \\ v_{z_0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 10 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

vodeći se pretpostavkom da će objekt biti detektiran tek kada mu se kamera dovoljno približi.

3. Rezultati

3.1. Detekcija objekata

Već istrenirani model YOLOv5 skinut je s Github stranice Ultralyticsa.[5] Budući da nije rađeno nikakvo treniranje modela; detekcija, a i ostatak rada, koristit će samo slike za treniranje. Prednost toga je što slike za treniranje, za razliku od onih za testiranje, imaju svoje odgovarajuće anotacije tako da je za njih moguće usporediti rad modela sa stvarnom situacijom. YOLO kao izlaz daje 2D granični okvir te klasu objekta. Te klase se ne podudaraju u potpunosti s klasama određenim u KITTI *datasetu*. YOLO ima čak 80 klasa koje može detektirati na slikama.[5] Većina njih nije relevantna za prometne scenarije, ali ih model svejedno detektira. Stoga je potrebno filtrirati detekcije, na način da se odbace detekcije objekata poput klupa, biljaka, torbi. Zadržane su samo klase koje sadrže objekte u pokretu za koje se može povući jasna paralela između KITTI klasa i YOLO klasa - automobil, osoba, kamion, autobus.

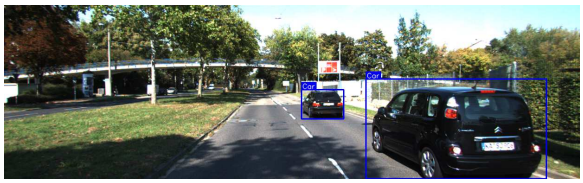


Slika 3.1. Udio zadržanih klasa u 4 odabrana scenarija

Rad modela je testiran na nekoliko različitih prometnih scenarija kako bi se procijenila njegova moć detekcije u različitim situacijama. Paralelno s time, izvučene su anotacije iz tekstualnih datoteka te su nacrtane na odgovarajućim slikama radi usporedbe. Na slici 3.1. vidljiv je udio 6 odabranih YOLO klasa u 4 promatrana scenarija. U 3 od 4 scenarija, veliku većinu objekata čine automobili. Samo u scenariju 19, smještenom u pješačkoj zoni bez mnogo cestovnog prometa, većinu detektiranih objekata čine pješaci. Preko 20% detektiranih objekata u scenariju 19 čine nesvrstani objekti. Među njima se nalaze nepomični predmeti uz samu cestu: parkirani bicikli, biljke, stolovi i stolice od restorana i slično. Ti objekti će biti zanemareni u ostatku rada pri lokalizaciji i procjeni brzine. Promatrajući detekcije na scenariju 7, možemo opaziti da YOLO nije uspio detektirati aute koji se nalaze dalje od kamere, a ujedno i djelomično zaklonjeni drugim objektima. Za bolje rezultate u takvim specifičnim slučajevima, trebalo bi razmotriti opciju *fine-tuninga* modela na KITTI slikama, čime se model adaptira specifičnostima tih slika.

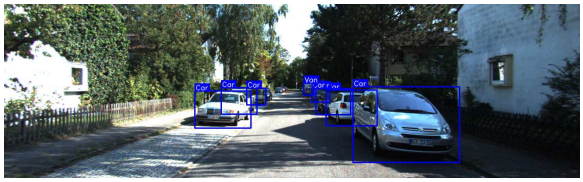
3.2. 3D lokalizacija objekata

Provedena je lokalizacija detektiranih objekata te su kao rezultat dobivene koordinate objekta (X, Y, Z) u metrima, u koordinatnom sustavu kamere prikazanom na slici 2.3. Kako bi se provjerila točnost lokalizacije, izračunate vrijednosti su uspoređene s *ground truth* vrijednostima iz anotacija. Za povezivanje detektiranog objekta s njegovom anotacijom, upotrijebljena je metrika presjek preko unije (en. *Intersection Over Union*, IoU).[22] Metoda uzima cijelu površinu predviđenog graničnog okvira i radi usporedbu s graničnim okvirima iz KITTI podataka za treniranje. IoU se računa kao omjer presjeka dvaju okvira i površine pod njihovom unijom. Predickija se spaja s anotacijom ako IoU prelazi određeni prag, najčešće postavljen na 0.5. U ovome slučaju, budući da su neke scene vrlo gužvovite, postavljen je stroži prag od 0.7 za IoU. Posljedica toga je da možda neke detekcije neće biti spojene sa svojom anotacijom, ali se i smanjuje broj lažnih pozitiva pri spajanju. Zbog neusklađenosti klasa YOLO algoritma i klasa KITTI podataka, pri asocijaciji podataka odrađeno je mapiranje klasa kako ne bi došlo do krivih asocijacija u gužvovitim kadrovima. YOLO klasama autobus i kamion nije dodijeljen direktan ekvivalent iz razloga što sam algoritam nekada pomiješa te objekte pri detekciji.



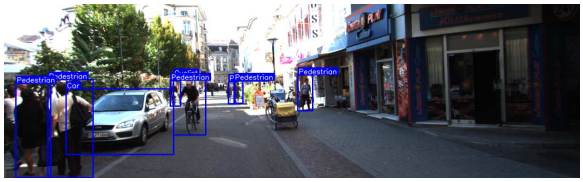
(a) Scenarij 3 - *ground truth*

(b) Scenarij 3 - detekcija



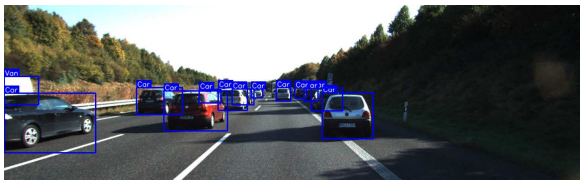
(c) Scenarij 7 - *ground truth*

(d) Scenarij 7 - detekcija

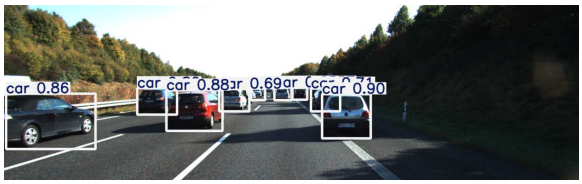


(e) Scenarij 19 - *ground truth*

(f) Scenarij 19 - detekcija



(g) Scenarij 20 - *ground truth*



(h) Scenarij 20 - detekcija

Slika 3.2. Usporeda *ground truth* informacija o graničnim okvirima s predviđenim okvirima dobivenim od YOLO-a

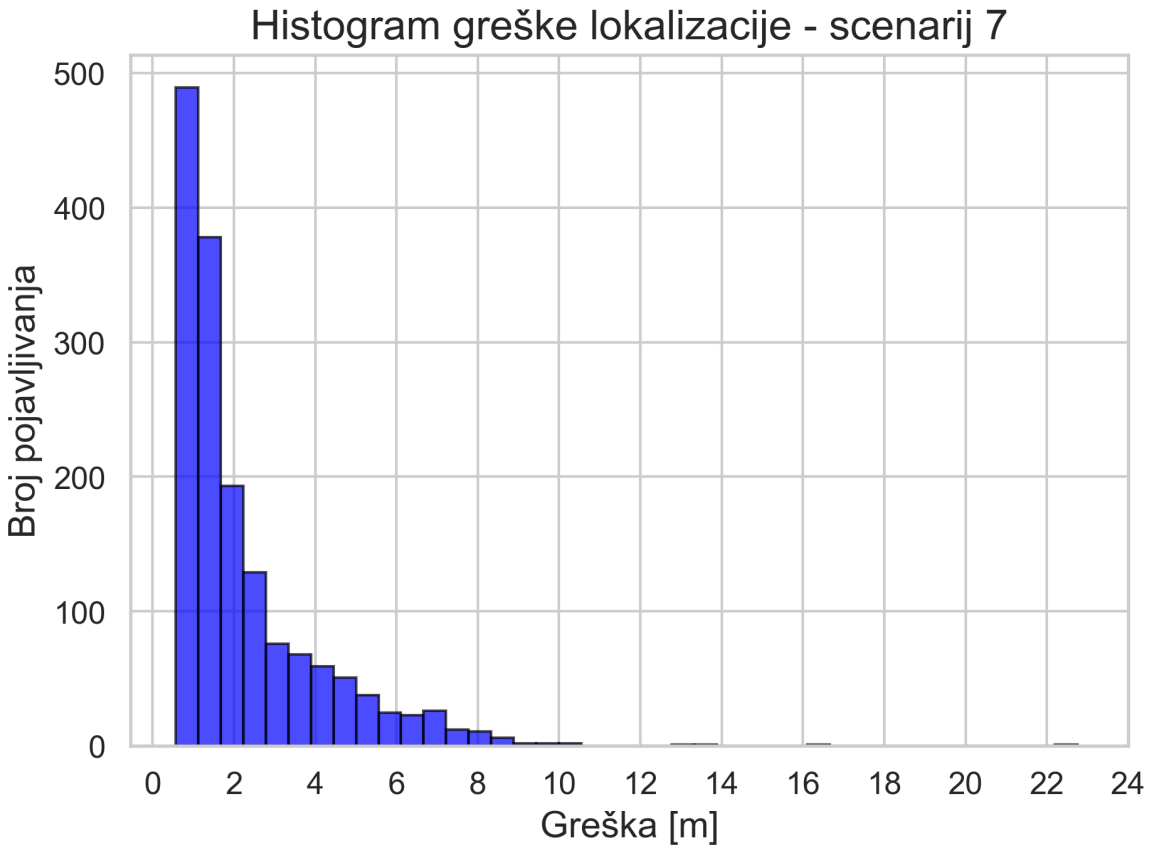
KITTI klase	**YOLO klase**
Pješak	Osoba
Automobil	Automobil
Kamion, Kombi	Kamion, Autobus

Tablica 3.1. Mapiranje klasa KITTI ↔ YOLO

Kao metrika za izračun greške lokalizacije korištena je euklidska udaljenost između dviju koordinata u trodimenzialnom prostoru:

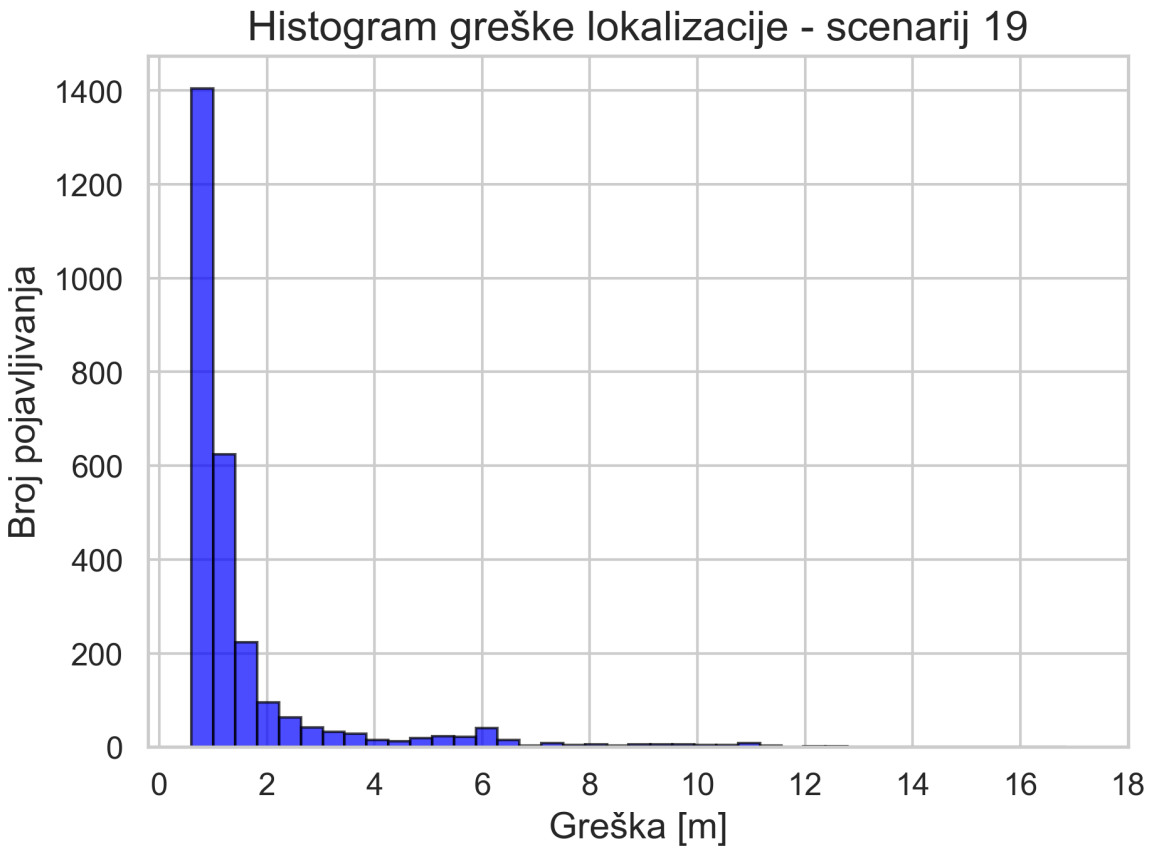
$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{3.1}$$

(a) Scenarij 3



(b) Scenarij 7

Slika 3.3. Greške lokalizacije izražene kao euklidska udaljenost između predikcije i *ground truth* vrijednosti



(a) Scenarij 19

(b) Scenarij 20

Slika 3.4. Greške lokalizacije izražene kao euklidska udaljenost između predikcije i *ground truth* vrijednosti

Na slikama 3.3. i 3.4. prikazana su četiri histograma grešaka lokalizacije. Svi grafovi imaju većinu grešaka u lijevoj polovici, s *peakom* između 0 m i 1 m, što znači da lokalizacijski sustav uglavnom dobro radi. Povećavanjem greške opada učestalost, odnosno velike greške se ne događaju toliko često. Sva četiri histograma imaju dugačak rep kojeg čine *outlieri*. U tablici 3.2. nalaze se podatci koji upisuju distribuciju greške lokalizacije - prosječna vrijednost, standardna devijacija te kvartili. Scenariji 3 i 19 imaju nižu prosječnu lokalizacijsku grešku i devijaciju. Njihova distribucija po kvartilima je također nešto povoljnija. U scenariju 19 se u 75% slučajeva greška nalazi ispod 1.43 m, dok u scenariju 3 ta granica iznosi 1.97 m. U scenarijima 7 i 20, od kojih oba imaju velik broj detektiranih vozila, a i samih detekcija; algoritam ima malo lošije rezultate. Prosječna greška iznosi preko 2.20 m za oba slučaja uz nešto raspršeniju distribuciju podataka. Treći kvartil je također na višoj vrijednosti - 2.80 m za scenrarij 7, te 2.53 m za scenarij 20.

Scenarij	e_{loc} **[m]**	σ **[m]**	**25% [m]**	**50% [m]**	**75% [m]**
3	1.81	1.49	0.88	1.30	1.97
7	2.27	1.90	1.01	1.53	2.80
19	1.62	1.70	0.88	1.00	1.43
20	2.47	2.92	1.08	1.61	2.53

Tablica 3.2. Greške lokalizacije - prosjek (e_{loc}), standardna devijacija (σ) i kvartili

Pitanje koje slijedi iz ovoga - je li ovaj način lokalizacije dovoljno dobar za upotrebu u stvarnom svijetu, primjerice u autonomnoj vožnji ili ADAS sustavima? Da bismo mogli na to odgovoriti potrebno je usporediti rezultate sa standardima u industriji. U tablici 3.3. se nalaze zahtjevi za grešku lokalizacije na gradskim prometnicama.[24] Greške su izražene u tri smjera kretanja. Lateralna os odgovara x-osi KITTI koordinatnog sustava, longitudinalna os z-osi, dok vertikalna os odgovara y-osi. Standardi za ADAS sustave su 0.15 m za lateralne i longitudinalne osi kretanja te 0.48 m za vertikalnu os. [24]

Os	Lateralna	Longitudinalna	Vertikalna
Greška [m]	0.15	0.15	0.48

Tablica 3.3. Zahtjevi za grešku lokalizacije[24]

Radi usporedbe sa standardnim zahtjevima, greške lokalizacije su rastavljene na svoje tri komponente te su prikazane u tablici 3.4. Niti jedna komponenta nema zadovoljen zahtjev za grešku lokalizacije niti u jednom scenariju. Longitudinalna greška, u smjeru

kretanja naprijed - nazad, ima najveća odstupanja. U najboljem scenariju greška iznosi 1.13 m što je znatno više od traženih 0.15 m. Uzrok ovih grešaka može biti više faktora, ali sve ukazuje na to da iako predloženi algoritam može locirati objekte, za upotrebu u prometu u stvarnom svijetu potrebno je napraviti određene prilagodbe. Najočitije rješenje je dovesti još podataka kako bi procjena dubine bila preciznija. To se može ostvariti u obliku stereo para slika ili drugih senzora poput radara i LiDAR-a. U slučaju da to nije moguće ili da se problem želi riješiti samo iz monokularnih slika, bilo bi dobro razmotriti drukčiji pristup modeliranju stvarne visine objekata. Upotreba jedne visine za cijelu klasu objekata je vrlo velika generalizacija. Jedan od načina kako bolje prilagoditi procjenu dubine je da se procjenu dubine tretira kao probabilistički proces. Uvođenjem funkcije gubitka koja bi penalizirala presamouvjerene, a netočne predikcije, modelirala bi se i dubina i pripadajuća nesigurnost.[15]

Scenarij	e_x [m]	e_y [m]	e_z [m]
3	0.55	0.75	1.37
7	0.47	0.74	1.95
19	0.30	0.83	1.13
20	0.50	0.76	2.14

Tablica 3.4. Lateralna (e_x), vertikalna (e_y), longitudinalna (e_z) komponenta izračunate greške lokalizacije

3.3. Praćenje objekata

Implementacija algoritma SORT preuzeta je s GitHuba [25]. Detektirani objekti s pripadajućim 2D graničnim okvirima poslani su algoritmu, koji svakom objektu dodjeljuje jedinstveni ID, analogno zapisu u tablici 1.1. Implementirana verzija SORT-a ima tri ključna promjenjiva parametra: *max_age*, *min_hits* i IOU prag[22].

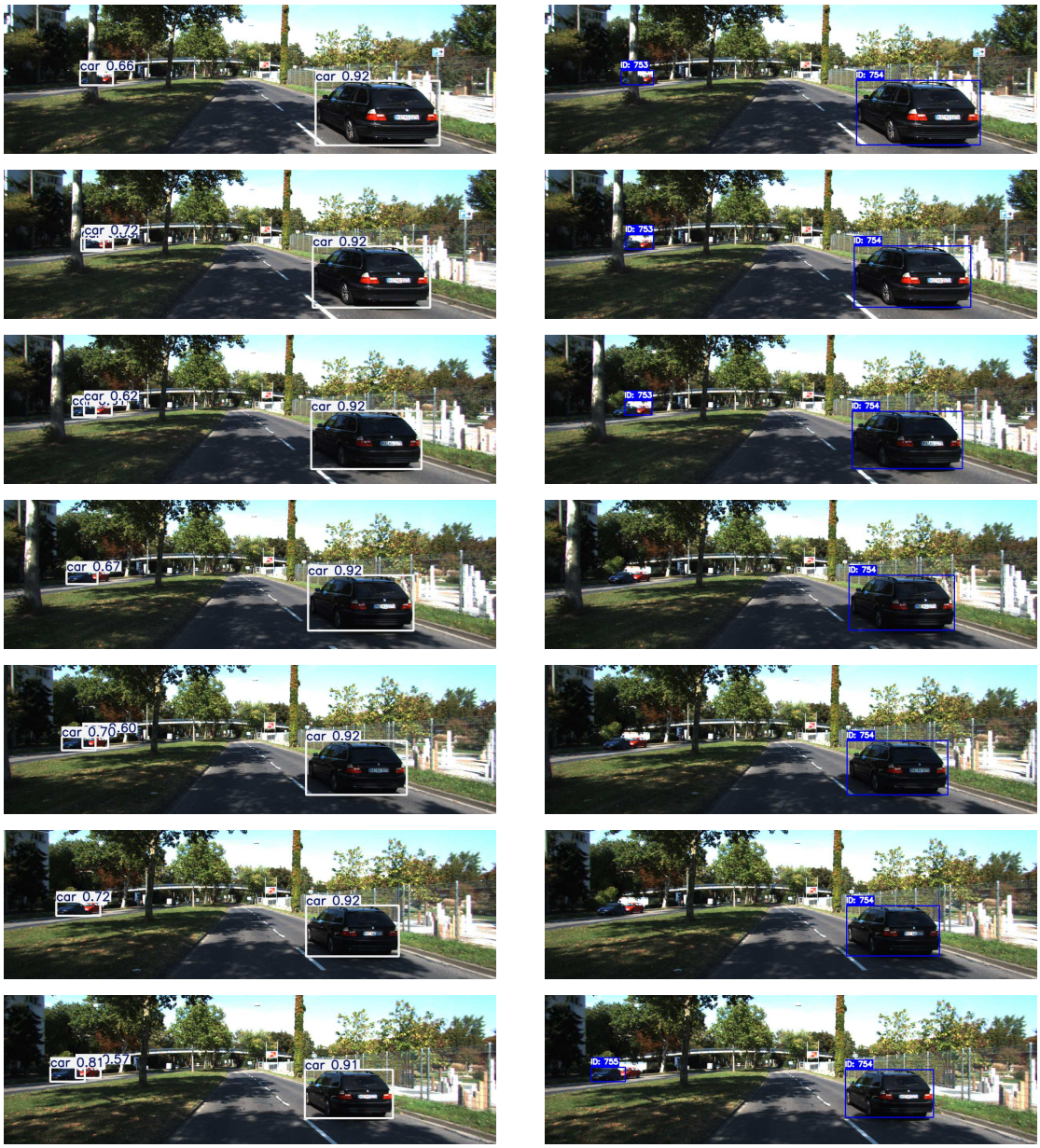
Parametar *max_age* određuje koliko dugo (u broju slika) će objekt ostati zapamćen bez asocijacije s drugom detekcijom. *Min_hits* specificira minimalan broj uzastopnih detekcija potreban da bi se objekt smatrao pouzdanim. IOU prag (*Intersection over Union*) koristi se za izračunavanje metrike tijekom Mađarskog algoritma, kojom se detekcije povezuju s postojećim tragovima.

Za sve analizirane scenarije korištene su iste vrijednosti parametara: $max\_age = 10$,

$min\_hits = 5$ i $IOU = 0.3$. Ove vrijednosti odabrane su kako bi se osigurala ravnoteža između praćenja objekata kroz privremene prekide i preciznosti asocijacija.

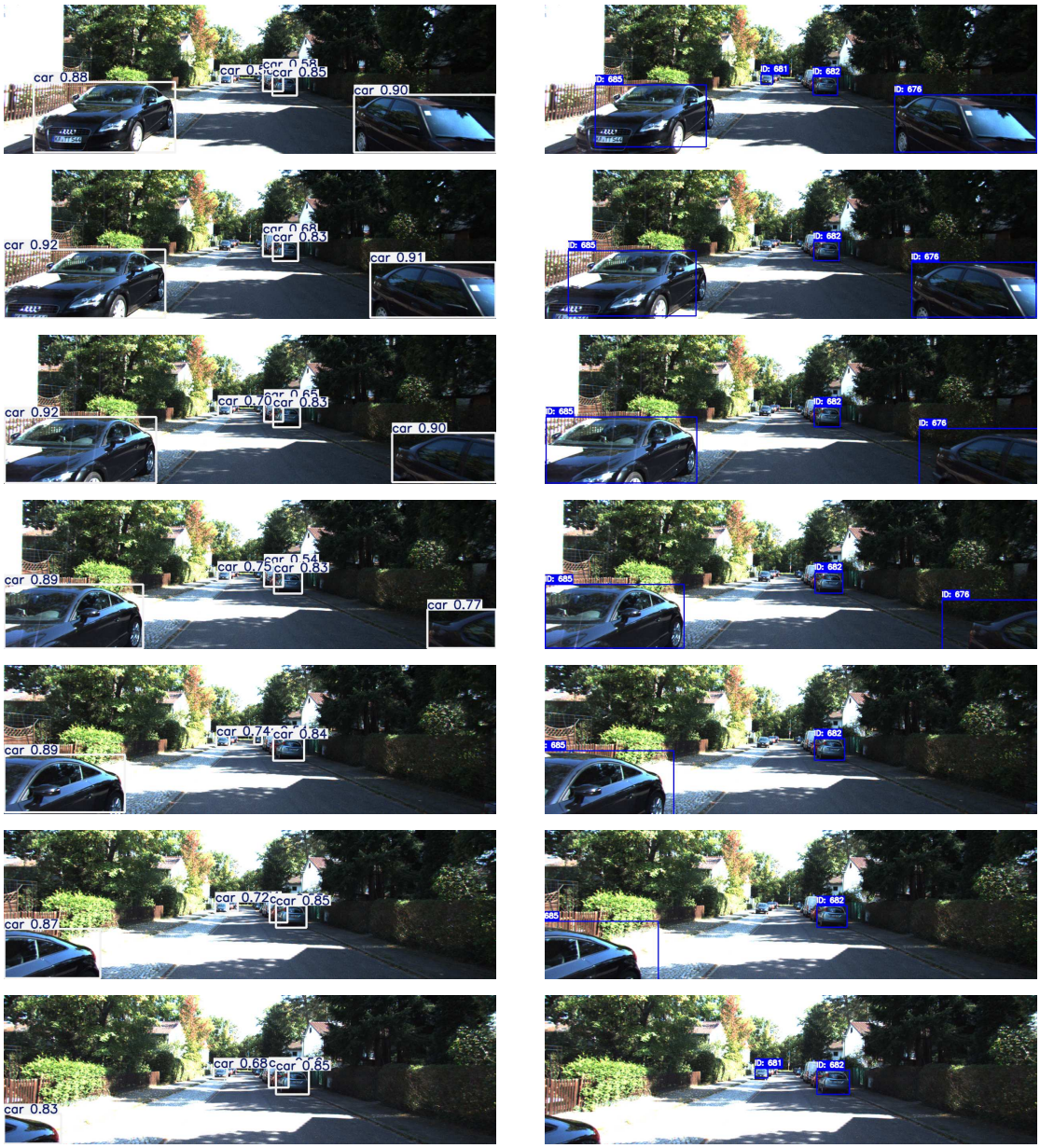
Budući da je SORT zbog svoje jednostavnosti sklon greškama u obliku miješanja ili propuštanja objekata, paralelno su prikazane 2D detekcije od YOLO-a, te SORT detekcije. Cilj toga je utvrditi događaju li se greške u praćenju samo zbog SORT-a ili i njegovi ulazni podatci imaju nedostatke.

Na slici 3.5. prikazane su iste slike iz 3. scenarija (redni brojevi 11 - 17). U desnom stupcu nalaze se izlazni podatci SORT algoritma (2D granični okvir i ID objekta) nacrtani na svojim odgovarajućim slikama. Kao kontrola, u lijevom stupcu, prikazani su isti ti trenutci samo s detekcijama YOLO modela. Treba primijetiti da SORT jako dobro prati objekt s ID-jem 754, koji je jasno vidljiv i nema preklapanja s drugim objektima. Usporedbom prvih dviju slika može se vidjeti da YOLO zbog okluzije na prvoj slici, dva auta detektira kao jedan objekt ID-ja 753. SORT, s obzirom da prima samo granične okvire kao ulaz, ne može to ispraviti nego prati grešku. Najveći problem na prikazanim slikama je propuštanje identifikacije na tri slike, iako je YOLO detektirao objekte. Konačno, na zadnjoj slici, SORT ponovno identifcira auto na lijevoj strani, ali ne radi reidentifikaciju nego mu dodjeljuje novi ID (755).



Slika 3.5. Scenarij 3 - usporedba YOLO detekcija i njima dodijeljenih ID-jeva. Lijevo: Izlaz iz YOLO modela. Desno: Izlaz iz SORT algoritma.

Na slici 3.6. prikazana je ista takva usporedba scenarija 7. Opet se može primijetiti lošiji rad u uvjetima zaklonjenosti objekata, gdje su dva auta cijelo vrijeme praćena kao jedan objekt (ID: 682). Opažamo i sposobnost algoritma da reidentificira objekte. Objekt s ID-jem 681 je nestao na 5 slika, odnosno 600 ms, nakon čega mu je ponovno dodijeljen isti ID. Na prvoj i drugoj slici, vidljiv je utjecaj Kalmanovog filtra koji je smanjio granični okvir objektu 685 naprema početnom okviru.



Slika 3.6. Scenarij 7 - usporedba YOLO detekcija i njima dodijeljenih ID-jeva. Lijevo: Izlaz iz YOLO modela. Desno: Izlaz iz SORT algoritma.

Praćenje objekata scenarija broj 20 prikazano je na slici 3.7. Prilikom praćenja ne dolazi do velikih grešaka. Jedini propust je kasnija identifikacija objekta 332 zbog okluzije drugim objektima. Razlika između ovog i drugih scenarija je što se svi objekti kreću u koloni u istome smjeru, malim brzinama. To znači da su relativne brzine objekata vrlo male te da ne dolazi do nepredvidivih pomaka, što je SORT-u velika prepreka.

Slika 3.7. Scenarij 20 - usporedba YOLO detekcija i njima dodijeljenih ID-jeva. Lijevo: Izlaz iz YOLO modela. Desno: Izlaz iz SORT algoritma.

Analizom rezultata praćenja objekata može se zaključiti da SORT uglavnom dobro obavlja zadatak identifikacije. Kao najveće prepreke su se pokazale zaklonjenost objekata i velika razlika u pomaku objekata. To znači da će algoritam bolje raditi za objekte koji se kreću u istom smjeru kao kamera, gdje će biti manje velikih pomaka, nego za objekte koji dolaze kameri u susret. Za bolje rezultate u tim slučajevima, potrebno je razmotriti složeniji algoritam poput algoritma Deepsort.[26] Deepsort ima SORT kao bazu, ali koristi tehnike dubokog učenja za ekstrakciju više podataka iz same slike. Time se ostvaruje bolja identifikacija objekata u zahtjevnijim uvjetima.

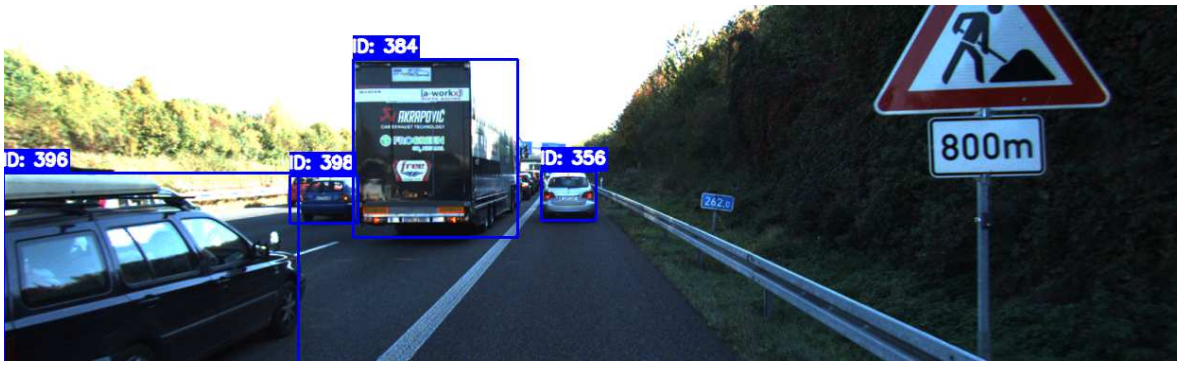
3.4. Estimacija pozicije i brzine

Opisana implementacija Kalmanovog filtra testirana je na sva 4 scenarija. Svaki identificirani objekt je praćen pojedinačno te su mu u konačnici estimirana pozicija i brzina na svakoj slici. Na odabranim scenarijima cilj je bio pronaći segmente gdje se može promatrati rad filtra u zahtjevnim uvjetima poput trenutaka kada nema mjerenja zbog greške u detekciji YOLO-a ili, vjerojatnije, greške u praćenju SORT-a (slika 3.6.). Drugi zahtjevni trenutci su kada su prisutni veliki šumovi u mjerenjima, nastali greškama pri geometrijskoj projekciji. Tada Kalmanov filtar ima efekt izglađivanja zašumljenih podataka, čime su estimirane vrijednosti pozicije i brzine znatno stabilnije.

Radi testiranja ispravnosti Kalmanovog filtra, na grafu prikaza brzine objekata, ažurirano stanje brzine nacrtano je zajedno s mjerenjima brzine. Brzine i pozicije su estimirane po svojim komponentama (x, y, z) te su tako i prikazane na grafovima. Osi brzina i pozicija odgovaraju koordinatnim osima kamere (slika 2.3.b). Budući da se rad bavi estimacijom pozicije i kretanja samo cestovnih vozila i pješaka, razumno je pretpostaviti da se najveći dio kretanja odvija po x i z-osi, s minimalnim kretanjem po y-osi. Stoga je u nastavku rađena analiza kretanja samo po lateralnim i longitudinalnim osima. Važno je napomenuti da su i pozicija i brzina izražene u odnosu na kameru. To znači da će brzina objekta zapravo biti razlika brzina objekta i kamere. Dakle brzina objekta od 0 km/h ne znači nužno da objekt miruje, nego samo da ima smjerom i iznosom jednaku brzinu kao kamera.

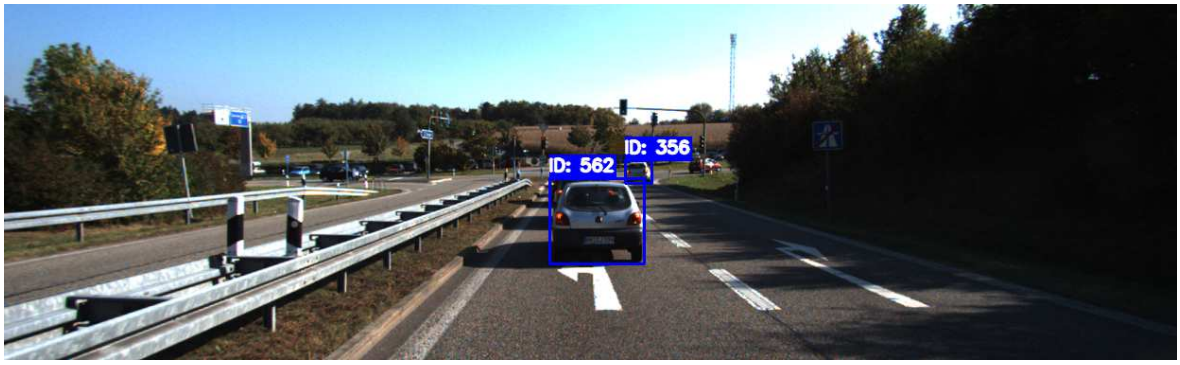
3.4.1. Scenarij kretanja vozila u istom smjeru s kamerom

Za početak, iz scenarija 20 odabran je objekt s ID-jem 356 za analizu kretanja. Razlog za to je što biva detektiran u duljini od preko 1 minute, tako da se njegovo kretanje može pratiti na većem broju slika. Na slici 3.10. prikazno je njegovo kretanje po lateralnoj osi kamere. Na samom početku vidljivo je odstupanje početnog uvjeta za poziciju ma x osi od stvare pozicije, no model brzo hvata trend podataka te ga dobro prati cijelom duljinom mjerenja. Brzina po x-osi ostaje vrlo blizu nuli prvih 400 slika. U tom periodu kamera prati vozilo ispred sebe bez ikakvog skretanja. Oko 600. slike objekt počinje skretati s autoceste na silaznu rampu. Tada objekt počinje pratiti cestu zavojem u desno, potom u lijevo, nakon čega se kamera zaustavlja na semaforu, a objekt skrene desno i nestane



(a) Slika 294 - vožnja autocestom

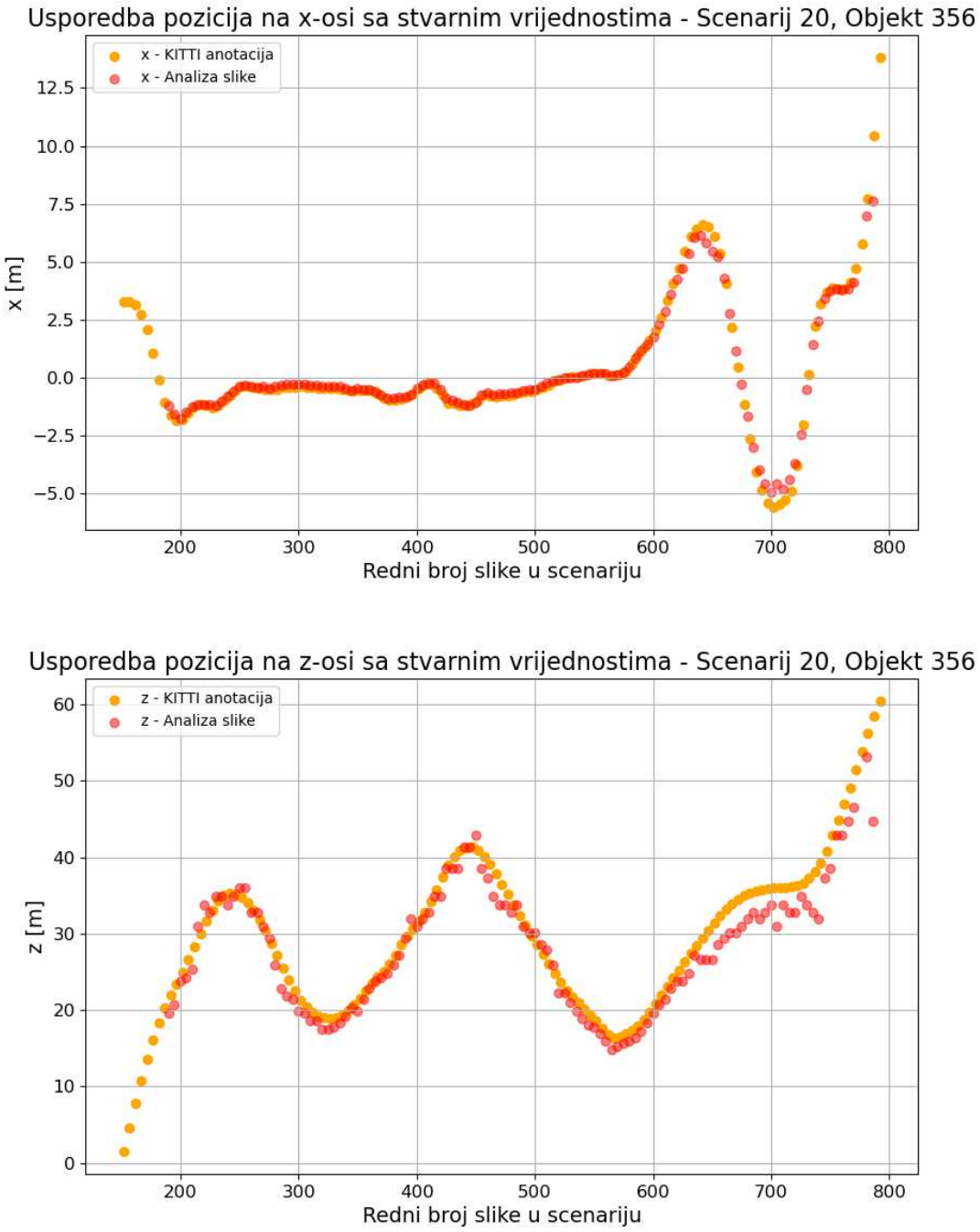
(b) Slika 617 - silazak s autoceste



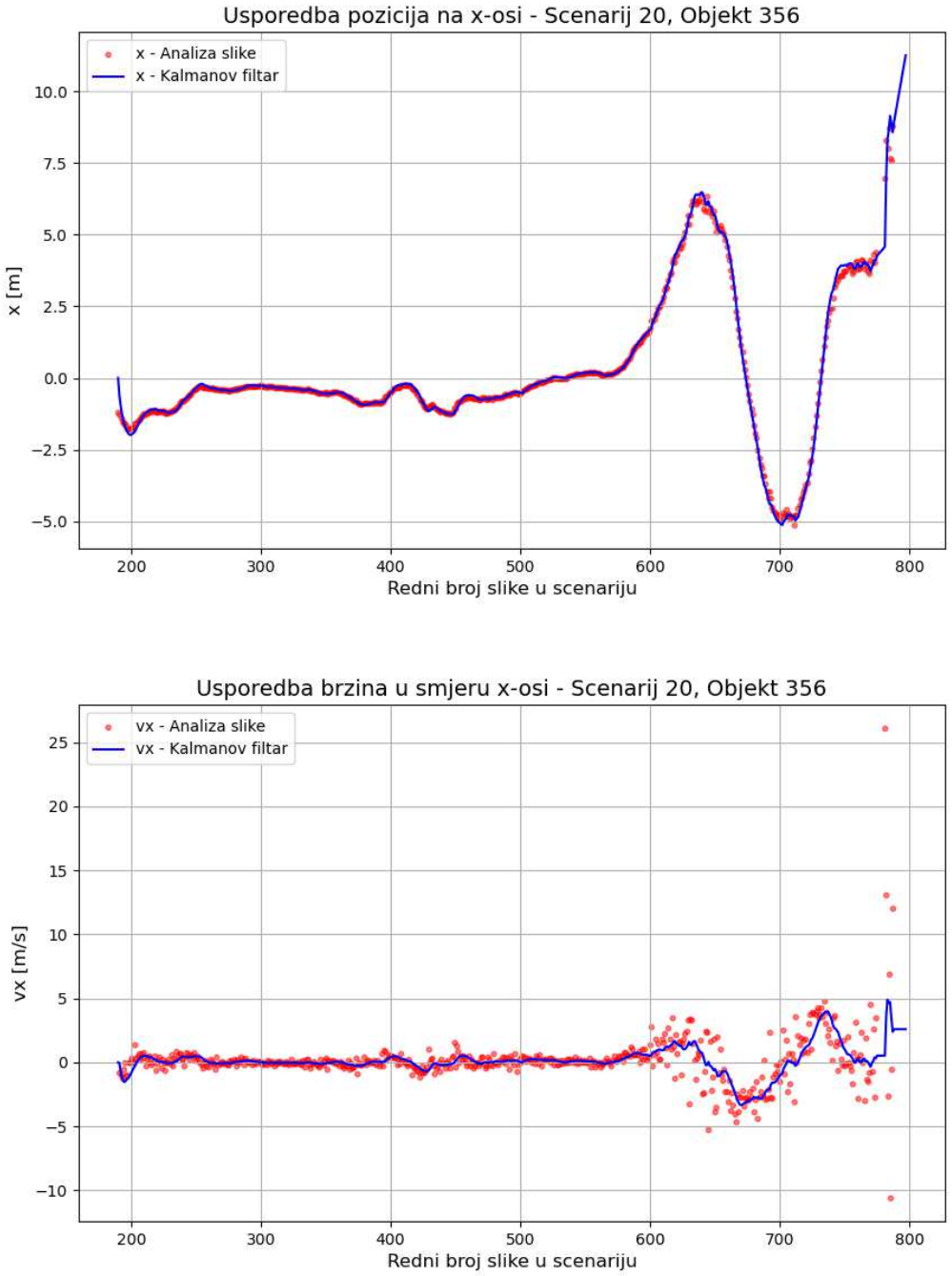
(c) Slika 771 - skretanje na semaforu

Slika 3.8. Scenarij 20, kretanje objekta 356

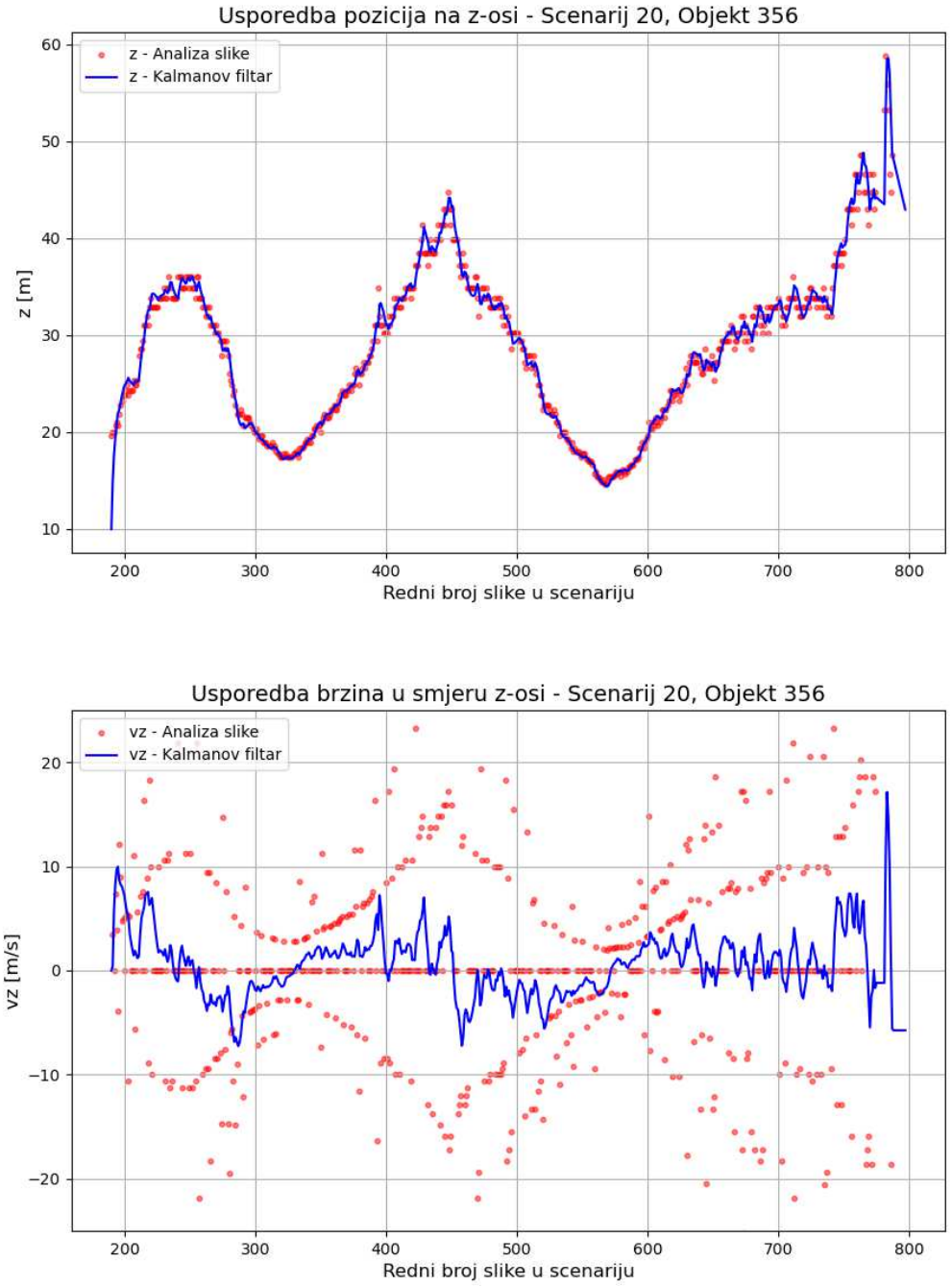
iz kadra. Na slici 3.11. može se pratiti kretanje objekta po longitudinalnoj osi. Podatci o brzini su vrlo zašumljeni, no Kalmanov filtar izglađuje šumove te daje interpretabilnije podatke. Trendovi brzine odgovaraju pomacima po z-osi. U trenutcima kada je brzina pozitivna, udaljenost između kamere i objekta se povećava i obrnuto. Iz KITTI anotacija izađeni su podatci o stvarnoj lokaciji objekta. Usporedba njih i estimacije filtra porikazana je na slici 3.12. Vidljivo je da sustav, uz određena odstupanja zbog šumova, dobro estimira udaljenost objekta, posebice na x-osi. Mjerenja longitudinalne komponente imaju znatno više šuma, što se ističe pri kraju mjerenja.



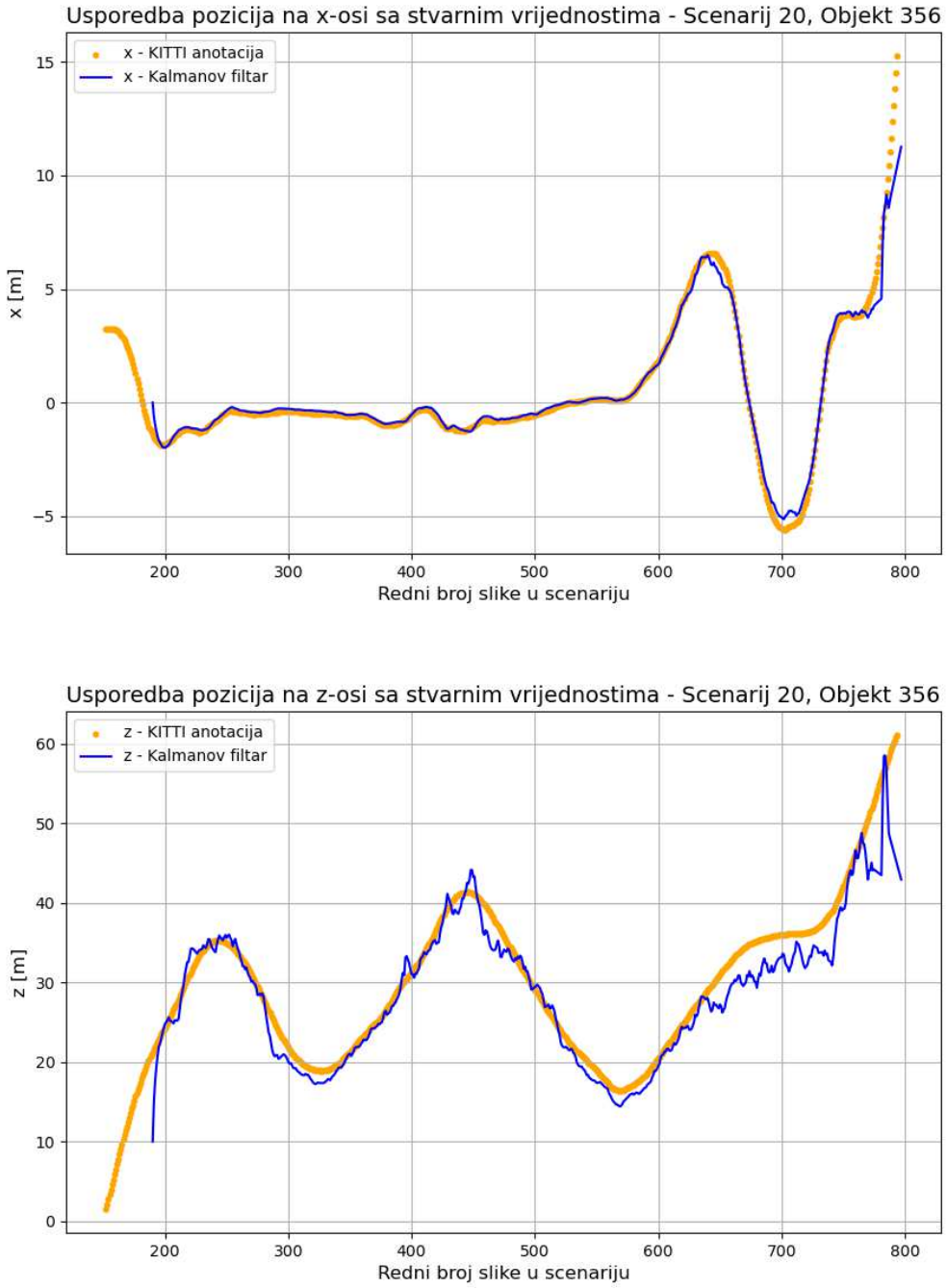
Slika 3.9. Scenarij 20, trajektorija objekta 356 (KITTI ID 12). Usporedba izračunatih koordinata s mjerenjima (lateralna i longitudinalna os)



Slika 3.10. Scenarij 20, trajektorija objekta 356 (KITTI ID 12). Usporedba izlaza Kalmanovog filtra s izračunatim vrijednostima (lateralna os).



Slika 3.11. Scenarij 20, trajektorija objekta 356 (KITTI ID 12). Usporedba izlaza Kalmanovog filtra s izračunatim vrijednostima (longitudinalna os)

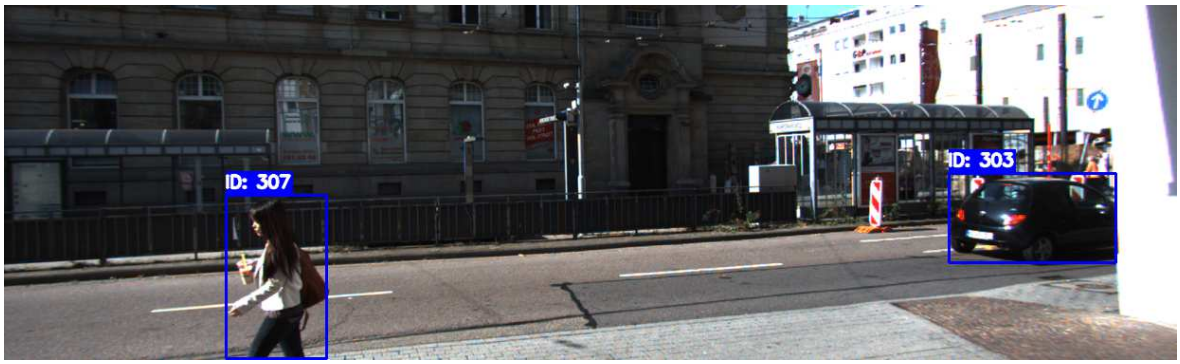


Slika 3.12. Scenarij 20, trajektorija objekta 356 (KITTI ID 12). Usporedba izlaza Kalmanovog filtra s mjerenjima (lateralna i longitudinalna os)

3.4.2. Scenarij prelaska pješaka preko ceste

Na slici 3.13. može se vidjeti kako pješak s ID-jem 307 prelazi cestu ispred kamere. Većina gibanja se odvija po lateralnoj osi, što potvrđuju i grafovi na slikama 3.15. i 3.16. Nakon što sustav uspije iz početnog uvjeta $x_0 = 0$ m uhvatiti poziciju objekta, prati precizno

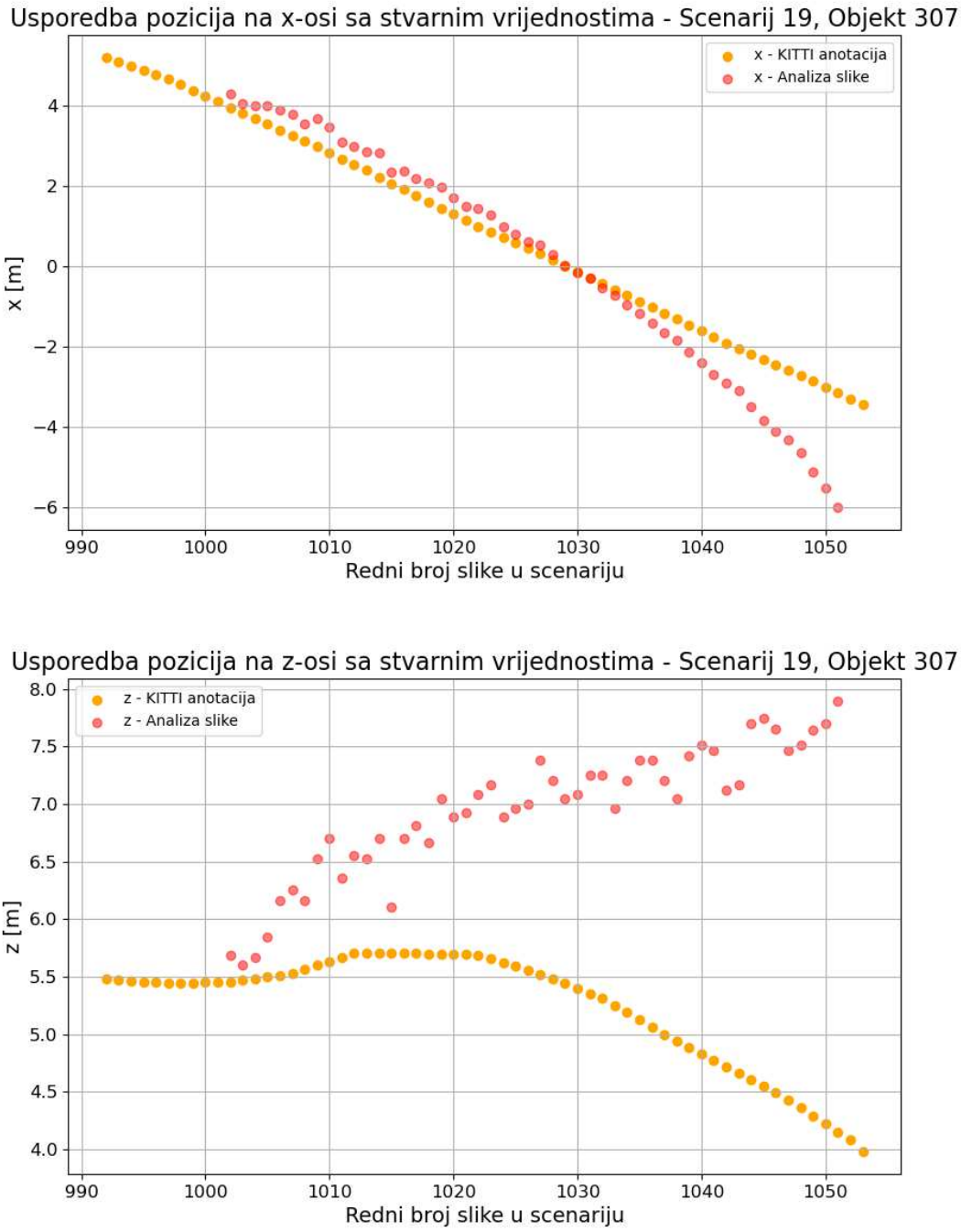
(a) Slika 1002 - prva detekcija pješaka



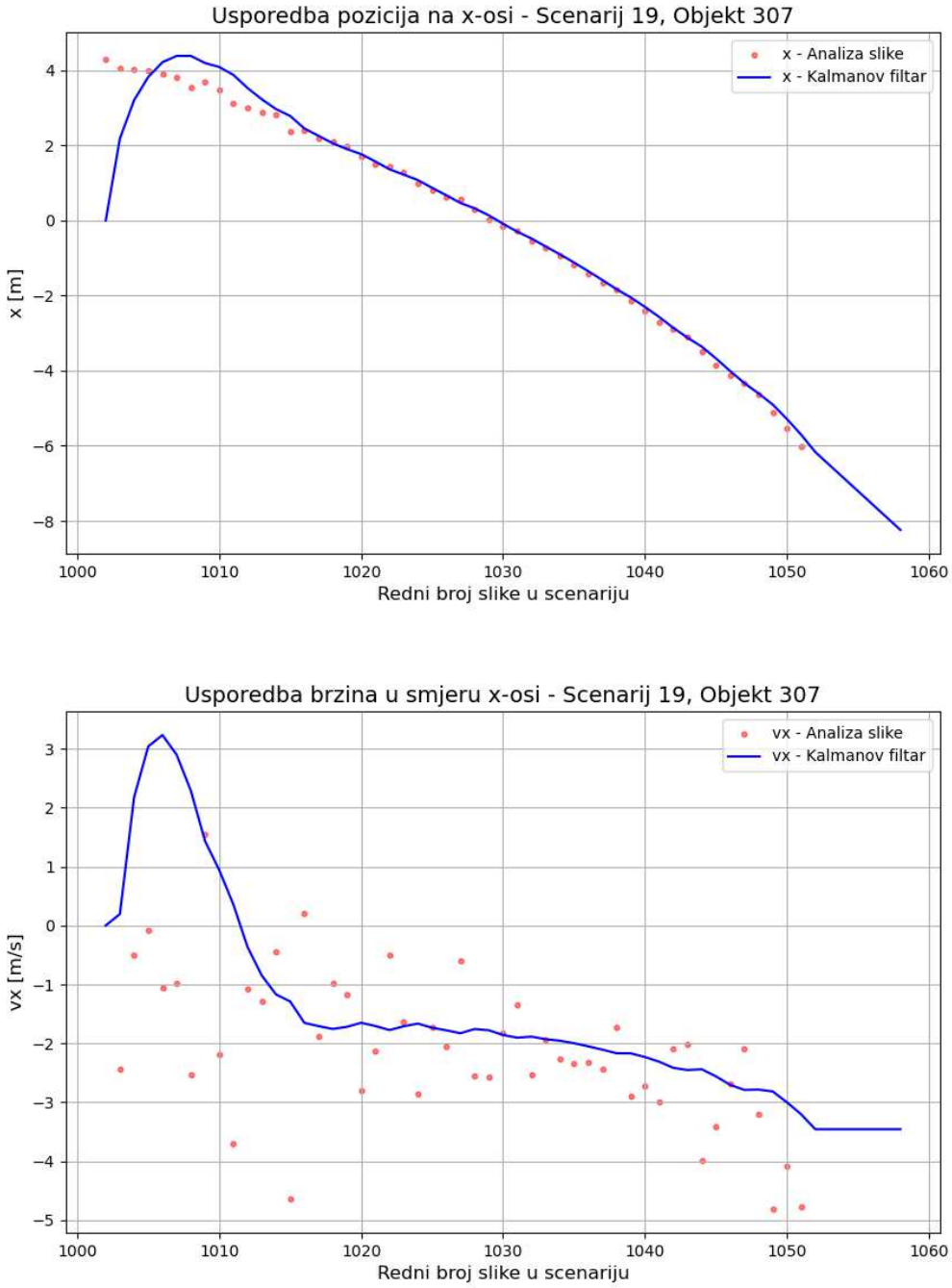
(b) Slika 1043 - pješak prelazi cestu

Slika 3.13. Scenarij 19, kretanje objekta 307

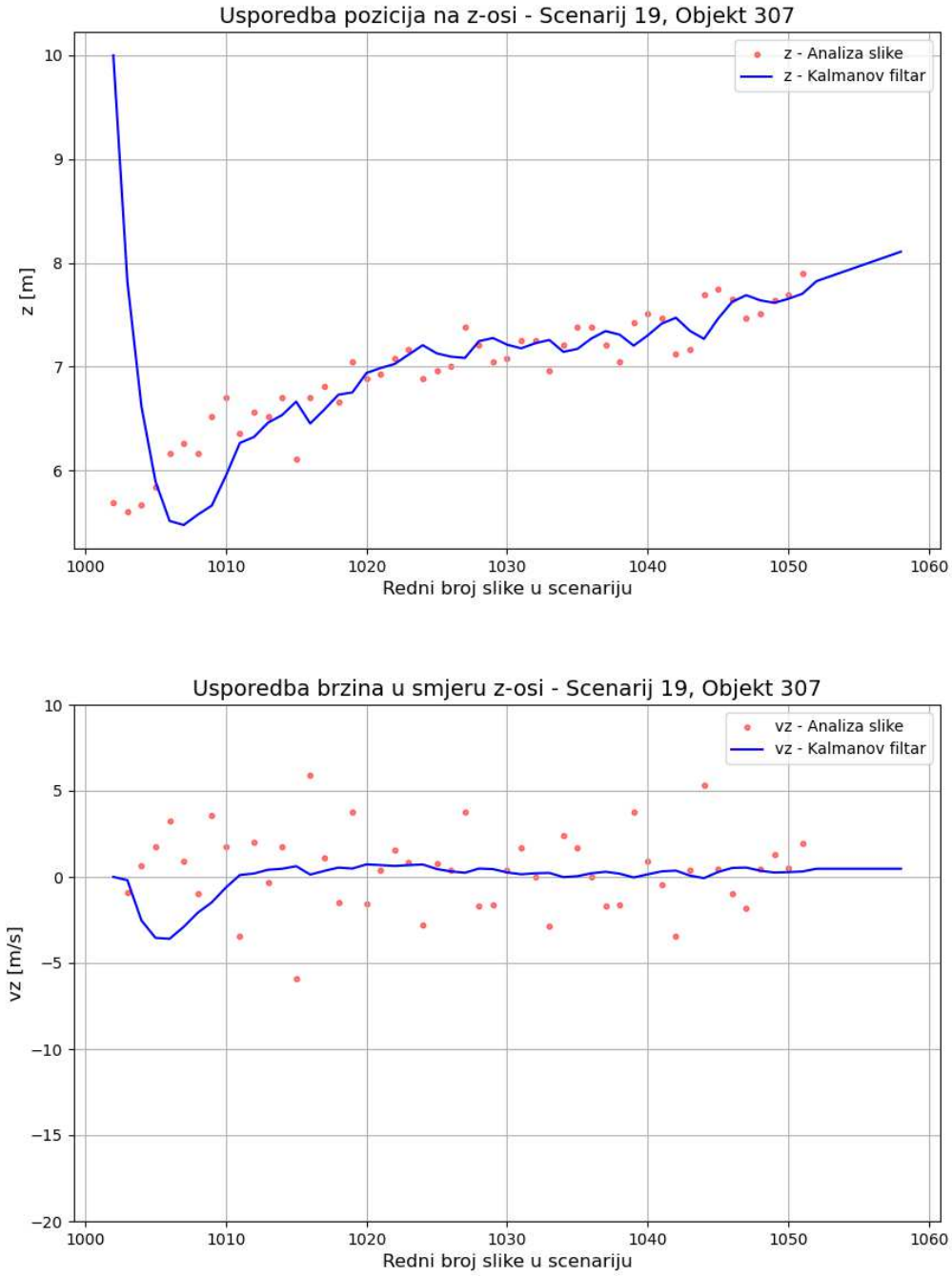
njegovo kretanje s desna na lijevo. U trenutku kada se pješak nađe na 6 m lijevo po x-osi od kamere, nestaje iz kadra te nestaju i sva mjerenja. Model po zadnjoj estimiranoj brzini predviđa njegovu poziciju na još 10 sljedećih slika. Zbog početnog uvjeta od $z_0 = 10$ m modelu treba neko vrijeme da dosegne mjerenja i po z-osi. Iz istog razloga brzina po z-osi ima veliki pad u početku gdje premašuje mjerenja. U početku je zbog velikog odmaka od početnog uvjeta, te model precjenjuje brzinu dok se ne uhvati mjerenje. Na slici 3.17. prikazana je usporedba estimacija sa stvarnim vrijednostima pozicije iz KITTI anotacija. Dok estimacije po x-osi bolje odgovaraju stvarnim podatcima, estimacija po z-osi ima potpuno suprotan trend od stvarnosti. Izvor tog trenda je formula 2.1, po kojoj se računa dubina na slici, odnosno koordinata z. Visina osobe i fokalna duljina kamere su konstantne vrijednosti, dakle jedina promjenjiva varijabla je visina graničnog okvira. Pregledom scene može se utvrditi da se pješak pri prelasku pomalo približava kameri čime ona više ne hvata u kadar donji dio tijela. Tako se smanjuje i visina graničnog okvira što uzrokuje rast udaljenosti po z-osi. Također, objekt je detektiran tek nakon što potpuno uđe u kadar, dok je u anotacijama prisutan i prije toga.



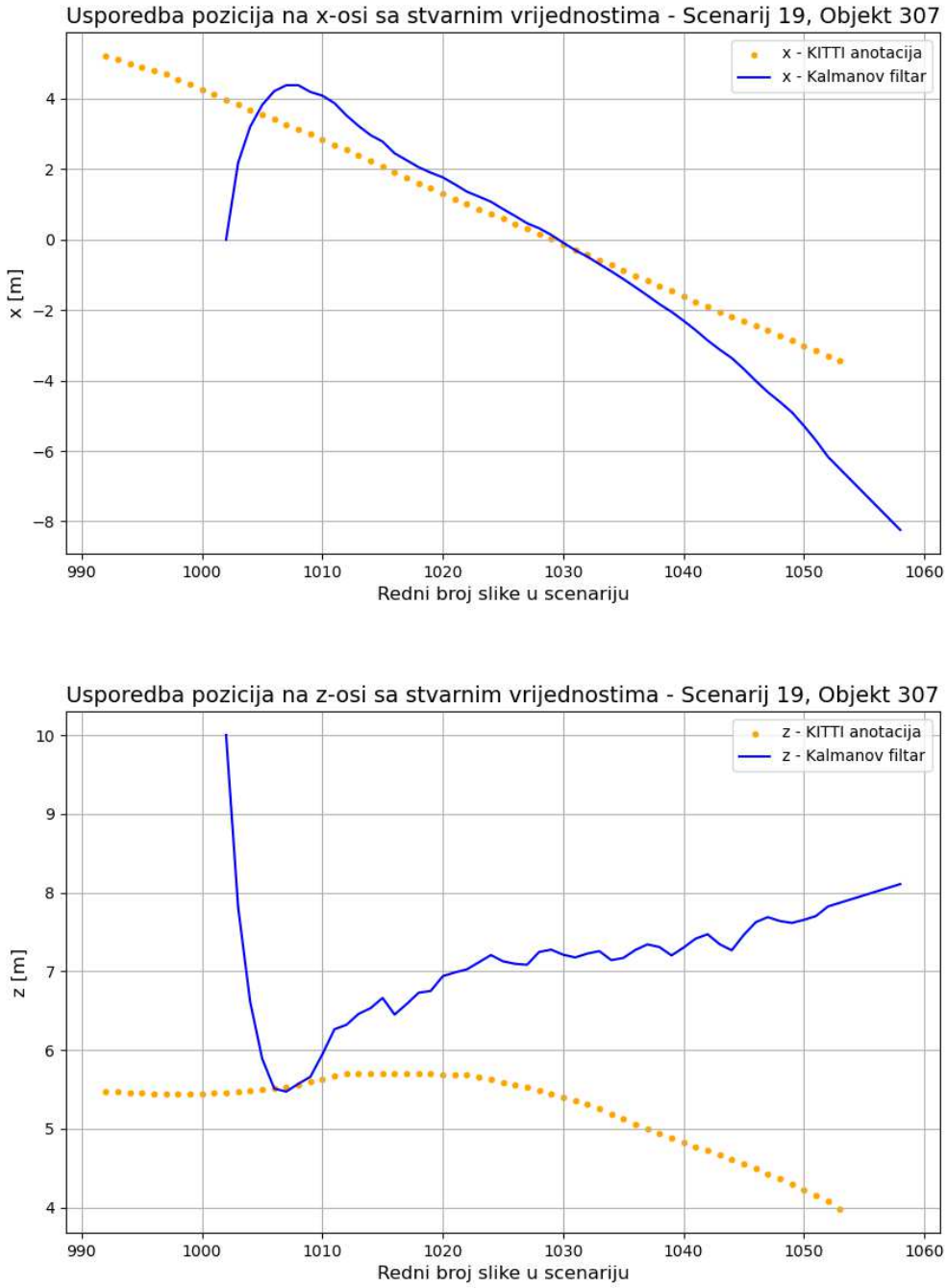
Slika 3.14. Scenarij 19, trajektorija objekta 307 (KITTI ID 83). Usporedba izračunatih koordinata s mjerenjima (lateralna i longitudinalna os)



Slika 3.15. Scenarij 19, trajektorija objekta 307 (KITTI ID 83). Usporedba izlaza Kalmanovog filtra s izračunatim vrijednostima (lateralna os)



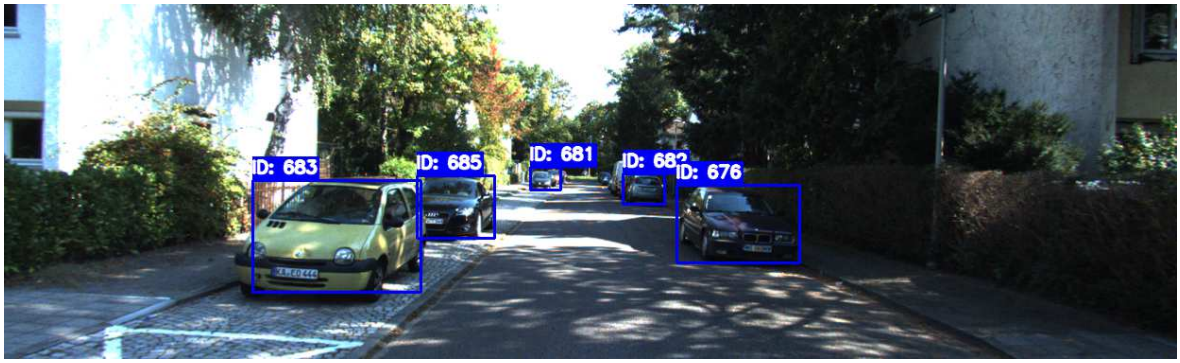
Slika 3.16. Scenarij 19, trajektorija objekta 307 (KITTI ID 83). Usporedba izlaza Kalmanovog filtra s izračunatim vrijednostima (longitudinalna os)



Slika 3.17. Scenarij 19, trajektorija objekta 307 (KITTI ID 83). Usporedba izlaza Kalmanovog filtra s mjerenjima (lateralna i longitudinalna os)

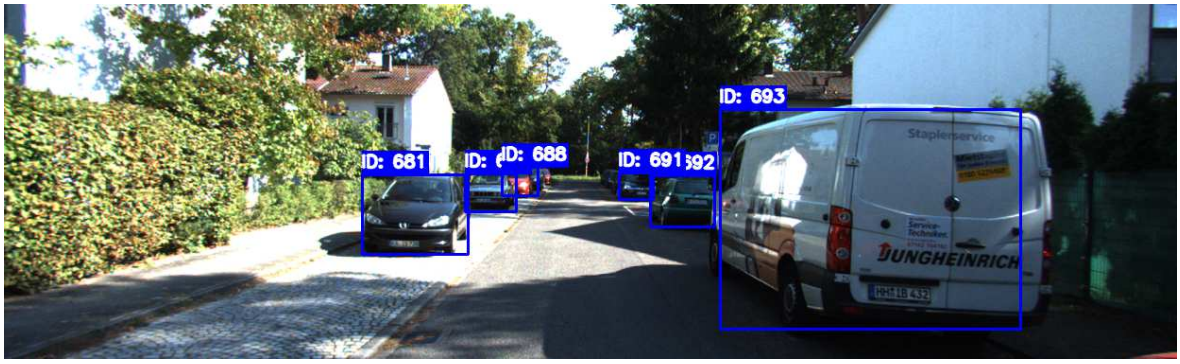
3.4.3. Scenarij prolaska pored parkiranog autmobila

U zadnjem primjeru prikazanom na slici 3.18., promotrit ćemo rad sustava za estimaciju kada dolazi do privremenog nestanka mjerenja. Estimacije pozicije trebale bi pomoći u trenutku kada objekt bude reidentificiran na način da se estimirana pozicija lakše uklopi u mjerenja, bez velikih skokova koje uzrokuju početni uvjeti. Na slikama 3.20. i 3.21. prikazan je rad Kalmanovog filtra u navedenom scenariju. Nakon što objekt nestane na slici 436, estimirane vrijednosti brzina, zbog prirode modela, ostaju iste sve dok se mjerenje ponovno ne pojavi. U skladu s tom estimacijom brzine nastavljaju se mijenjati pozicije. Kada se mjerenje ponovno pojavi na slici 441, dolazi do korekcije te se estmirane vrijednosti glatko uklope u mjerenja. Da toga nema, sustav bi ponovno prolazio kroz promjene kakve su vidljive na početcima grafova, gdje estimacija kreće iz početnih uvjeta. Na slici 3.22. uspoređene su estimacije sa stvarnim vrijednostima. Pozicija na x-osi je dobro procijenjena sve do 470. slike kada zbog greške u lokalizaciji ona krene naglo opadati. Na z-osi estimirana pozicija je nešto bliža stvarnosti i bolje prati trend pada, odnosno približavanje parkiranom autu. Na slici 480 kamera prolazi pored auta čime nestaju sva mjerenja. Sustav nastavlja estimirati njegovu poziciju još neko vrijeme dok se ne dosegne prag od 10 nestalih mjerenja. Tada se ispravno pretpostavlja da je objekt dugotrajno nestao iz kadra te sustav zaboravlja prethodno stanje tog objekta. Kada bi došlo do njegove ponovne detekcije, filtar bi ponovno morao proći kroz cijeli postupak dodjeljivanja početnog stanja i kovergencije do mjerenih vrijednosti.



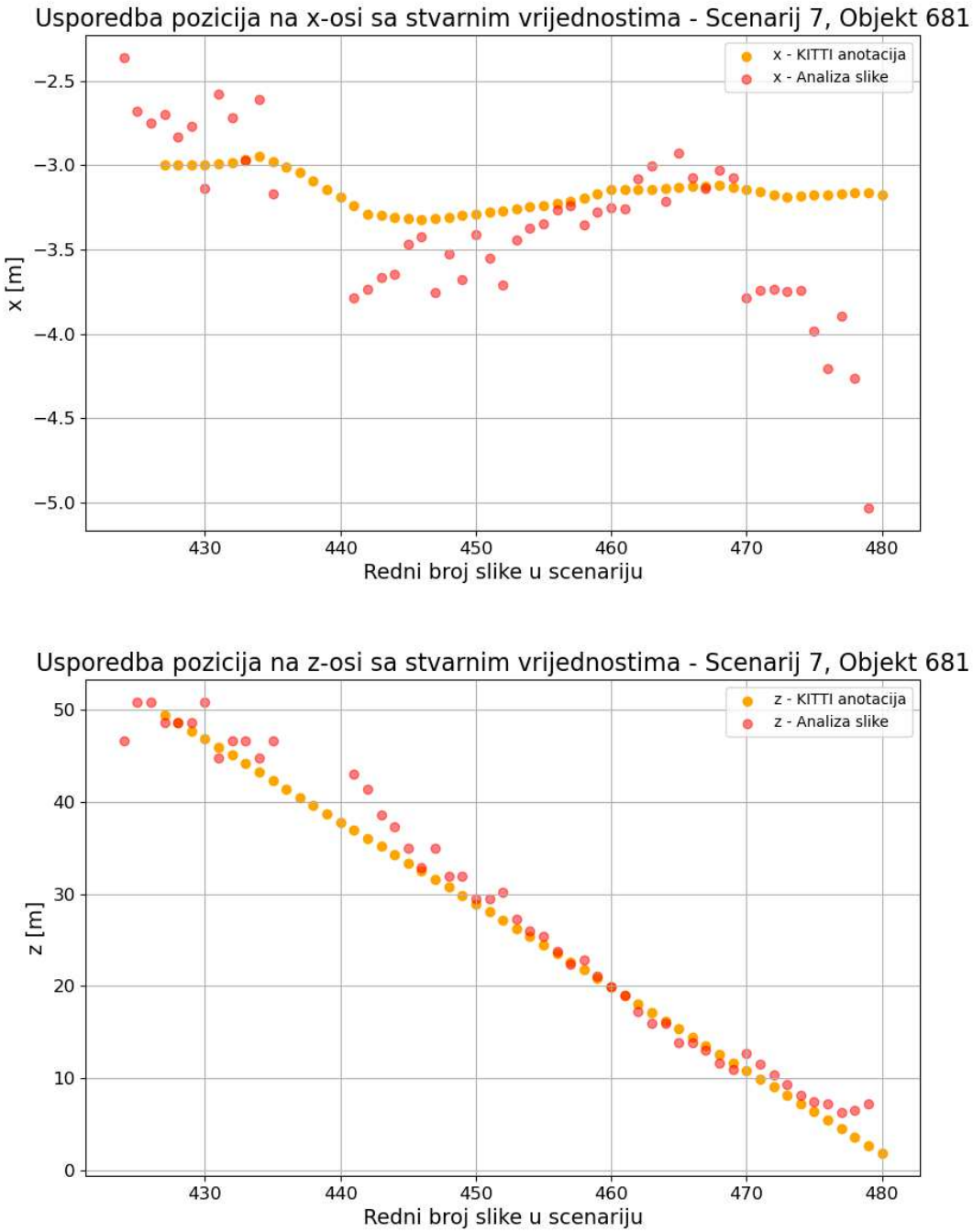
(a) Slika 424 - detektiran auto

(b) Slika 439 - nestanak detekcije/mjerenja

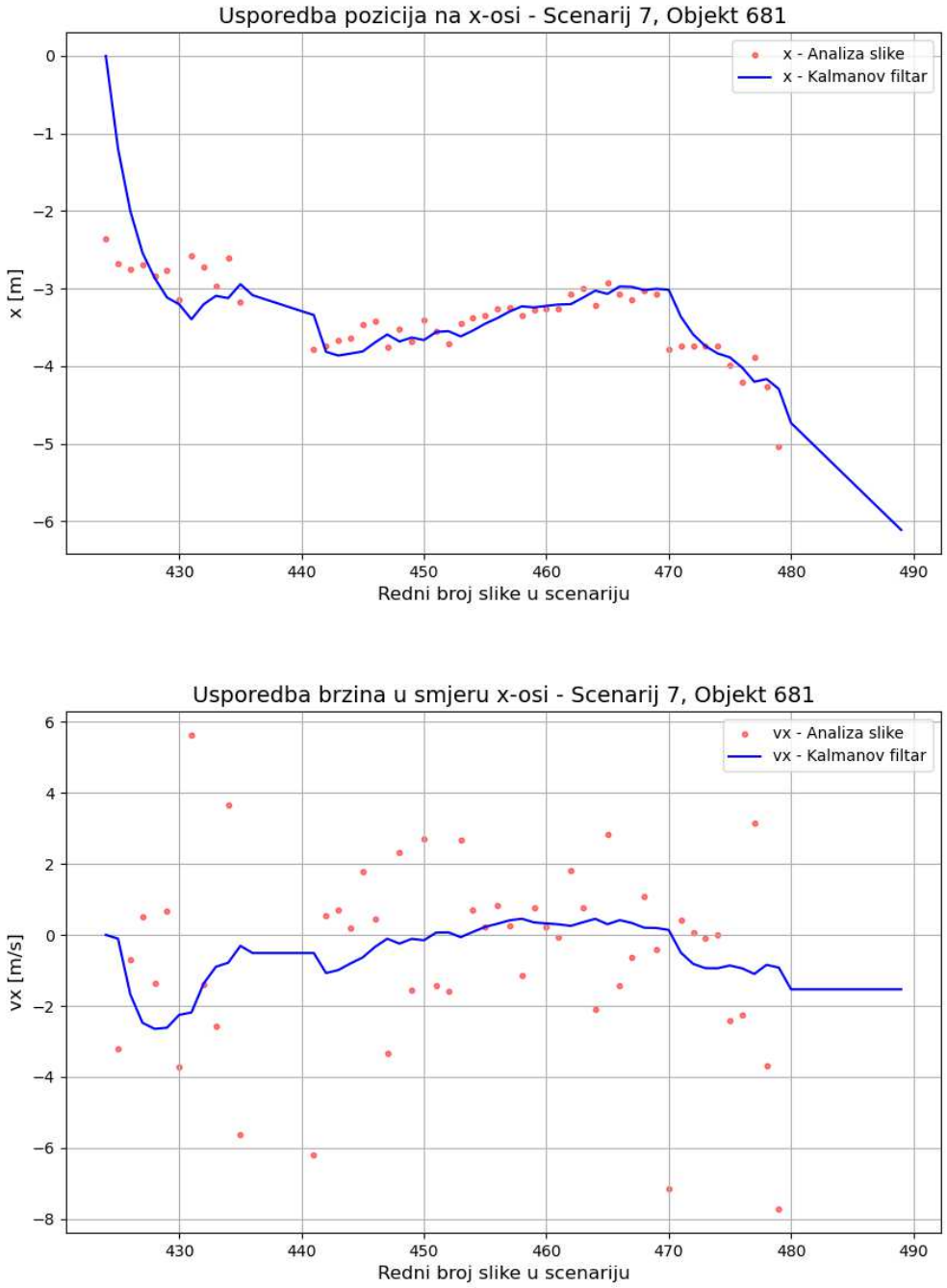


(c) Slika 467 - približavanje autu

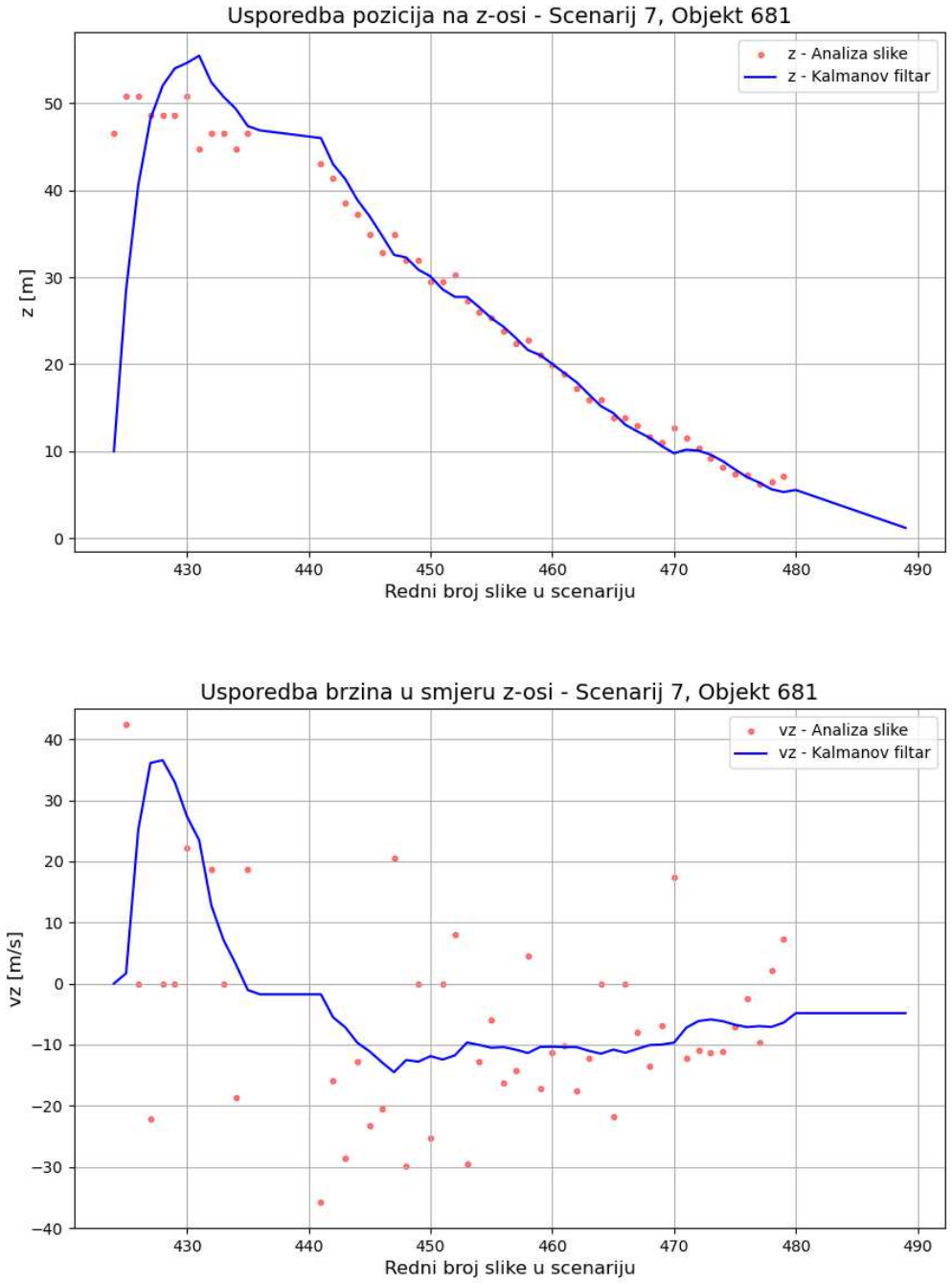
Slika 3.18. Scenarij 7, kretanje objekta 681



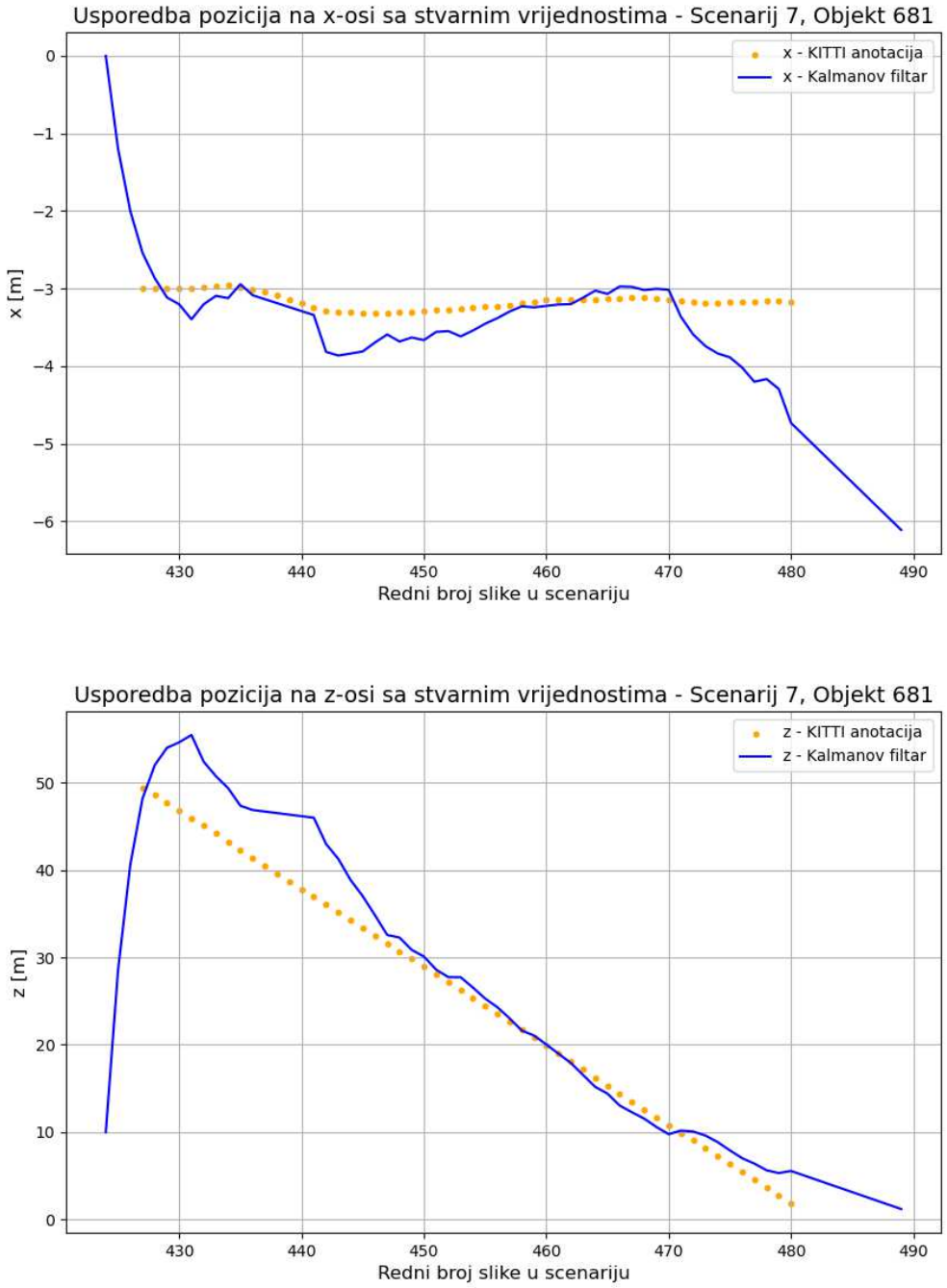
Slika 3.19. Scenarij 7, trajektorija objekta 681 (KITTI ID 44). Usporedba izračunatih koordinata s mjerenjima (lateralna i longitudinalna os)



Slika 3.20. Scenarij 7, trajektorija objekta 681 (KITTI ID 44). Usporedba izlaza Kalmanovog filtra s izračunatima vrijednostima (lateralna os)



Slika 3.21. Scenarij 7, trajektorija objekta 681 (KITTI ID 44). Usporedba izlaza Kalmanovog filtra s izračunatim vrijednostima (longitudinalna os)



Slika 3.22. Scenarij 7, trajektorija objekta 681 (KITTI ID 44). Usporedba izlaza Kalmanovog filtra s mjerenjima (lateralna i longitudinalna os)

3.4.4. Pregled rada Kalmanovog filtra

Iz navedenih primjera može se izvesti nekoliko zaključaka o radu modela. Model vrlo uspješno radi kao filtar čime smanjuje šumove koji u korištenim podatcima nisu zanemarivi. Time daje podatke koji su interpretabilni i nemaju toliko izražene skokove. Testirana je i mogućnost filtra da radi kao estimator, što je posebice bitno u trenutcima kada mjerenja nestanu zbog greški u drugim komponentama cijeloga sustava poput detekcije ili praćenja objekata. Estimacija u tim trenutcima radi dobro i omogućuje bolju procjenu kada se mjerenja ponovno vrate. Zbog nedostataka drugih mjerenja, poput akceleracije, model ne može predvidjeti kretanje brzine kada mjerenje izostane. Tada se oslanja samo na pretpostavku o konstantnoj brzini. Za bolju estimaciju brzine, a time i pozicije, bilo bi korisno koristiti i podatke o akceleraciji čime Kalmanov filtar postaje složeniji i dobiva ulaznu varijablu. Dodatno, razlika između zadanih i stvarnih početnih uvjeta uzrokuje značajne greške u estimaciji sve dok ne dođe do konvergencije. U budućem radu bi se mogle razmotriti naprednije tehnike inicijalizacije stanja poput inicijalizacije s više hipoteza, gdje se prati estimacija s nekoliko različitih početnih uvjeta, a na kraju se bira ona koja najbolje odgovara ranim mjerenjima [27]. Kao nedostatak se pokazalo i fiktivno udaljivanje objekta primijećeno na slici 3.17., gdje se zbog približavanja objekta kameri smanjuje visina graničnog okvira. Taj slučaj se u implementaciji ne razlikuje od onoga kada se okvir smanjuje zbog stvarnog udaljavanja objekta od kamere te dolazi do pogrešne projekcije koordinata u 3D prostor. Kao rješenje nudi se mogućnost postavljanja kamere na sam branik automobila, kako bi do rezanja objekata u kadru došlo na manjoj udaljenosti.

4. Zaključak

Cilj ovoga rada je bio razviti metodologiju za estimaciju pozicije i brzine objekata na nizu slika. Problem je razložen na nekoliko podkoraka: detekcija uz YOLOv5, lokalizacija geometrijskom projekcijom, praćenje objekata koristeći SORT algoritam te na kraju estimacija pozicije i brzine Kalmanovim filtrom. Predloženi pristup pokazao je više prednosti. YOLOv5 efektivno detektira objekte na slikama, uz male nedostatke kada je riječ o udaljenijim objektima. Geometrijskom projekcijom uz dodatno apriorno znanje o stvarnoj visini objekata sustav može mapirati detektirane objekte u koordinate u stvarnom svijetu koristeći samo monokularne slike. Iako je distribucija greške lokalizacije povoljna, za veliku većinu objekata ona odstupa od traženih normi od 0.15 m za lateralnu i longitudinalnu te 0.48 m za vertikalnu komponentu. Oslanjanje na apriorno znanje limitira mogućnost sustava da točno lokalizira objekte u scenarijima gdje pretpostavke odstupaju od stvarnosti. Iz tog razloga, samo monokularne slike nisu najbolji izvor podataka za estimaciju pozicije. Za precizniju estimaciju preporučljivo je uključiti druge senzore poput LiDAR-a ili stereo para kamera. Pri praćenju objekata pomoću alrgoritma SORT, veliku prepreku je radila zaklonjenost objekata te nagle promjene pozicije. Unatoč tome, algoritam većinski dobro identificira objekte, a potencijalni propusti u praćenju mogu biti kompenzirani korištenjem estimatora poput Kalmanovog filtra. Usprkos ograničenjima, sustav opisan u ovom radu pokazuje potencijal korištenja dubokog učenja, geometrijske projekcije i filtiranja u estimaciji pozicije i brzine objekata, ali ostavlja i prostora za dodatan razvoj, posebice u lokalizaciji objekata u trodimenzionalnom prostoru.

Literatura

[1] Hwee Yng Yeo. *How Automotive Radars Are Advancing Safety Features*. Pristupljeno 30.12.2024. URL: https://www.keysight.com/blogs/en/tech/educ/2023/automotive-radar.

[2] SWARCO. *LiDAR for cars: autonomous driving and the technology of the future*. Pristupljeno 30.12.2024. URL: https://www.swarco.com/mobility-future/intelligent-transportation-systems/lidar-cars.

[3] Pranav Durai. *Stereo Vision in ADAS: Pioneering Depth Perception Beyond LiDA*. Pristupljeno 30.12.2024. URL: https://learnopencv.com/adas-stereo-vision/.

[4] David J. Fleet. „Optical Flow Estimation". *Mathematical Model in Computer Vision: The Handbook*. Springer, 2005. URL: https://www.cs.toronto.edu/~fleet/research/Papers/flowChapter05.pdf.

[5] Ultralytics. *YOLOv5: Pretrained and Custom Object Detection Models*. 2020. URL: https://github.com/ultralytics/yolov5.

[6] Andreas Geiger. „Vision Meets Robotics: The KITTI Dataset". *International Journal of Robotics Research (IJRR)* 32.11 (2013.), str. 1231–1237. URL: https://www.cvlibs.net/publications/Geiger2013IJRR.pdf.

[7] Karlsruhe Institute of Technology. *The KITTI Vision Benchmark Suite*. Pristupljeno 2024. URL: https://www.cvlibs.net/datasets/kitti/setup.php.

[8] Karlsruhe Institute of Technology. *The KITTI Vision Benchmark Suite*. Pristupljeno 2024. URL: https://www.cvlibs.net/datasets/kitti/eval_tracking.php.

[9] Andreas Geiger, Philip Lenz i Raquel Urtasun. „Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012.). URL: https://www.cvlibs.net/publications/Geiger2012CVPR.pdf.

[10] Encyclopaedia Britannica. *Optical Axis*. Pristupljeno: 1.1.2025. n.d. URL: https://www.britannica.com/technology/optical-axis.

[11] IBM Data i AI Team. *Faster R-CNN vs YOLO vs SSD: Object Detection Algorithms*. Pristupljeno: 1.1.2025. 2020. URL: https://medium.com/ibm-data-ai/faster-r-cnn-vs-yolo-vs-ssd-object-detection-algorithms-18badb0e02dc.

[12] Joseph Redmon i dr. „You Only Look Once: Unified, Real-Time Object Detection". *arXiv preprint arXiv:1506.02640* (2016.). URL: https://arxiv.org/pdf/1506.02640.

[13] COCO Consortium. *COCO: Common Objects in Context*. Pristupljeno 3.1.2025. 2014. URL: https://cocodataset.org/#home.

[14] Stanford University. *Lecture Notes on Stereo Vision Systems*. 2021. URL: https://web.stanford.edu/class/cs231a/course_notes/04-stereo-systems.pdf.

[15] Peng Lu i dr. „Geometry Uncertainty Projection Network for Monocular 3D Object Detection". *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021., str. 3111–3120. URL: https://openaccess.thecvf.com/content/ICCV2021/papers/Lu_Geometry_Uncertainty_Projection_Network_for_Monocular_3D_Object_Detection_ICCV_2021_paper.pdf.

[16] Basler Zeitung. *Autos werden immer breiter und länger*. Pristupljeno 9.1.2025. 2018. URL: https://www.bazonline.ch/autos-werden-immer-breiter-und-laenger-288912673833.

[17] World Data. Pristupljeno 9.1.2025. URL: https://www.worlddata.info/average-bodyheight.php.

[18] Mercedes-Benz. Pristupljeno 9.1.2025. URL: https://www.mercedes-benz-bus.com/en_DE/models/citaro/facts.html.

[19] Depth Roro. Pristupljeno 9.1.2025. URL: https://depthroro.com/all-vehicle-dimensions/mercedes-benz-actros-2544l-6x2-rigid-fully-built-up-and-self-propelled-dimensions/.

[20] Nela Lepur. *Geometrijske Transformacije*. Prezentacija FER-ovog kolegija Računalna grafika. URL: https://web.math.pmf.unizg.hr/~nela/rgpredavanja/geometrijske_transformacije.

[21] Clement Hardy. *The Hungarian Algorithm: Assign Detections to Trackers Like a Pro*. Pristupljeno 12.1.2025. URL: https://www.thinkautonomous.ai/blog/hungarian-algorithm/.

[22] V7 Labs. *Intersection over Union (IoU): A Complete Guide*. Pristupljeno 10.1.2025. 2022. URL: https://www.v7labs.com/blog/intersection-over-union-guide.

[23] Abraham Bewley i dr. „SORT: Simple Online and Realtime Tracking". *arXiv:1602.00763* (2016.). URL: https://arxiv.org/pdf/1602.00763.

[24] Alex H. Lang i dr. „Point Cloud Processing Techniques for Autonomous Driving: Towards Better Generalization and Robustness". *arXiv:1906.01061* (2019.). URL: https://arxiv.org/pdf/1906.01061.

[25] Abraham Bewley i dr. *SORT: Simple Online and Realtime Tracking*. 2016. URL: https://github.com/abewley/sort.

[26] Ikomia.ai. *Deep SORT Object Tracking Guide*. Pristupljeno 17.1.2025. URL: https://www.ikomia.ai/blog/deep-sort-object-tracking-guide.

[27] Gustaf Hendeby. *Lecture Notes on Kalman Filtering*. 2023. URL: https://mtt.edu.hendeby.se/file/le5.pdf.

Sažetak

Cilj ovog rada je razviti sustav za estimaciju pozicije i brzine objekata na nizu monokularnih slika. Objekti su detektirani modelom YOLOv5, nakon čega je slijedila lokalizacija objekata u trodimanzionalnom prostoru. Pomoću geometrijske projekcije i apriornog znanja o visini objekata, određena je njihova pozicija u stvarnom svijetu. Radi praćenja objekata, dodijeljeni su im jedinstveni ID-jevi korištenjem algoritma SORT. Primijenjen je Kalmanov filtar za uklanjanje šuma u podatcima te estimaciju pri nedostatku mjerenja. Predloženi pristup, uz vrlo ograničenu količinu resursa, omogućava praćenje objekata u stvarnom svijetu.

Ključne riječi

Računalni vid, monokularne slike, detekcija objekata, YOLO, geometrijska projekcija, SORT, estimacija pozicije i brzine, Kalmanov filtar

Summary

The purpose of this thesis was to develop a system for estimating the position and speed of objects in a series of monocular images. Objects were detected using the YOLOv5 model, followed by localization in three-dimensional space. Geometric projection and prior knowledge of object height were used to determine their real-world positions. To track objects, unique IDs were assigned using the SORT algorithm. A Kalman filter was applied to reduce noise in the data and estimate positions in the absence of measurements. The proposed approach enables real-world object tracking with very limited resources.

Key words

Computer vision, monocular images, object detection, YOLO, geometric projection, SORT, position and velocity estimation, Kalman filter