

# Fotometrijsko izjednačavanje videozapisa iz različitih kamera

---

Levak, Tomislav

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:500664>

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

Download date / Datum preuzimanja: **2025-01-02**



Repository / Repozitorij:

[University North Digital Repository](#)



**SVEUČILIŠTE SJEVER**  
**SVEUČILIŠNI CENTAR VARAŽDIN**



DIPLOMSKI RAD br. 007/MMD/2020

**Fotometrijsko izjednačavanje videozapisa iz  
različitih kamera**

Tomislav Levak

Varaždin, srpanj 2020.

**SVEUČILIŠTE SJEVER**  
**SVEUČILIŠNI CENTAR VARAŽDIN**  
**Studij Multimedija**



DIPLOMSKI RAD br. 007/MMD/2020

**Fotometrijsko izjednačavanje videozapisa iz  
različitih kamera**

Student:

Tomislav Levak, 0778/336D

Mentor:

doc. dr. sc. Emil Dumić

Varaždin, srpanj 2020.

# Prijava diplomskog rada

## Definiranje teme diplomskog rada i povjerenstva

ODJEL	Odjel za multimediju		
STUDIJ	diplomski sveučilišni studij Multimedija		
PRIJAVNIK	Tomislav Levak	MATUČNI BROJ	0778/336D
DATUM	16.06.2020.	KOLEGIJ	Računalni vid
NASLOV RADA	Fotometrijsko izjednačavanje videozapisa iz različitih kamera		
NASLOV RADA NA ENGL. JEZIKU	Photometric alignment using surround view cameras		
MENTOR	Emil Dumić	ZVANJE	doc.dr.sc.
ČLANOVI POVJERENSTVA	1. doc.art.dr.sc. Mario Periša - predsjednik		
	2. doc.art. Robert Geček - član		
	3. doc.dr.sc. Emil Dumić - mentor		
	4. doc.dr.sc. Andrija Bernik - zamjenski član		
	5.		

## Zadatak diplomskog rada

RADI	007/MMD/2020
OPIS	<p>U ovom radu će se napraviti algoritam za fotometrijsko izjednačavanje videozapisa dobivenih iz više kamera. Napredni sustavi za pomoć vozaču (ADAS) je skup pametnih sustava koji pomažu vozaču upravljati vozilo, poput davanja upozorenja, pomaganja u donošenju odluka i poduzimanja automatskih odluka za zaštitu svih sudionika u prometu. Jedan od tih sustava pomaže vozačima da se sigurnije parkiraju tako da prikazuje okruženje vozila. Za dobivanje virtualnog pogleda odzgo nekog objekta često se koristi više širokokutnih kamera koje zajedno mogu pokriti cijelu panoramu. Iako kamere prikazuju različite scene, mogu se iskoristiti za dobivanje virtualnog pogleda odzgo. Osim problema s dobivanjem samog virtualnog pogleda (npr. iz širokokutnih leća se trebaju izračunati normalne, pravocrtne slike), prilikom kalibracije više kamera je bitno i njihovo fotometrijsko izjednačavanje. U slučaju kada algoritam mora raditi u realnom vremenu, bitno je i da algoritam brzo izračuna koeficijente kalibracije.</p> <p>Praktični dio problema je da se iz 4 različite kamere dobije jedan videozapis sa što bolje riješenim fotometrijskim izjednačavanjem, uz što kraće vrijeme izvođenja. Ispitat će se i različiti formati izlaza iz kamera (RGB i YCbCr), te na koji način direktno izračunati kalibracijske koeficijente, za fotometrijsko izjednačavanje, za svaku kameru. Kalibracijski koeficijenti se iz praktičnijih područja mogu izračunati koristeći neke od mjera pogreške, poput MSE. Budući ovaj problem mora imati i ograničenje s donje strane, može se upotrijebiti neka od metoda za rješavanje minimizacije funkcije, uz poznate funkcije ograničavanja, poput KKT metode (Karush-Kuhn-Tucker).</p>

ZADATAK URUČEN

18. 6. 2020



POTPIS MENTORA

Emil Dumić

## **Predgovor**

Sustav okruženja kamera za automobile je tek u nastajanju, poznatiji kao ADAS (Advanced Driver Assistance System). To je tehnologija koja pomaže vozačima da se parkiraju sigurnije dopuštajući im da vide kompletnu okolinu od vlastitog vozila za 360 stupnjeva.

ADAS sustav se najčešće sastoji od četiri do šest širokokutnih kamera, postavljenih oko vozila te svaka kamera je okrenuta u jednom smjeru. Sa postavljenih kamera složen je pogled iz ptičje perspektive vozila, sintetiziran i prikazan vozaču u stvarnom vremenu prilikom parkiranja.

Problemi koji se pojavljuju prilikom implementacije sustava su ti da prikaz na vozačevom ekranu ima vidljive granice između susjednih kamera zbog različitih uvjeta u kojima se kamere nalaze, kao i različite automatske ekspozicije (AE) i automatskog balansa bijele boje (AWB) koji ima svaka kamera. Na primjer, prilikom ulaska u podzemnu garažu, prednji dio automobila će biti u mraku s obzirom da se ulazi u mračni prostor, dok će stražnji kraj biti još uvijek na osvijetljenom prostoru.

Ključne riječi – fotometrijsko poravnanje, podudaranje kamera s više boja, spajanje slika, sustav kamera okruženja, ADAS

## Sažetak

U ovom radu će se napraviti algoritam za fotometrijsko izjednačavanje videozapisa dobivenih iz više kamera.

Napredni sustavi za pomoć vozaču (ADAS) je skup pametnih sustava koji pomažu vozaču upravljati vozilo, poput davanja upozorenja, pomaganja u donošenju odluka i poduzimanja automatskih odluka za zaštitu svih sudionika u prometu. Jedan od tih sustava pomaže vozačima da se sigurnije parkiraju tako da prikazuje okruženje vozila. Za dobivanje virtualnog pogleda odozgo nekog objekta često se koristi više širokokutnih kamera koje zajedno mogu pokriti cijelu panoramu. Iako kamere prikazuju različite scene, mogu se iskoristiti za dobivanje virtualnog pogleda odozgo. Osim problema s dobivanjem samog virtualnog pogleda (npr. iz širokokutnih leća se trebaju izračunati normalne, pravocrtne slike), prilikom kalibracije više kamera je bitno i njihovo fotometrijsko izjednačavanje. U slučaju kada algoritam mora raditi u realnom vremenu, bitno je i da algoritam brzo izračuna koeficijente kalibracije.

Praktični dio problema je da se iz 4 različite kamere dobije jedan videozapis sa što bolje riješenim fotometrijskim izjednačavanjem, uz što kraće vrijeme izvođenja. Ispitat će se i različiti formati izlaza iz kamera (RGB i YCbCr), te na koji način direktno izračunati kalibracijske koeficijente, za fotometrijsko izjednačavanje, za svaku kameru. Kalibracijski koeficijenti se iz preklapajućih područja mogu izračunati koristeći neke od mjera pogreške, poput MSE. Budući ovaj problem mora imati i ograničenje s donje strane, može se upotrijebiti neka od metoda za rješavanje minimizacije funkcije, uz poznate funkcije ograničavanja, poput KKT metode (Karush–Kuhn–Tucker).

## Popis korištenih kratica

- ADAS (Advanced Driver Assistance System)** – napredni sustavi za pomoć vozaču
- AE (Automatic Exposure)** – automatska ekspozicija
- AEB (Auto Emergency Braking)**– autonomni izvanredni sustav kočenja
- AWB (Auto White Balance)** – automatski balans bijele boje
- MSE (Mean Squared Error)** – mjeri prosjek kvadrata pogrešaka
- ABS (Automatic Brake System)** – sustav za kočenje
- ACC (Adaptive Cruise Control)** – sustav adaptivnog prilagođavanja brzine
- DAS (Digital Assistance)** – sustav za pomoć vozaču
- ASV (Autonomous Surface Vehicle)** – vozila s aktivnom sigurnošću
- LKS (Lane Keeping Support)** – sustav vozačeve trake, upozorenje o vozačevoj traci
- EDF (Earliest Deadline First)** – funkcija raspodjele ruba
- ROI (Region of Interests)** – područje interesa
- IPM (Intersection Priority Management)** – prioritet upravljanja raskrižjem
- MRF (Markov Random Field)** – Markov nasumična polja
- KF (Kalman Filtration)** – Kalman filtriranje
- LDW (Lane Departure Warning)** – upozorenje o odlasku sa prometne trake
- CICAS (Cooperative Intersection Collision Avoidance Systems)** – sustavi za izbjegavanje sudara u kolapsu
- BSD (Blind Spot Detection)** – sustav detekcije mrtve toče
- RCTA (Rear Crossing Traffic Alert)** – obavijest o stražnjem križnom prometu
- EBA (Emergency Brake Assistance)** – pomoć pri kočenju u nuždi
- IHC (Intelligent Headlight Control)** – inteligentna kontrola prednjih svjetala
- SLR (Single – Lens Reflex Camera)** – senzor kratkog dometa Lidar
- LCS (Lane Control Signs)** – pomoćni senzor pri promjeni prometne trake
- TSA (Transportation Security Assistance)** – sustav za prepoznavanje znakova u prometu
- RACam (Radar and Camera System)** - integrirani radar i senzor za fotoapararat
- KKT (Karush – Kuhn – Tucker)** – metoda za rješavanje minimalizacije funkcije uz otprije poznatu funkciju ograničavanja
- GB (Gigabyte)** – višestruki bajt jedinice za digitalne informacije

## Abstract

In this graduate thesis, an algorithm for photometric equalization of videos obtained by multiple cameras will be developed.

Advanced Driver Assistance Systems (ADAS) is set of smart systems that help the driver steer the vehicle, such as giving alerts, assisting in decision-making, and making automatic decisions to protect all road drivers and users. One of these systems helps drivers park more safely by displaying the vehicle's environment. To get a virtual view from above of an object, multiple wide-angle cameras are often used, which together can cover entire panorama view.

In addition to the problem of obtaining the virtual view itself (for example, normal, straight-line images should be calculated from wide-angle lenses), when calibrating multiple cameras, their photometric equalization is also important. In case the algorithm has to work in real time, it is also important that the algorithm quickly calculates the calibration coefficients.

The practical part of the problem is to get one video from 4 different cameras with the best possible photometric equalization, with the shortest possible execution time. Different output formats from cameras (RGB and YCbCr) will also be examined, and how to directly calculate the calibration coefficients for photometric equalization for each camera.

Calibration coefficients can be calculated from overlapping areas using some of the error measures, such as MSE. Since this problem must also have a limitation from below, some of the methods can be used to solve the minimization of the function, in addition to the known constraint functions, such as the KKT method (Karush – Kuhn – Tucker).



## List of abbreviations used

- ADAS** - Advanced Driver Assistance System
- AE** - Automatic Exposure
- AEB** - Auto Emergency Braking
- AWB** - Auto White Balance
- MSE** - Mean Squared Error
- ABS** - Automatic Brake System
- ACC** - Adaptive Cruise Control
- DAS** - Digital Assistance
- ASV** - Autonomous Surface Vehicle
- LKS** - Lane Keeping Support
- EDF** - Earliest Deadline First
- ROI** - Region of Interests
- IPM** - Intersection Priority Management
- MRF** - Markov Random Field
- KF** - Kalman Filtration
- LDW** - Lane Departure Warning
- CICAS** - Cooperative Intersection Collision Avoidance Systems
- BSD** - Blind Spot Detection
- RCTA** - Rear Crossing Traffic Alert
- EBA** - Emergency Brake Assistance
- IHC** - Intelligent Headlight Control
- SLR** - Single – Lens Reflex Camera
- LCS** - Lane Control Signs
- TSA** - Transportation Security Assistance
- RACam** - Radar and Camera System
- KKT (Karush – Kuhn – Tucker)** – a method for solving function minimization with a previously known constraint function
- GB** - Gigabyte

# Sadržaj

1.	Uvod.....	1
2.	Fotometrijsko izjednačavanje videozapisa .....	4
2.1.	Primjer uzorka .....	4
2.2.	Krivulje podešavanja boje .....	5
2.3.	Procjena dobitka globalne korekcije .....	6
2.4.	Procjena krivulje tonskog preslikavanja .....	7
3.	3D Prikaz ADAS.....	10
3.1.	Trenutni 3D prikaz .....	10
3.2.	Prikaz sustava .....	12
3.3.	Sustav 3D okruženja.....	12
3.3.1.	<i>Model broda.....</i>	13
3.3.2.	<i>Usmjeravanje leća ribljeg oka pomoću kolinearne konstante i točaka vrhova .....</i>	15
3.3.3.	<i>Transformacija koordinata pomoću virtualne površine za snimanje .....</i>	19
3.3.4.	<i>3D Preslikavanje tekstura .....</i>	21
3.3.5.	<i>Spajanje slike .....</i>	22
3.4.	Ekperimentalni rezultati.....	24
3.4.1.	<i>Ispitivanje predloženog kalibracijskog algoritma s tradicionalnim kalibracijskim algoritmom..</i>	24
3.4.2.	<i>Ispitivanje sustava dok je automobil u mirovanju ili u pokretu.....</i>	25
3.4.3.	<i>Testiranje sustava s drugim sustavima istog tipa.....</i>	25
4.	Trenutne ADAS solucije.....	27
4.1.	Potreban hardver za ADAS .....	30
4.1.1.	<i>Senzor kratkog dometa – Lidar (SLR).....</i>	30
4.1.2.	<i>SRR senzor .....</i>	30
4.1.3.	<i>Višenamjenska kamera.....</i>	31
5.	Rješenje za prikaz okruženja kod ugrađenih sustava.....	32
5.1.	KKT uvjeti.....	32
5.1.1.	<i>Uvjet izvedivosti .....</i>	33
5.1.2.	<i>Stanje gradijenta .....</i>	33
5.1.3.	<i>Komplementarna slabost.....</i>	33
5.1.4.	<i>Pozitivni Lagrange množitelji .....</i>	33
6.	Rezultati provedenog testiranja.....	41
6.1.	Izjednačavanje RGB i Y komponente .....	41
6.1.1.	<i>RGB komponenta.....</i>	41
6.1.2.	<i>Y komponenta .....</i>	44
6.2.	Rezultati različitih granica KKT-a .....	47
6.2.1.	<i>RGB komponenta.....</i>	47
6.2.2.	<i>Y komponenta .....</i>	54
6.3.	Podešavanje svjetline slike .....	61
6.3.1.	<i>RGB komponenta.....</i>	62
6.3.2.	<i>Y komponenta .....</i>	64
6.4.	Usporedba optimizacija po RGB koristeći YCbCr .....	66

6.5. Zapis bez izjednačavanja.....	71
6.6. Direktna usporedba slika.....	75
7. Zaključak.....	79
8. Literatura.....	82
Popis slika.....	85

# 1. Uvod

Sustav kamera za okruženje je u nastajanju, te se koristi ADAS tehnologija koja pomaže vozačima u sigurnijoj vožnji i parkiranju. Tipično okruženje čine sustav kamere oko vozila koje renderiraju i prikazuju virtualnu sliku u 360 stupnjeva na vozačevom ekranu. Kako bi se uspješno implementirao pogled od 360 stupnjeva, virtualna kamera je postavljena iznad vozačevog vozila u ptičjoj perspektivi.

Sintetizira se spajanjem ulaznih slika sa četiri ili više široko-kutne kamere prikazujući pogled iz ptičje perspektive. Kamere su obično instalirane sprijeda i straga na odbojnicima i bočnim rubovima. Zbog različitog osvjetljenja scena, AE kamere i AWB, boja i osvjetljenje istog objekta snimljenog različitim kamerama može biti vrlo različita. Stoga je jedan od glavnih problema sa kojim se možemo susresti taj da slika pati zbog različite svjetline i neusklađenosti boja po cijelom ekranu, što rezultira vidljivim kutovima na ekranu. To ukratko znači da se slika vidi da je spajana, što se pokušava izbjeći.

Izazov fotometrijskog poravnanja za okruženje sustava kamera nalazi se u:

- Istovremenom uspoređivanju više (najmanje četiri a najviše šest) ulaznih slika koje pokrivaju vidno polje od 360 stupnjeva koje može imati značajno različite uvjete osvjetljenja u različitim dijelovima scene.
- Takva aplikacije je ciljana na ugrađene sustave u automobilima te bi naš algoritam trebao biti učinkovit u prvom slučaju te ujedno i u maloj potrošnji memorije s obzirom na hardver.

Fotometrijsko poravnanje istraženo je u literaturi za izradu spajanja slika s više kamera [5, 6, 2, 7] i stetoskopsko 3D prikazivanje slika [31]. Međutim niti jedna od ovih metoda se ne može prilagoditi automobilskom okruženju na odgovarajući način. Han i suradnici [31] tvrde da slika odgovara intenzitetu histograma dvije stereo slike u globalnom smislu da se kompenzira razlika svjetline. Ova tehnika se ne generalizira dobro u slučajevima gdje različite kamere imaju različite poglede u zajedničkom prostoru. Brown i suradnici [5] te Lue i suradnici [2] koriste linearni model za ispravljanje fotometrijskog neusklađivanja za panoramsko spajanje slike i okružujući prikaz sinteza, odnosno njihov linearni model ne može u potpunosti ispraviti nelinearno izobličenje boje u uvjetima slabijeg i slabog osvjetljenja. Uz neometanu optimizaciju, njihove metode također trpe gubitak kontrasta. Suen i suradnici [29] i Uyttendaele i suradnici [30] predložili su lokalne adaptivne metode fotometrijske korekcije. Takve metode zahtijevaju prostorno filtriranje susjedstva kod svakog piksela, koje je računski intenzivno, pa nisu pogodne za ugrađene sustave.

U ovom radu nastojimo opisati novo fotometrijsko poravnanje algoritma koji je ujedno i vrlo učinkovito u fotometrijskom poravnanju s više kamera te je ujedno i računski učinkovito.

Naš algoritam procjenjuje optimalne krivulje mapiranja slike za prikaz, točnije jedan ulaz za kameru i svaki kanal u boji, pod pretpostavkom da se ne kontroliraju AE (Automatic Exposure) kamere i AWB (Auto White Balance).

Krivulja tona preslikavanja globalna je korekcija bez prostorne ovisnosti koja se može učinkovito primijeniti, tako da je pogodna za ugrađene sustave.

Da bismo stvorili kompozitni pogled iz ptičje perspektive, rješenje za okruženi pogled mora sadržavati dva ključna koraka:

- Geometrijsko poravnanje (usmjeravanje),
- Sinteza kompozitnog pogleda

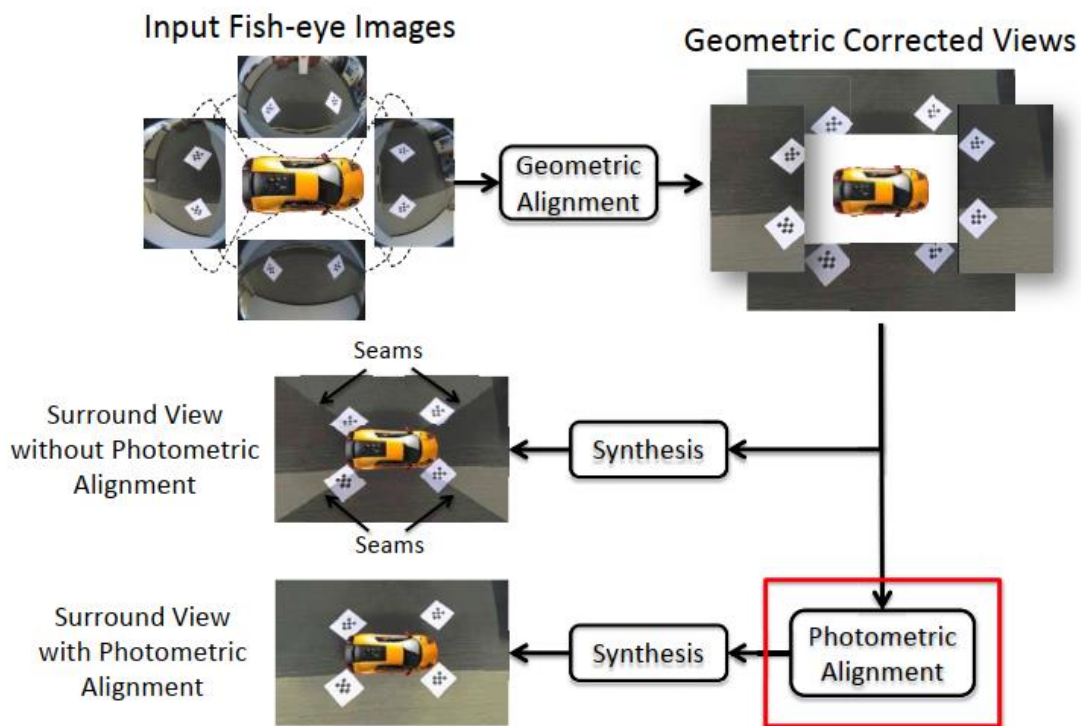


*Slika 1.1 Automobilski sustav za prikaz okruženja. U ovom primjeru prikazane su četiri širokokutne kamere (koje predstavljaju riblje oko) oko automobila, te je svaka od kamera okrenuta u drugom smjeru kako bi pokrila različite kutove.*

Geometrijsko poravnanje ispravlja distorziju leće ribljeg oka i primjenjuje se pravilna transformacija perspektive svakom pogledu tako da su svi prilazi za ulaze zabilježeni u ravnini tla iz pogleda ptičje perspektive. Nakon geometrijskog poravnanja, svaki piksel je složeni u prikaz okruženja, a generira se odabirom jednog ili kombinirajući više piksela iz odgovarajućih ulaznih slika. Taj se proces naziva sintezom.

Međutim, bez odgovarajućeg fotometrijskog poravnanja, složeni prikaz trpjet će inkonzistenciju boje i osvjetljenja te će biti vidljive granice spajanja slika.

Zbog toga je važno izvesti fotometrijsku korekciju nakon geometrijskog poravnanja i prije sinteze da bi se postigao kvalitetan pogled okruženja. Razlika između spajanih prikaza sa i bez fotometrijskog poravnanja je prikazana na slici 1.2.



Slika 1.2 Ilustracija sustava za prikaza okruženja sa i bez fotometrijskog poravnanja.

## 2. Fotometrijsko izjednačavanje videozapisa

Cilj fotometrijskog poravnanja za prikaz sustava okruženja je uskladiti ukupnu svjetlinu i različitost boja, takvih da se složeni prikaz slika čini kao da je snimljen jednom kamerom postavljenom iznad vozila.

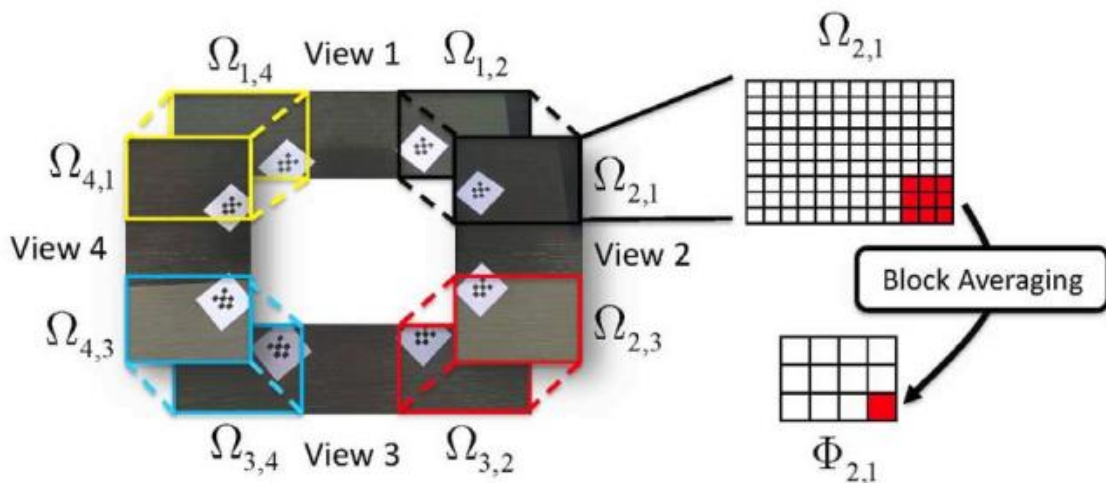
Da bi se to postiglo, može se koristiti krivulja tonskog preslikavanja za svaki prikaz i svaku boju kanala tako da je razlika u preklapajućim regijama susjednih slika minimalna.

Algoritam se sastoji od tri ključnih koraka:

1. Odabiranja podataka iz svake ulazne slike i izračuna statistike koja će se koristiti za analizu fotometrijskog pomaka. Označit će se statistike izračunate iz odabranih podataka kao uzroci u sljedećim diskusijama.
2. Procjenjuje se vrijednosti korekcije dobitka za svaki kanal u boji i svaki pogled iz ovih koraka.
3. Može se procijeniti će se optimalna funkcija mapiranja tonova za svaku kameru ograničenih optimalnih vrijednosti izračuna dobivenih iz prethodnih koraka. Izračuni dobiveni kao inicijalni rezultati korekcije prikazuju početnu grubu korekciju, a krivulje tona preslikavanja prikazuju daljnje fino podešavanje.

### 2.1. Primjer uzorka

Odabir pravih podataka slike će biti korišten za fotometrijsku korekciju, te je ujedno on i presudan. Koristimo samo slikovne podatke koji padaju u preklapajuća područja susjednih slika jer ovi podaci odgovaraju istim objektima u fizičkom svjetlu koje su zabilježile dvije susjedne kamere. Područje preklapanja je također područje gdje se nalazi spajanje između dvije susjedne slike. Odgovarajuće svjetline i boje između svakog para preklapajuće regije mogu smanjiti vidljivost samog spajanja.



Slika 2.1 Slika preklapajućeg područja i presjeka blokova. Pod pretpostavkom da je geometrijsko poravnanje već izvedeno na svakom je prikazu  $\Omega$  skup piksela iz preklapajućeg područja.  $\Omega_{m,n}$  sadrži skup piksela iz preklapajućeg područja.  $\Omega_{m,n}$  sadrži piksele iz prikaza.

## 2.2. Krivulje podešavanja boje

Globalne krivulje prilagođavanje boja dizajnirane su na temelju uzorka boja unutar područja preklapanja kompozita prikazane slike. Označavalo skup piksela u pogledu  $m$  koji pada u područje preklapanja između pogleda  $m$  i  $n$  kao  $\Omega_{m,n}$  gdje je  $m = 1, 2, 3, 4$ , dok je  $n = (m + 1) \bmod 4$ . Vrijednost  $m$  indeksira trenutni prikaz, a  $n$  indeksira susjedni prikaz u smjeru kazaljke sata.

Nadalje, koristimo  $\Omega_{m,n}$  za označavanje skupa piksela u pogledu  $n$  koji pada u isto područje preklapanja kao  $\Omega_{m,n}$ . Kako bismo umanjili utjecaj šuma fotoaparata na podudaranje boja, a također i smanjili složenost računanja, možemo provjeriti svaku preklapajuću regiju prosjekom blokova.

Skup uzoraka prosjeka blokova izračunatih iz  $\Omega_{m,n}$  je označeno kao  $\Phi$ .

Da bi isključili neusklađene uzorke boja zbog ne-komplementarnih objekata, definiramo  $S_{m,n}$  kao skup unutarnjih blokova u preklapajućim regijama.



$$S_{mn} = \left\{ [i, j] \mid \frac{1}{3} \sum_{k=1}^3 (\Omega_{m,n}^{(k)}[i, j] - \Omega_{n,m}^{(k)}[i, j])^2 < T, [i, j] \in [1, N_{mn}] * [1, M_{mn}] \right\} \quad (1.1)$$

$N_{m,n}$  i  $M_{m,n}$  su broj blokova u  $\Omega_{m,n}$  u okomitom i vodoravnom smjeru. Kako su ulazne slike u RGB formatu, svaki uzorak ima tri dimenzije indeksirane s k. T predstavlja ulazni prag.

Samo uzroci u S koristit će se za krivulju pojačanja i procjene.

### 2.3. Procjena dobitka globalne korekcije

Početna korekcija temelji se na multiplikativnom modelu dobitka. Funkcija korekcije može se izraziti kao:

$$\begin{aligned} \hat{f}_l^{(k)}[i, j] &= g_l^{(k)} * f_l^{(k)}[i, j] \\ [i, j] &\in [1, H_1] * [1, H_2] \end{aligned}$$

$f_l^{(k)}[i, j]$  predstavlja ulaznu sliku,  $l = 1, 2, 3, 4$  je pogled indeksa, a  $k = 1, 2, 3$  je indeks kanala u boji.  $H_1$  i  $H_2$  su dimenzije ulaznih okvira.  $g_l^{(k)}$  je dobitak za rješenje prikaza i kanal boje k da bi se podudarao s globalnom svjetlinom i bojom.

Za svaki kanal boja k, optimalan dobitak za sve prikaze  $g_1, g_2, g_3, g_4$  su zajednički optimizirani kako bi se minimizirala ukupna odstupanja između odgovarajućih uzoraka iz iste preklapajuće regije.

$$\{g_1^{(k)*}, g_2^{(k)*}, g_3^{(k)*}, g_4^{(k)*}\} = \arg \min \{g_1^{(k)}, g_2^{(k)}, g_3^{(k)}, g_4^{(k)}\}$$

$$\left\{ \sum_{\substack{m=1 \\ n \equiv (m+1) \\ \text{mod } 4}}^4 \frac{1}{\rho_{mn}} \sum_{[i, j] \in S_{mn}} (g_m^{(k)} \Omega_{m,n}^{(k)}[i, j] - g_n^{(k)} \Omega_{n,m}^{(k)}[i, j])^2 \right\} \quad (2.2)$$

$$\text{Podložan } g_l^{(k)} \geq 1, \text{ za } l = 1, 2, 3, 4$$

$$k = 1, 2, 3$$

$\rho_{m,n}$  je broj elemenata u  $S_{mn}$ . Imamo na umu da se optimalni dobitci rješavaju neovisno za svaki kanal u boji.

Ograničenja nejednakosti služe u tri svrhe:

1. Regulaciju troškovne funkcije,
2. Izbjegavanje artefakta u boji kod zasićenih regija slike,
3. Očuvanje kontrasta slike.

Budući da je to problem konveksnog kvadratnog programiranja s ograničenjima nejednakosti, koristili smo standardnu optimizaciju KKT da bismo pronašli optimalne dobitke [34].

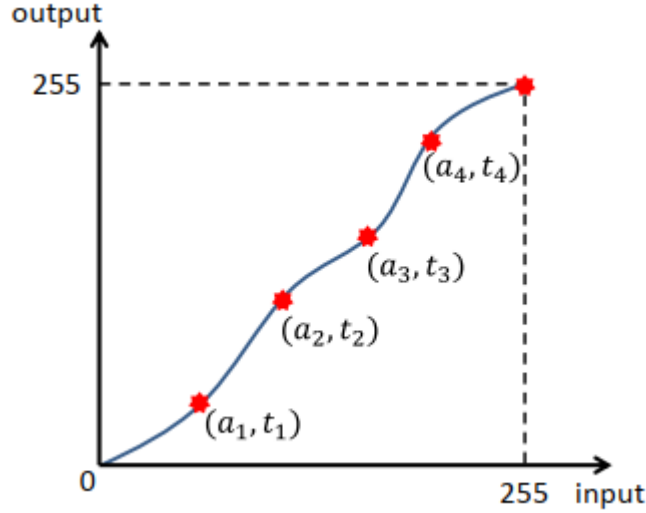
## 2.4. Procjena krivulje tonskog preslikavanja

Procjene krivulje tonskog preslikavanja se mogu primijeniti na ulazne slike. Koristi se za dodatno poboljšanje slike, no međutim mi to nećemo koristiti u našem algoritmu.

Nakon dobivanja  $\{g_l^{(k)*}, k = 1, 2, 3 \text{ i } l = 1, 2, 3, 4\}$ , vrijednosti se primjenjuju na ulazne slike za početnu fotometrijsku korekciju. Zatim ponovno prikupljamo unutarnje uzorke iz preklapajućih regija u svrhu dobivanja ispravljenih slika  $f_l^{(k)}$  ( $k = 1, 2, 3, l = 1, 2, 3, 4$ ). Maknemo to iz ažuriranih uzoraka iz preklapajućih područja kao  $\Phi^{m,n}$  i  $\Phi^{n,m}$ , i ažurirani skup unutarnjih indeksa uzoraka  $\hat{S}^{mn}$  kao  $m = 1, 2, 3, 4, n = (m + 1) \bmod 4$ .

Iako dobitci mogu nadoknaditi većinu neusklađenosti svjetline ili boje, to ne može u potpunosti ispraviti nelinearno neusklađivanje pod zahtjevnim uvjetima osvjetljenja i pri vrlo niskim ili visokim rasponom piksela.

Krivulja tona preslikavanja  $T$  je nelinearna funkcija koja preslikava ulaznu vrijednost između 0 i 255 u ulaznu vrijednost između 0 i 255, s dvije fiksne krajnje točke, npr.  $T(0) = 0$  i  $T(255) = 255$ . Veći stupanj slobode koji dolazi s preslikavanjem tonova krivulja ga čini dobro prilagođenim za ispravljanje nelinearno neusklađenih podataka.



Slika 2.2 Ilustracija krivulje tona preslikavanja i sidrenih točaka.

Slično kao kod optimalne procjene pojačavanja, optimalnom tonu krivulje mapiranja procjenjuju se zajednički za sva četiri ulaza kamere i neovisno za svaki kanal u boji, tako da je nepodudarnost piksela u preklapajućim regijama dodatno smanjena s obzirom na dodatni stupanj slobode u korelaciji modela.

$$\{T_1^{(k)*}, T_2^{(k)*}, T_3^{(k)*}, T_4^{(k)*}\} = \arg \min \{T_1^{(k)}, T_2^{(k)}, T_3^{(k)}, T_4^{(k)}\}$$

$$\left\{ \sum_{\substack{m=1 \\ n \equiv (m+1) \\ \text{mod } 4}}^4 \frac{1}{\hat{\rho}_{mn}} \sum_{[i,j] \in \mathcal{S}_{mn}} \left( T_m^{(k)}(\hat{\Phi}_{m,n}^{(k)}[i,j]) - T_n^{(k)}(\hat{\Phi}_{n,m}^{(k)}[i,j]) \right)^2 \right. \\ \left. + \beta \sum_{\substack{m=1 \\ n \equiv (m+1) \\ \text{mod } 4}}^4 \frac{1}{\hat{\rho}_{mn}} \sum_{[i,j] \in \mathcal{S}_{mn}} \left( T_m^{(k)}(\hat{\Phi}_{m,n}^{(k)}[i,j]) - \hat{\Phi}_{n,m}^{(k)}[i,j] \right)^2 \right\}, \quad (2.3)$$

$$k = 1, 2, 3$$

$\mathcal{S}^{mn}$  je skup ažuriranih indeksa unutarnjih blokova,  $\hat{\rho}^{mn}$  je ukupni broj elemenata u  $\mathcal{S}^{mn}$ , a  $\beta$  je kazneni faktor.

Prvi pojam u troškovnoj funkciji ima cilj smanjiti nedosljednosti u preklapajućim regijama, a drugi pojam ograničava da preslikane vrijednosti također ne odstupaju mnogo od vrijednost korigirane dobicima, jer dobici pružaju dobre početne procjene i krivulje tonskog preslikavanja

su samo za fino podešavanje. Takvo ograničavanje pomaže stabiliziranju rješenja, zbog čega je važno prvo procijeniti dobitak.

Da bismo pojednostavili problem optimizacije, svaku krivulju tona karakteriziramo setom sidrenih točaka  $(a_m^{(k)}, T_m^{(k)})$ ,  $m$  indeksira pogled,  $k$  indeksira kanal u boji, a  $i = 1, 2, \dots, p$  indeksira točke sidrenja.

Dalje trebamo popraviti  $a_{m,i}^{(k)}$  ( $i = 1, 2, \dots, p$ ) za svaki pogled kanala u boji ravnomjernim raspoređivanjem unutar raspona podataka o uzorku duž osi  $x$ . Sad nam ostaje samo  $T_{m,i}^{(k)}$  ( $i = 1, 2, \dots, p$ ) koje treba optimizirati. Ostatak krivulje tona dobiva se Hermite interpolacijom [35] iz optimalno sidrenih točaka. Približavamo tonu funkciju mapiranja s linearnom funkcijom koja povezuje sve točke sidra i sa fiksnim krajnjim točkama na  $(0, 0)$  i  $(255, 255)$ . Zatim iterativno i sekvencijalno ažuriramo  $T_{m,i}^{(k)}$  za  $m = 1, 2, 3, 4$  i  $i = 1, 2, 3, \dots, p$  zadržavajući ostale sidrene fiksne točke.

Budući da je varijacija jedne sidrene točke utječe samo na segmente linija s obje strane sidrene točke, optimizacija se svodi na rješavanje niza problema s kvadratnim programiranjem. Kada imamo  $\{ T_l^{(k)*}, l = 1, 2, 3, 4 \text{ i } k = 1, 2, 3 \}$ , postiže se konačna fotometrijska korekcija.

$$\bar{f}_l^{(k)}[i, j] = T_l^{(k)*}(\hat{f}_l^{(k)}[i, j]), [i, j] \in [1, H_1] * [1, H_2] \quad (2.4)$$

$\{ \bar{f}_l^{(k)}, l = 1, 2, 3, 4, k = 1, 2, 3 \}$  su fotometrijski ispravljene ulazne slike koje se mogu koristiti za sintezu konačne kompozicije slika.

### 3. 3D Prikaz ADAS

Kao glavno prijevozno sredstvo u modernom društvu, automobil se razvija prema trendu inteligencije, automatizacije i udobnosti. 3D prikaz okruženja pomaže vozačima da budu svjesniji okruženja prilikom vožnje i eliminira čovjeku vidljive slijepe točke. Sustav koristi četiri leće za prikaz ribljeg oka, instalirane oko vozila kako bi snimale sliku iz određenih kutova. Tada se prema obrascu snimanja slike, kalibraciji, spajanju slike i stvaranju scena, stvara 3D okruženje. Da bismo postigli performanse i jednostavnost u prikazu, u stvarnom vremenu koristimo samo jednu sliku za dovršavanje kalibracije kamere pomoću posebno dizajnirane komandne ploče.

U procesu spajanja slika, izrađen je 3D brodski model koji će biti primjer, odnosno baza za prikaz, pri čemu će se koristiti algoritmi za mapiranje tekstura i algoritmi za spajanje slike kako bi se očuvale stvarne informacije o teksturama.

Algoritmi koji se koriste u ovom sustavu mogu smanjiti računsku složenost i poboljšati učinkovitost spajanja slika. Poboljšana je i određenost okruženja čime se optimizira iskustvo sustava pod pretpostavkom da će očuvati informacija okoline [11].

#### 3.1. Trenutni 3D prikaz

Trenutačno, autonomna vozila su vrlo aktualna tema u akademiji i industriji. Međutim, kako praktična vozila učiniti praktičnijima nije samo tehnički problem, već između ostalog uključuje i aspekte sigurnosti te pravna i društvena prihvaćanja [6].

Suprotno tome, napredni sustavi za pomoć vozačima (ADAS) koji uključuju ljudsku interakciju praktičniji su za aplikacije [7]. Nominalno otkrivanje u prometno implementiranim scenama kroz prostorno kretanje pokreta smanjila bi se mogućnost prometnih nesreća koje nastaju zbog vozačevih mrtvih kutova.

Međutim autori u [7] također primjećuju da je gotovo nemoguće dizajnirati sustav koji može besprijekorno detektirati sve vrste nenormalnih događaja. Kamera za okruženje sastoji se od pogleda odozgo odnosno iz ptičje perspektive koja vozaču omogućuje gledanje okoline od 360 stupnjeva oko vozila [8], [9].

S jedne strane postojeći sustavi za pomoć pri sustavu za prikaz ne mogu generirati integrirane i prigodne slike zbog algoritma kalibracije. S druge strane, takvi algoritmi kao što je napomenuto [8], nisu dizajnirani da bi postigli performanse u stvarnom vremenu ili nisu testirani u ugrađenim platformama.

Najvažnije je da sustav pogleda iz ptičje perspektive može pružiti osnovnu perspektivu prikaza iznad vozila kao što je prikazano lijevo na slici 3.1., ili su slike jednostavno složene kao što je prikazano desno na slici 3.1.



*Slika 3.1 Izlazni rezultat postojećeg sustava. Lijevo je 2D prikaz, dok je desno 2D prikaz sa spojenim slikama iz svih kutnih kamera.*



*Slika 3.2 Slika zaslona sustava.*

Problem je u tome što će oba prikaza najvjerojatnije omesti vozača.

3D okružujući pogled za ADAS mogao bi riješiti ovaj problem pružanjem znatno boljeg osjećaja „uranjanja“ i svjesnosti pogleda okruženja. Snimka zaslona 3D pogleda okruženja prikazana je na slici 3.2., na kojoj se mogu vidjeti i očitati više informacija o okruženju samog vozila.

Iako je najveća tvrtka za izradu hardvera i softvera „Fujitsu“ izjavila da će njeni čipovi podržavati 3D pogled okruženja oko vozila [36], točni detalji ove tehnologije tek moraju biti završeni i implementirani.

Dalje u ovom radu ćemo prikazati prikaze sustava koji uključuje posebni algoritam kalibracije ribljeg oka, perspektivnu transformaciju, izgradnju modula, oblikovanje tekstura i linearno spajanje. Ovaj tip sustava može pomoći vozačima da budu svjesni vožnje i okoline oko vozila, eliminirajući vidljive mrtve točke i spriječiti sve vrste nevidljivih opasnosti.

### 3.2. Prikaz sustava

Sustav se sastoji od četiri leće ribljeg oka postavljene oko vozila i ekrana unutar upravljačke ploče. Minijaturni prikaz vozila je vidljiv na slici 3.3.



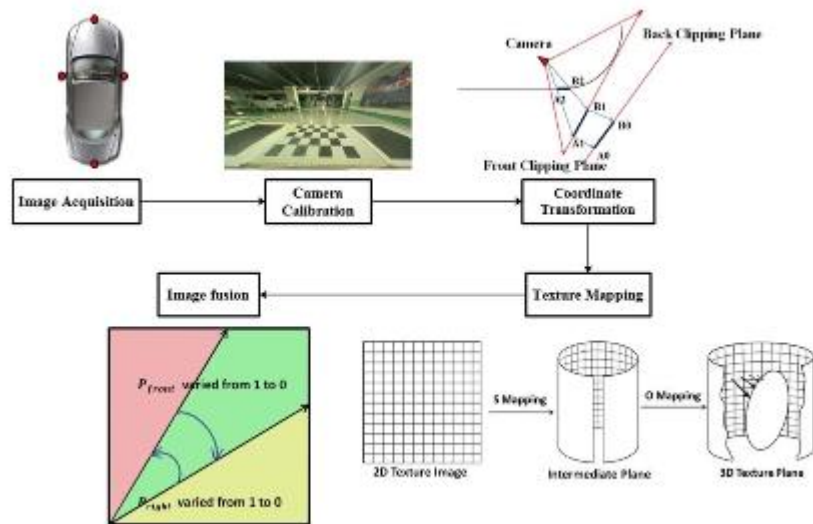
*Slika 3.3 Prikaz sustava unutar vozila.*

Za smanjenje troškova koriste se leće ribljeg oka „HK8067A“ koje koštaju nešto više od 3 američka dolara. Ove vrste leća pružaju širokokutni prikaz od 180 stupnjeva kako bi se osiguralo dovoljno preklapanja slika između svake pojedinačne slike. Kroz niz postupaka spajanja slika pomoću ugrađenog sustava koristeći „Freescale“ procesor može se formirati 3D prikaz sustava okruženja automobila.

### 3.3. Sustav 3D okruženja

Shema 3D sustava prikazana je na slici 3.4., a uključuje četiri glavna koraka:

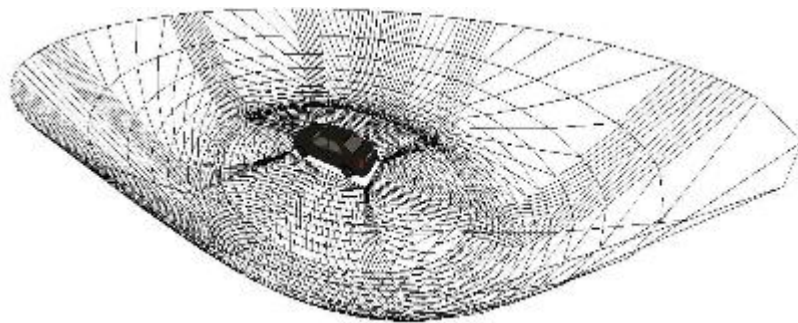
1. Kalibracija kamere,
2. Transformacije koordinata,
3. Mapiranje tekstura,
4. Spajanje slike.



Slika 3.4 Prikazuje dijagram ključnih koraka za dobivanje slike okruženja.

### 3.3.1. Model broda

U usporedbi s uobičajeno korištenim cilindričnim modelima, koji su spomenuti u [32], vodoravno dno i zid u obliku luka bit će u skladu s vizualnim navikama vozača kako bi sami vozači dobili širi prikaz okruženja.

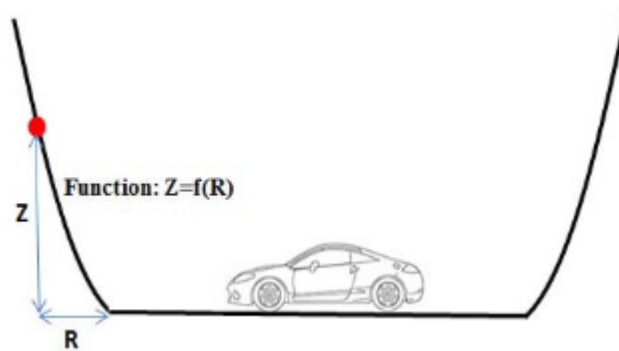


Slika 3.5 Prikaz modela Broda.

Konstrukcija 3D modela sastoji se od povezivanja točaka u 3D prostoru, u liniju te zatim ravninu i konačno tijelo. Nakon što se točke postave u određenom redoslijedu, nacrtava se 3D model prema potrebi.

S obzirom na učinak fuzije tekstura, model broda odabire funkciju  $Z = R^4$  kao funkciju rampe, prilikom koje je  $R$  udaljenost između točaka projekcije i ruba dna, a  $Z$  visine točke luka. Prikaz presjeka modela broda je prikazan na sljedećoj slici, 3.6.





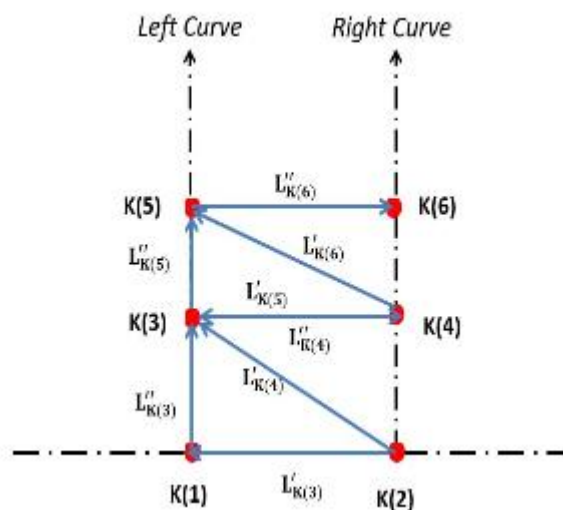
Slika 3.6 Pogled u presjeku 3D modela broda.

Kao što prikazuje pogled odozgo, model se zapravo sastoji od kruga elipsa. Točke modela su zapravo sjecišta linija koje se protežu od donjeg okvira i tih elipsa. Prema eksperimentima, površina nagiba je glađa kada gustoća točaka padne na 15. Drugim riječima, za crtanje modela izabrano je 15 točaka sjecišta na svakoj krivulji nagiba. Međutim, u gornjoj elipsi gdje je udaljenost između svake točke velika, dodajemo niz ravnih linija koje prolaze kroz uglove donjeg okvira i formiramo više sjecišta za izgradnju 3D modela. Zatim projiciramo ove točke u dno modela i izračunamo udaljenost između točke projekcije i ruba donje površine.

Funkcijom rampe, mogu se izračunati koordinate tih točaka nagiba. Dobiveni nagib je glatki a slike su ujedno i prirodnije.

Ovo ne da pruža vozačima širi vid, već i ubrzava postupak spajanja slika.

Dvije susjedne krivulje na nagibu mogu činiti paralelnu regiju. Prikaz konstrukcije točaka možemo vidjeti na slici 2.7.



Slika 3.7 Prikaz konstrukcije modela.

Slovo K na slici označava točku, a L označava liniju. Oznake u zagradama su brojevi točka ili linija.

Prvo označimo točke na lijevoj krivulji s neparnim brojevima, a na desnoj krivulji s parnim brojevima. Nakon toga pretpostavimo da postoji točka s oznakom K(n). Ako je ta točka na lijevoj krivulji, spojimo točke K(n-1), K(n-2) i K tim redoslijedom. U suprotnom spojimo točke K(n-2), K(n-1) i K(n). Određivanjem točaka na svim linijama pomoću ove metode će se svi trokuti crtati u istom smjeru. Nakon rješavanja svih točaka na određenom području, konstrukcija modela je završena.

### **3.3.2. Usmjeravanje leća ribljeg oka pomoću kolinearne konstante i točaka vrhova**

Budući da leće ribljeg oka koje su u uporabi mogu snimiti prizor s širokim kutom od 180 stupnjeva, četiri leće ribljeg oka oko vozila mogu osigurati da nema mrtvih kutova. Međutim time široki kut leće žrtvuje kvalitetu snimljene slike.

Postoji izobličenja koristeći leće ribljih očiju, određena svjetla koja su pozicionirana dosta daleko od optičkog centra [10], [37]. Izobličenje otežava prenošenje snimljenih informacija o slici [11]. Zbog toga je potrebno dobro kalibrirati kameru, ispraviti izobličenja i zatim ispraviti sliku kako bi se zadovoljile ljudske vizualne potrebe.

Tradicionalne mjere kalibracije kamere obično koriste otprilike 20 snimljenih slika s različitih položaja [12], [13], [14]. To znači da zahtjeva značajne ljudske interakcije, intervencije i vremenske troškove [38]. Prema tome, tradicionalne metode mjerenja pretpostavljaju model s „rupicom“ (pin-hole), koji se razlikuje od leća ribljih očiju.

Točnije, dobavljač leća daje popis podataka o krivulji polja koja prikazuje odnos između stvarne visine i referentne visine. Pomoću tog popisa može se ispraviti izobličena slika dobivena lećama ribljeg oka. Međutim, zbog cjelokupnog procesa izrade leća, sljedećeg centriranja i neusklađivanja između ravnine osjetnika i ravnine objektiva, navedeni popis možda neće biti sasvim točan.

Štoviše, leće ribljev oka instalirane u sustavu su vrlo jeftine i loše kvalitete što često može dovesti do netočnih podataka. Kroz mnoge eksperimente je utvrđeno da je optički centar najvažniji faktor koji utječe na ispravljenu sliku. Ako se može dobiti procjena optičkog središta, tada se još uvijek može koristiti popis zakrivljenosti polja.

Ono što je najvažnije je da možemo ispraviti sliku i procijeniti optičko središte pomoću određenog pravila te tako dobiti konačnu ispravljenu sliku.

S obzirom da će se sustav postavljati u proizvodnji automobila ili u trgovini, preferiraju se što manje ljudske intervencije, jednostavan i lagan rad s manje vremenskih troškova. Zbog toga se predlaže algoritam kalibracije leće ribljev oka koji koristi kolinearno ograničenje i rubne točke.

Algoritam ima dva važnija ograničenja, prvo ograničenje je da bi se kolinearne točke morale ispraviti kako bi bile nekolinearne, dok drugo ograničenje je da svjetlost kroz optičko središte ima manja izobličenja dok ono preko rubova leća ima najveće izobličenje.

Za to se prvo treba koristiti posebna ploča za provjeru koja se može ispisati i lako postaviti kao što je prikazano na slici 3.8.



*Slika 3.8 Specijalna kalibracijska ploča (označena crvenim linijama).*

Pomoću te kontrolne ploče kalibriramo kameru bez pomicanja vozila. U iskrivljenoj slici se može precizno pretraživati kutne točke blizu objektiva, jer te točke imaju relativno beznačajna izobličenja. U globalnim koordinatama ove su točke u istom retku ili istom stupcu. Kao što je prikazano na slici 3.8., kutne točke na ispravljenoj slici i dalje bi trebale biti kolinearne.

Pretpostavimo da je  $d(i)$  udaljenost između točke  $i$ , i prijanjajuće linije  $i$  da je  $\mu(i)$  težina svake točke. Tada znači da je ponderirani zbroj:

$$L = \sum_{i=1}^M \mu(i)d(i) \quad (3.1)$$

gdje  $M$  predstavlja broj od ukupnih rubnih točaka.

S obzirom na posebnu strukturu leće za riblje oko, što su točke udaljenije od optičkog središta, to će izobličenje biti veće.

Stoga bi težina svake točke trebala biti drugačija, a težina je postavljena uzimajući u obzir fizičku udaljenost točka od leće.

Uzimajući u obzir optičku središnju vrijednost, uvijek možemo dobiti potvrđenu sliku pomoću popisa podataka zakrivljenosti polja. Potom bi se kroz tražene kutne točke mogle uklopiti linije. Optička središnja koordinata prelazi se u određenom rasponu, te samim time slika izlazi iz kadra. Rješenje je da se fiksira središnja optička koordinata kako ne bi dolazilo do pogrešaka. To znači da nam treba odgovarajući optički centar.

Zbog posebne strukture leće ribljeg oka, izobličenje kraj ruba slike je veće. Kao što je prikazano na slici 3.9., iskrivljenost pravokutnika na bočnim stranama podloge je veća od one na sredini kontrolne ploče.



Slika 3.9 Perspektiva transformacije procesa. [6]

Osim toga, leće za riblje oko se postavljaju iznad zemlje, pri čemu se nalazi kut nagiba između vodoravnog položaja tla i same leće, zbog čega se pravokutnik pojavljuje kao trapez ili drugi nepravilni prostor.

Spomenute činjenice rezultiraju rezultatom da točke sa bočnog ugla nije lako otkriti. Ako izvršimo kalibraciju koristeći samo srednji ugao na kontrolnoj ploči (manjoj ploči), tada procjena optičkog centra neće biti točna što će utjecati na konačnu potvrđenu sliku.

Kako bismo otkrili kutne točke daleko od leća ribljih očiju, dizajniramo veliki kvadrat na rubu i dva puta izvršimo perspektivnu transformaciju.

Postupak otkrivanja ugla može se podijeliti na nekoliko koraka:

- Prvo se dobiva binarna slika segmentacijom ruba. Pored toga koristimo metodu adaptivnog ruba uzimajući u obzir nejednoliku svjetlinu slika. Prilagodljivi pikseli rubova određuju se rasponom vrijednosti susjednih točaka piksela.
- Drugo, proširenje slike koristi se za odvajanje veze svake crne kocke na kalibracijskoj ploči. Ova metoda zahtjeva element strukturiranja koji može biti kvadrat ili krug sa središnjom točkom. Vrijednost piksela ove središnje točke uspoređuje se sa vrijednošću svakog piksela slike. Tada će veća točka biti postavljena na vrijednost piksela slike.

Nakon ovog postupka, bijele točke piksela će se proširiti. Stoga se može smanjiti crni četverokut i prekinuti vezu između svakog četverokuta. Nadalje broj vrhova i kontura grafa koriste se za razlikovanje kvadrata.

Konačno neki restriktivni uvjeti, kao što su omjer slike, perimetar i površina, koriste se za uklanjanje ometanih podataka.

Nakon gornjih koraka, točke korektora mogu se otkriti. Kombinacijom tih otkrivenih većih kutnih točaka kvadrata i malih kutnih točaka kvadrata može se izvršiti kalibracija kamere, kao što je prikazano na slici 3.9.

Neke kutne točke je teže za prepoznati jer su odaljene od optičkog središta i imaju velika izobličenja. Stoga se izvode perspektivna projekcija na originalnoj slici. Kako je ploča postavljena na zemlju, ako gledamo odozgo ispada nam da će pravokutnik biti normalnog oblika. Slijedom toga, možemo dobiti sliku ptičje perspektive u kojoj lako možemo prepoznati kutne točke. Matrica perspektivne transformacije sadrži devet parametara,

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad (3.2)$$

Međutim plan mape je paralelan u odnosu na originalan plan i formula je homogena. Prema tome  $[a_{33}] = 1$ . Prikaz matrica transformacije sadrži 9 parametara, prikazanih redoslijedom.

$$(x, y, 1) = (u, v, 1) = x \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad (3.3)$$

Među njima su  $(u, v)$  koordinate piksela u iskrivljenoj slici, a  $(x, y)$  su koordinate piksela u transformiranoj slici. Matrica transformacije dobiva informaciju o skaliranju, kraćenju, rotaciji i translaciji. Pomoću 4 pouzdanih parova rubnih točaka, koje su od njih 4 najbliže leći, mogu se izračunati pomoću pretvorbe matrice. Stoga svaku točku transformirane slike možemo dobiti na sljedeći način:

$$x = \frac{a11u + a21v + a31}{a13u + a23v + 1} \quad (3.4)$$

Naknadno izvodimo drugu perspektivnu transformaciju na slici iz ptičje perspektive. Stoga možemo izračunati koordinate kutnih piksela velikih pravokutnih točaka u originalnoj slici. Sa svim dobivenim kutnim točkama, algoritam ugradnje linija koristi se za traženje optičkog sjedišta i ispravljene slike, kao što je prikazano u formuli iznad za ponderirani zbroj.

U usporedbi s algoritmom tradicionalne kalibracije, predloženi algoritam bilježi samo jednu sliku za dovršavanje kalibracije, stoga je prikladniji za sustav pomoći vozaču.

Treba imati na umu da tradicionalni algoritmi mogu ispraviti samo bližnje piksele, dok moderniji mogu ispraviti sve točke.

### 3.3.3. Transformacija koordinata pomoću virtualne površine za snimanje

Nakon dobivanja optičkog centra i ispravljenih slika, mora se transformirati 2D slika u 3D model broda. Međutim, direktan odnos između 2D slike i 3D modela je teško dobiti. Dakle, plan virtualne slike uspostavljen između 2D slike i 3D slikovne ravnine kroz perspektivnu transformaciju i formaciju. Leća je montirana na vrhu stošca. Prema matrici ribljeg oka, kutu vida i orijentacijama matrica stošca može se potvrditi.

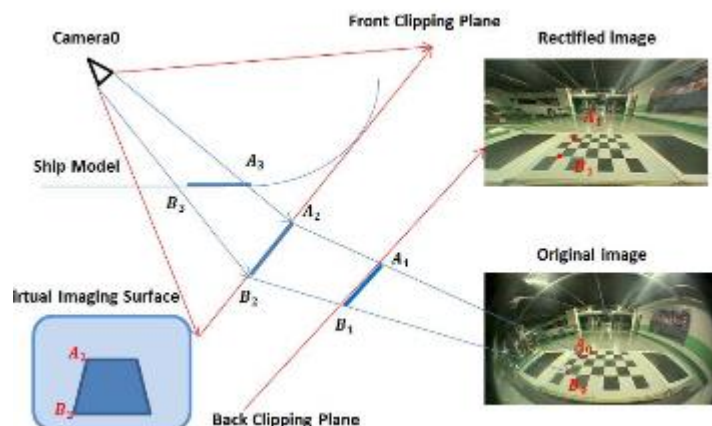
$$\left\{ \begin{array}{l} \begin{pmatrix} \frac{2N}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2N}{t-b} & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & -1 & 0 \end{pmatrix} \\ \begin{cases} a = -\frac{F+N}{F-N} \\ b = -\frac{2NF}{F-N} \end{cases} \end{array} \right. \quad (3.5)$$

N je udaljenost od oka do prednje ravnine sjecanja, a F je udaljenost od oka do stražnje ravnine sjecanja. r, l, i i b su desna granična vrijednost, lijeva granična vrijednost, gornja granična vrijednost i donja granična vrijednost ravnine projiciranja.

Koordinate piksela točaka na ravnini virtualne slike može se dobiti množenjem gornje matrice s koordinatama piksela kutnih točaka na 3D modelu.

$$\begin{pmatrix} \frac{2Nx}{r-l} \\ \frac{2Ny}{t-b} \\ az+b \\ -z \end{pmatrix} = \begin{pmatrix} \frac{2N}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2N}{t-b} & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & -1 & 0 \end{pmatrix} x \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (3.6)$$

Zatim se matrica perspektivne transformacije može riješiti kombiniranjem koordinata piksela odgovarajuće rubne točke na 2D ravnini. Nakon dobivanja matrica dvije transformacijske projekcije, svaka točka na 2D slici može se pretvoriti u 3D model pomoću virtualne slike. Prikaz shematskog postupka možemo vidjeti na slici 3.10.



Slika 3.10 Proces transformacije koordinata.

### 3.3.4. 3D Preslikavanje tekstura

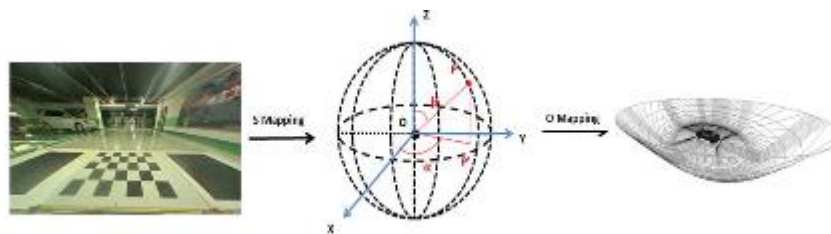
Nakon rečenih koraka i postupaka, dobivaju se koordinate svakog piksela u 3D modelu broda. Ako su preslikane samo tražene točke, tada rezultirajuća slika okruženja nije prirodna i uzrokovat će gubitak podataka o slici. Zbog toga se koristi preslikavanje tekstura. U usporedbi s tradicionalnim načinom preslikavanja, preslikavanje tekstura je složena grafička tehnologija a najčešće se koristi za izražavanje geometrijskih detalja tekstura i svjetlosti. Dobivena slika okruženja postat će životopisnija i prirodnija.

Preslikavanje tekstura proces je koji preslikava 2D teksture na 3D površinu. Ovaj je postupak sličan postavljanju slike na površinu 3D objekta radi poboljšanja osjećaja stvarnosti. Srž ove metode je uvođenje srednjeg 3D kartografskog prikaza kao posrednog medija. Osnovni postupak se može postići kroz sljedeća dva koraka [15][16].

$$(u, v) \rightarrow (x', y', z') \rightarrow (x, y, z) \quad (3.7)$$

$(u, v)$  su koordinate 2D prikaza.  $(x', y', z')$  su površine jednostavnog 3D objekta.  $(x, y, z)$  su koordinate 3D modela.

Prvo, 2D tekstura je preslikana na jednostavan 3D objekt, kao što su sfera, kocka, cilindar i tako dalje. Zatim se uspostavlja sljedeće preslikavanje  $T(u, v) \rightarrow T'(x', y', z')$ . Ovo preslikavanje naziva se S preslikavanje koje preslikava 2D teksturu u sferu s polumjerom R. Proces preslikavanja možemo vidjeti na slici 3.11.

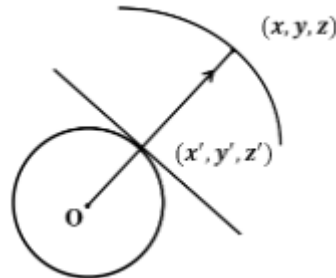


Slika 3.11 Mapiranje u dva koraka.

$$\begin{cases} x = R * \cos \alpha * \sin \beta \\ y = R * \sin \alpha * \sin \beta \\ z = R * \cos \beta \end{cases} \quad (3.8)$$



P je točka na sferi kao što je prikazano na slici 3.11. Preslikavanjem linije OP u ravninu XOY,  $\alpha$  je tada kut između linije projekcije i osi X. Kut između linije OP i osi Z je  $\beta$ , gdje  $0 \leq \alpha \leq 2\pi$ ,  $0 \leq \beta \leq 2\pi$ . Potom se tekstura na površini posrednog medija preslika na površinu konačnog objekta, kao što je prikazano na slici 3.12.



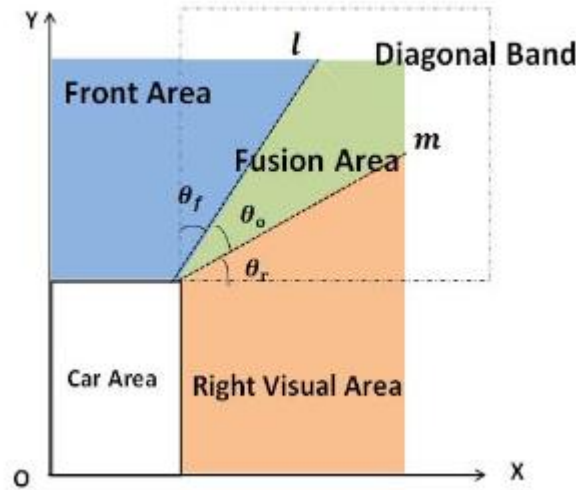
Slika 3.12 O preslikavanje.

Sjecište posrednog objekta i radijala koji povezuje točku O i  $(x, y, z)$  je  $(x', y', z')$ . Ova točka se smatra točkom preslikavanja. Taj opisani postupak se O preslikavanje.  $T'(x', y', z') \rightarrow O(x, y, z)$ . Kroz ta dva koraka, tekstura na 2D ravnini može se preslikati na površinu 3D modela broda kroz sfernu površinu. Izobličenje slike može se također smanjiti, a informacije o slici mogu se spremati u najvećoj mjeri [15][16].

### 3.3.5. Spajanje slike

Izračunom kalkulacije parametara kamera i pravila preslikavanja tekstura, utvrđen je relativni odnos između susjednih slika.

Projektirajući 3D model broda u prvi kvadrant, možemo primijetiti da će se područje podešenih leća najčešće preklapati. Da bi se smanjili preklapanje slika i učinili sliku što prirodnijom, potrebno je ukloniti preklapajuća područja sa slike 3.13. [17][18][33].



Slika 3.13 Projekcija modela u prvom kvadrantu.

Radi jednostavnosti, ovdje koristimo alpha spajanje. Kao što je prikazano na slici 3.13., postoje dvije granične regije. Prva, odnosno linija prednje segmentacije regije  $l$  i desna vidna regija segmentacije  $m$ . Za prednju regiju vrijednost mase piksela varira od 1 do 0, odgovarajući  $l$  i  $m$ . Za desnu vizualnu regiju, masa piksela varira od 1 do 0 što odgovara  $l$  i  $m$ . Za točku  $A$  u modelu, vrijednost piksela može se izračunati pomoću sljedeće formule.

$$\begin{cases} \alpha = \arctan\left(\frac{|y| - \omega/2}{|x| - l/2}\right) \\ P_{front} = \frac{\alpha - \theta_o}{\theta_f - \theta_o} \\ P_{right} = 1 - P_{front} \\ C_A = P_{front} * C_{front} * C_{right} \end{cases} \quad (3.9)$$

U ovoj formuli,  $\omega$  je širina vozila a  $l$  je duljina vozila.  $\theta_o$  i  $\theta_v$  su prikazani na slici 3.13. Ovisno o lećama ribljeg oka koje se koriste,  $\theta_o$  ovog sustava je  $15^\circ$ ,  $\theta_f$  je  $41.5^\circ$  i  $\theta_v$  je  $33.5^\circ$ .  $C_{front}$  i  $C_{right}$  predstavljaju vrijednosti piksela točke  $A$  u prednjem području i desnoj vizualnoj regiji.  $P_{front}$  i  $P_{right}$  su mase vrijednosti  $C_{front}$  i  $C_{right}$ .  $0 \leq P_{right} \leq 1$ ,  $0 \leq P_{front} \leq 1$ .

Ovaj algoritam ima dobar efekt spajanja. Štoviše, sustav slike daje prirodno povezan i gladak prijelaz. Također eliminira razlike u svjetlini i slikama, a vizualni efekti su znatno poboljšani [39].

### 3.4. Eksperimentalni rezultati

#### 3.4.1. Ispitivanje predloženog kalibracijskog algoritma s tradicionalnim kalibracijskim algoritmom

Prije nego što automobili napuste proizvodni proces, potrebno je na brzinu izvršiti neke provjere i ispitivanja. U ovom se postupku manje prednosti daje ljudskoj intervenciji. Međutim, tradicionalni algoritam kalibracije potrebno je više slika za dovršavanje postupka, pa je potrebno nekoliko puta mijenjati automobile ili kalibracijsku ploču [12]. Samim time tradicionalni algoritmi nisu dobro prikladni za ovakve aplikacije.

Suprotno tome, algoritam kalibracije koji spominjemo trebalo bi usvojiti u kalibracijsku ploču. Može upotrijebiti samo jednu sliku za ispravljanje izobličenja slika i dobivanje točnijih vrijednosti optičkog centra. Ta snaga odgovara sustavu i drugim jačim sustavima u realnom vremenu [12]. Kalibracija se provodi u zatvorenom okruženju u kojem je svjetlost blaga. Takvo okruženje pomaže u lakšem otkrivanju rubova. Ispravljena slika obrađena pomoću ovih algoritama prikazana je na slici 3.14(b), a uspoređuje se s tradicionalnim algoritmom na slici 3.14(a).



Slika 3.14 Usporedba dva algoritma, (a) je tradicionalan algoritam a (b) je predloženi algoritam.

Prema slikama, možemo jasno promatrati da je slika pomoću algoritma ispravljala izobličenja. Ne samo da otkriva kutove srednje ploče, već otkriva i kutove pravokutne kalibracijske ploče na rubu slike.

Međutim, tradicionalni algoritam ne može otkriti kutove pravokutne kalibracijske ploče. Također slika u (a) primjeru sadrži distorziju na rubu slike. Samo je središnja regija dobro kalibrirana.

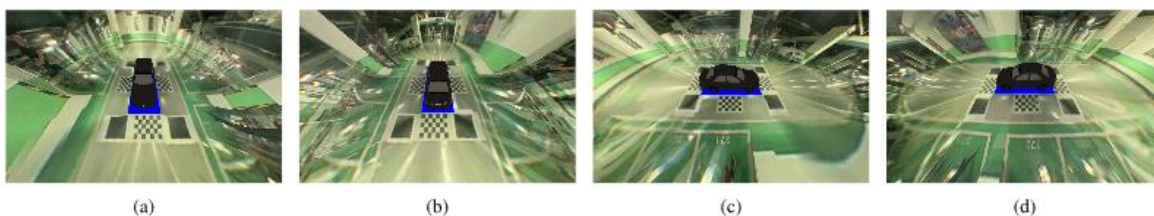
### 3.4.2. Ispitivanje sustava dok je automobil u mirovanju ili u pokretu

Parkirani automobil je u podzemnoj garaži gdje je prigušena svjetlost. Slika 3.15. prikazuje slike koje su uzete lećama ribljeg oka. Slike snimljene lećama ribljeg oka imaju sva iskrivljenja. Što je optičko središte udaljenije, to je distorzija veća. U međuvremenu, postoje susjedne regije u susjednim vizualnim regijama. Ako samo snimimo te slike, tada će informacije na slici dovesti vozača u zabludu.



Slika 3.15 Prikaz slika uzetih s lećama ribljih očiju. (a) prednji pogled, (b) stražnji pogled, (c) desni pogled i (d) lijevi pogled

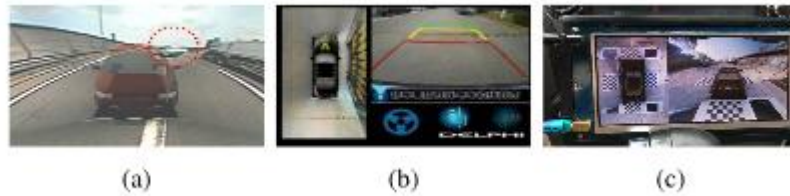
Kao što je prikazano na slici 3.16., distorzija je već ispravljena a tekstura je preslikana na 3D model na odgovarajući način. Prekriveno područje znatno je poboljšano u pogledu boje, svjetline i ostalih vrijednosti nakon procesa spajanja. Sve u svemu 3D prikaz okruženja može pomoći vozačima da prirodno dobiju znanje o svojem okruženju.



Slika 3.16 Prikaz okruženja u podzemnoj garaži. (a) prednji pogled, (b) stražnji pogled, (c) desni pogled i (d) lijevi pogled.

### 3.4.3. Testiranje sustava s drugim sustavima istog tipa

Tvrtka Fujitsu predlaže korištenje tehnologije videozapisa s 360 stupnjeva [36]. To je prikazano na slici 3.17(a). Još jedan sustav za okruženje od 360 stupnjeva prikazan je na slici 3.17(b). Te dvije slike koriste se kao usporedba.



*Slika 3.17 Usporedba s drugim sustavima. (a) Fujitsu sustav, (b) Delphi sustav i (c) moderni sustav.*

Tehnologija prikazana na slici pod (a) može stvoriti panoramsku sliku oko vozila ali slika nije prirodna i postoji velika razlika u svjetlini i boji između generirane slike i stvarnosti. Sustav prikazan na slici pod (b) stvara prirodniju sliku ali prikaz koji vozač može vidjeti iznad zemlje vrlo je ograničen u usporedbi s modernim sustavom kao što je prikazano na slici pod (c). Sustav pod (c) može stvoriti prirodnu sliku a vozači mogu dobiti uvid u kontrolnu ploču. Također pomaže vozačima da dobiju točnije podatke o svom okruženju.

## 4. Trenutne ADAS solucije

Sustavi smanjenja nesreća postaju presudni za automobilsku industriju jer potrošači puno više pažnje posvećuju sigurnosti. Kako bi se spriječile prometne nesreće, stvoren je plan puta za potpuno autonomnu vožnju. Razvoj sustava pomoću vozača počeo je sustavom kočenja protiv blokiranja (ABS) uvedenim u serijsku proizvodnju krajem 70-ih godina prošlog stoljeća.

Mapa koncepta sastoji se od sljedećih senzora:

- Proprioceptivni senzori – sposobni su otkriti i reagirati na opasnu situaciju analizirajući ponašanje vozila,
- Eksteroceptivni senzori – sposobni su reagirati u ranijoj fazi i predvidjeti moguće opasnosti (ultrazvuk, radar, infracrveno),
- Senzorske mreže – primjena više senzorskih platformi i mreža senzora prometa.

ADAS pruža dodatne informacije iz okruženja automobila kako bi podržao vozača i pomogao u provedbi kritičnih radnji. Sinkronizacija djelovanja vozača i informacija iz okoline ključna su za učinkovito djelovanje različitih primjena ADAS-a [19].

U sustavu adaptivnog tempomata (ACC) obično su potrebna tri radarska senzora jer se dva radara kratkog dometa koriste za otkrivanje objekata u susjednoj traci, a jedan radar dugog dometa koristi se za otkrivanje objekata u putu. Kako postaju popularni sustavi za pomoć vozačima (DAS) i vozila s aktivnom sigurnošću (AVS) s različitim funkcijama, nije rijetkost da se na vozilo ugrađuje više sustava.

Ako svaka funkcija koristi vlastite senzore i procesorsku jedinicu, to će otežati ugradnju i povećati troškove vozila. Kao kontra mjera provedena su istraživanja više funkcija u jedinstveni sustav i očekuje se da će olakšati instalaciju, smanjiti potrošnju električne energije i cijene vozila [20].

Granice ceste mogu pružiti korisne informacije za procjenu sigurnih staza vozila u inteligentnim vozilima. Mnoga prethodna istraživanja proučavala su otkrivanje granica na cesti, koristeći različite vrste senzora kao što su vid, radar i lidar. Osobito Lidar senzori pokazuju prednosti za vađenje granica na cesti, uključujući visoku rezoluciju i široko vodno polje.

Međutim niti jedna prethodna studija nije proučavala problem otkrivanja granica ceste kada bi ceste mogle biti strukturirane ili nestrukturirane [21].

Trenutačni trendovi na tržištu su integracija LKS i ACC, nazvanih LKS + ACC sustav. Ukupna struktura otkrivanja prometnih traka jednaka je uobičajenoj metodi korištenja monokularnog vida, inicijalizacija zasnovana na EDF, pod ROI za slojeve bazirane na lijevo ili desno i udaljenosti, mogućnost upravljanja temeljenog na filtrima ekstrakcija i uklapanje modela u svaki pod-ROI. Rezultati pokazuju da kao jednostavan prilagodljivi ROI može prevladati okuliranje traka i uznemiravanje susjednih vozila [22].

Oni koriste uklonjenu perspektivnu domenu danu inverznim perspektivnim mapiranjem (IMP) kako bi definirali brzi i učinkoviti model vjerojatnosti. Uz to Markov nasumična polja (MRF) omogućuju tretiranje ovisnosti između pokreta ciljeva, pored najsuvremenijih metoda za praćenje vozila, tj. neovisno praćenje cilja s Kalmanovim filtriranjem (KF) i zajedničkom praćenjem filtriranja čestica [23].

Drugi pristup temeljen na slici zasnovan je na sustavu stereo vizualnih kamera. Položaj, orijentacija i stanje punog gibanja, uključujući brzinu, ubrzanje i nižu brzinu detektiranog vozila, procjenjuju se iz oblaka. Ovaj oblak predstavlja 3D objektni model a izračunava se analizom slikovnih sekvenca u prostoru i vremenu, tj. spajanjem stereo vizije i obilježjima praćene slike ili procjenom vertikalne brzine [24][25].

Polazeći od automatizirane početne hipoteze vozila, praćenje se provodi pomoću proširenog Kalmanovog filtra. Filter kombinira znanje o okretanju točaka oblaka u svijetu s dinamičkim modelom vozila. Radarske informacije koriste se za poboljšanje otkrivanja objekata na temelju slike na velikim daljinama. Predloženi sustav primjenjuje se za predviđanje putanje ostalih suučesnika u prometu i trenutno radi na 25 Hz, slike rezolucije 640x480 [26]. Brzi pristupi temeljeni na slici mogu se realizirati i korištenjem ugrađenog sistemskog rješenja [27].

Prepoznavanje traka važna je komponenta potrebna za razne sustave za pomoć vozaču. Na primjer, upozorenje o odlasku sa prometne trake (LDW) i zadržavanje trake oslanjaju se na informacije dobivene algoritmom za procjenu traka. Važan korak postupka procjene traka je vađenje mjerenja ili otkrivanja koja se mogu koristiti za procjenu oblika ceste ili trake. Te se detekcije generaliziraju pomoću bijelih oznaka ili same granice ceste. Procjena staza je dugi niz godina pod velikim razvojem korištenjem kamere sive skale.

Sustavi temeljeni na pasivnim kamerama mogu se pogoršati pod određenim uvjetima, kao naprimjer dinamičnim promjenama svjetline okoline. Sustavi za izbjegavanje sudara u kolapsu (CICAS) otkrivaju informacije putem senzora temeljenih na vozilu, koji se mogu kombinirati za stvaranje boljeg znanja u stvarnom vremenu o dinamičkoj karti stanja raskrižja. Te informacije mogu se izračunati obavještajnim sustavom u svakom automobilu i mogu upozoriti vozače na nadolazeće opasnosti.

Prepoznavanje prometnih znakova prikaz je na instrumentalnoj ploči koji podsjeća vozače na trenutačno ograničenje brzine. To se postiže korištenjem istog sustava kamere koji također može prepoznati znakove ograničenja brzine. Navigacijski sustavi također podržavaju ovo rješenje pohranjivanjem podataka o ograničenju brzine na neprilagođenoj cesti. Upozorenje krivulje i prilagodljivi tempomat primjeri su ADAS aplikacije koja koriste te podatke.

Adaptivna prilagodba tempomatom može utvrditi da li je vozilo sa gusjenicama privremeno izgubljeno zbog nadolazeće krivulje. Kada to učini, ACC tada može održavati brzinu vozila i odgovarajuću sljedeću udaljenost. Ovaj sustav prati protok prometa koji je ispred vozila, čak i ako je njegov napredak naprijed samo zaustavljanje i kretanje, što je vrlo korisno u zastojsima prometa.

Detekcija mrtvog kuta detekcije (BSD) pomaže vozaču kad izađe kako bi ga pretekao. Senzori nadziru područje na cesti iza i pored vozila i upozoravaju vozača ako se pokuša izvući kad nema mjesta. Sustav je posebice koristan u gustom prometu na više tračnim autocestama ili običnim autocestama, kao i u gradskom prometu. Obavijest o stražnjem križnom prometu (RCTA) može omogućiti nesreće prilikom skretanja sa parkirališta koje često uključuju ozbiljne ozljede.

Rješenje se temelji na dva radarska senzora kratkog dometa koji prate kut od 120 stupnjeva. Kad ADAS otkrije neposredan sudar, upozoriti će vozača zvukom i LED lampama koje će se upaliti u unutarnjem retrovizoru. ADAS također može automatski kočiti vozilo kad se oglasi upozorenje. Ovaj sustav se može koristiti za pouzdan izračun putanje sudara i brzine prijelaza vozila i koliko je udaljen, što ujedno omogućava poboljšanje ADAS-a ovom funkcijom upozorenja.

Pomoć pri kočenju u nuždi (EBA) osigurava optimalno kočenje otkrivanjem kritičnih prometnih situacija. Kad EBA otkrije prednji sudar, sustav za kočenje se uključuje u stanje pripravnosti. Zatim upozorava vozača i počinje lagano kočenje kako bi se spasila važna distanca zaustavljanja. Gradska verzija EBA mogla bi spriječiti mnogo sudara u stražnjem djelu automobila prilikom malih brzina. Posebno je optimiziran za uporabu u gradskom prometu. Inteligentna kontrola prednjih svjetla (IHC) pruža optimalan noćni vid. Prednja svjetla su postavljena tako da omogućuju optimalno osvjetljenje stalnom izmjenom visokih i niskih snopa svjetla. Drugi važan parametar je brzina prikupljanja i obrada podataka. Pretpostavlja se da kod vožnje brzinom od 90 km/h prostorna razlučivost detekcije nije veća od 1 m. Stoga radarska, vizualna ili infracrvena kamera mora učitati i obraditi najmanje 25 sličica u sekundi.



Uz već spomenuto, nove aplikacije se kontinuirano razvijaju i optimiziraju kako bi se povećala sigurnost putnika i pješaka ili životinja te ujedno omogućila udobnija i ekonomičnija vožnja.

## **4.1. Potreban hardver za ADAS**

Napredni sustavi za pomoć vozaču potrebni su zbog povećanja prometa te prema nacionalnim i međunarodnim propisima. ADAS-ove aplikacije mogu izbjeći nezgode i sve prateće ozljede ili moguće smrtne slučajeve. ADAS koristi okolne senzore kao što su radar, infracrveno, video ili ultrazvuk za nadgledanje i analizu okruženja [28].

Različite tvrtke kao što su Bosch, Continental, Delphi Automotive, Freescale, Texas Instruments i mnogi drugi dobavljači pružaju korisnicima različite vrste ADAS rješenja za aplikacije.

### **4.1.1. Senzor kratkog dometa – Lidar (SLR)**

Senzor kratkog dometa Lidar je infracrveni laserski senzor koji djeluje na principu metode vremena prolaska impulsa. Udaljenost od objekta koji se reflektira (točnost  $\pm 0.1 \text{ m} \pm 10\%$ ) može se odrediti bez dodira.

SLR senzor koristi tri neovisna kanala, a osim udaljenosti može odrediti i brzinu (točnost  $\pm 2 \text{ km/h} \pm 10\%$ ) više objekata. Doseže se vidno polje od 27 stupnja u vodoravnom i 11 stupnja u okomitom smjeru. Senzor SLR koristi se za raspon mjerenja od 1m do 13,5m. U slučaju da je udaljenost premala, vozača se upozorava na mogući sudar.

Pored toga senzor se koristi za otkrivanje objekata na područjima koja nisu vidljiva.

### **4.1.2. SRR senzor**

SRR senzor se temelji na modulaciji kompresije impulsa. Senzor mjeri udaljenost do objekta do 50m, relativnu brzinu i kutni odnos dvaju ili više objekata istovremeno u stvarnom vremenu. SRR obično radi u 24 GHz. Senzor može doći do vidnog polja do  $\pm 75$  stupnjeva. Koristi se za otkrivanje slijepih mrlja (BSD) i nadzire područje iza i pored vozila. Funkcija BSD-a je upozoriti

vozača kada se na mrtvim kutovima nalaze premeti vizualnim ili zvučnim signalom kako bi se izbjegle potencijalne nezgode, posebno tijekom manevara promjene staze u gustom prometu.

Drugi senzor koji se koristi je pomoćnik pri promjeni kontrole trake (LCA) koji nadgleda ulične trake i vozaču daje vizualno ili zvučno upozorenje ako pokuša napraviti promjenu trake.

SRR senzor koristi se i za upozorenje o stražnjem križnom prometu (RCTA) koje je dizajnirano tako da izbjegne nezgode prilikom skretanja sa parkirališta.

### **4.1.3. Višenamjenska kamera**

Više namjenska kamera dostupna je u mono ili stereo verziji. Kamera može izmjeriti udaljenost do objekta ispred vozila u rasponu između 20 – 30 m. Nadalje kamera može otkriti pomiče li se i gdje je neki objekt.

Zbog ovakvih značajki stereo kamera se može koristiti za različite napredne funkcije pomoći u prometnom znaku (TSA).

TSA automatski prepoznaje prometne znakove i stoga vozaču može pružiti važne informacije poput ograničenja brzine. Nadalje ACC i EBA su druge aplikacijske kamere. Za ovu funkciju mjeri se udaljenost do prometa ispred kako bi se izbjegli sudari straga. Pored toga kamera se koristi za LDW koji vozaču pruža osjetno upozorenje poput vibracija na upravljaču kako bi promijenio trake. Kamera se koristi i za IHC koji optimizira promjene između dugih i kratkih svjetla tijekom noćnih vožnji.

Sustav koristi integrirani radar i senzor za fotoaparatus (RACam), koji je ujedno i cjelovit sustav koji omogućuje detekciju mrtvih kutova, pomoć pri mijenjanju trake u prometu i upozorenje o raskrižjima. Kompletan RACam sustav uključuje radarski senzor, senzor vida i spajanje podataka u jedan modul koji omogućuje ACC, LDW, upozorenje o frontalnom sudaru i autonomno kočenje za pješake, vozila, životinje i predmete. RF dio sustava koristi radar kratkog dometa od 76 GHz. Jedan od čimbenika aktivnog sigurnosnog sustava je otkrivanje ljudskog lica.

Iako je nemoguće postići radarima, lako se može provesti vizualnom ili infracrvenom kamerom.

## 5. Rješenje za prikaz okruženja kod ugrađenih sustava

Praktični dio zadatka je da se iz 4 različite kamere dobije jedan videozapis sa što bolje riješenim fotometrijskim izjednačavanjem, uz što kraće vrijeme izvođenja.

Ispitat će se i različiti formati izlaza iz kamera (RGB i YCbCr), te na koji način direktno izračunati kalibracijske koeficijente, za fotometrijsko izjednačavanje, za svaku kameru.

Kalibracijski koeficijenti se iz preklapajućih područja mogu izračunati koristeći neke od mjera pogreške, poput MSE.

Budući da ovaj problem mora imati i ograničenje s donje strane, može se upotrijebiti neka od metoda za rješavanje minimizacije funkcije, uz poznate funkcije ograničavanja, poput KKT metode (Karush–Kuhn–Tucker).

### 5.1. KKT uvjeti

Potrebni uvjeti za ograničeni lokalni optimum nazivaju se Karush Kuhn Tucker (KKT), a ti uvjeti imaju vrlo važnu ulogu u ograničenoj teoriji optimizacije i razvoja algoritma. U matematičkoj optimizaciji, KKT uvjeti su prvi derivatni testovi (koji se ponekad nazivaju i nužni uvjeti prvog reda), kako bi rješenje u nelinearnom programiranju bilo optimalno, pod uvjetom da su neki uvjeti pravilnosti zadovoljeni.

Dopuštajući ograničenja nejednakosti, pristup KKT nelinearnom programiranju generalizira metodu Lagrangeovih množitelja, koja omogućuje samo ograničenja jednakosti.

Slično kao i Lagrangeov pristup, problem ograničenog maksimiziranja (minimiziranja) prepisuje se kao Lagrangeova funkcija čija je optimalna točka sedla, tj. globalni maksimum (minimum) nad domenom izbornih varijabli i globalni minimum (maksimum) preko multiplikatora, zbog čega se KKT ponekad naziva i teoremom sedla.

Uvjeti KKT-a izvorno su imenovani po Haroldu W. Kuhnu i Albertu W. Tuckeru, koji su prvi puta objavili uvjete 1951. godine. Kasnije znanstvenici su otkrili da je potrebne uvjete za ovaj problem izjavio Wiliam Karush u svom magistarskom radu 1939. godine.

Za probleme optimizacije koristimo

$$\begin{aligned} & \min_x f(x) \\ & \text{Podložan } g_i(x) - b_i \geq 0 \quad i = 1, \dots, k \\ & g_i(x) - b_i = 0 \quad i = k + 1, \dots, m \end{aligned} \quad (3.10)$$

Postoje četiri uvjeta KKT za optimalne primarne ( $x$ ) i dualne ( $\Lambda$ ) varijable. Zvezdica (\*) označava optimalne vrijednosti.

### 5.1.1. Uvjet izvedivosti

$$g_i(x^*) - b_i, \text{ je izvedivo} \quad (3.11)$$

### 5.1.2. Stanje gradijenta

$$\nabla f(x^*) - \sum_{i=1}^m \Lambda_i^* \nabla g_i(x^*) = 0 \quad (3.12)$$

### 5.1.3. Komplementarna slabost

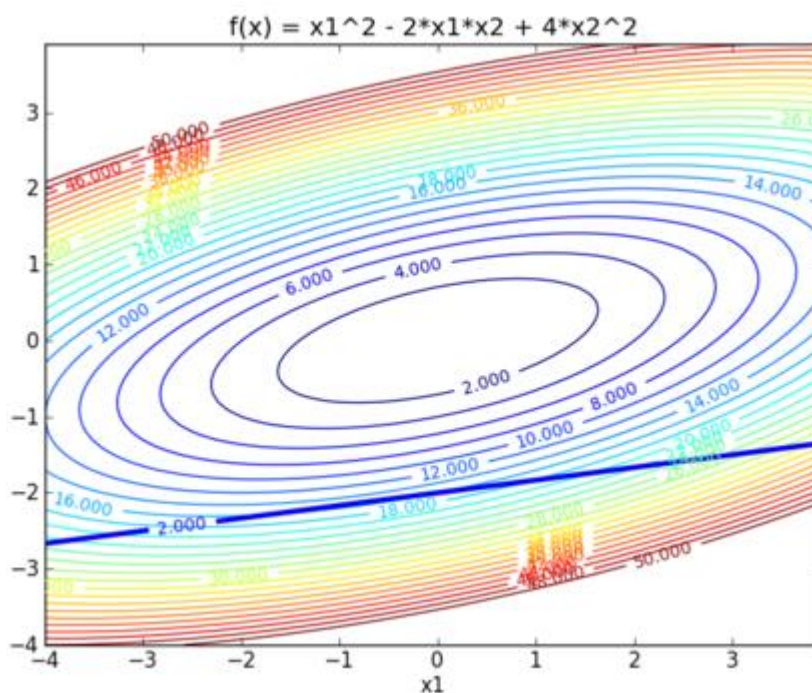
$$\Lambda_i^* (g_i(x^*) - b_i) = 0 \quad (3.13)$$

### 5.1.4. Pozitivni Lagrange množitelji

$$\Lambda_i^* \geq 0 \quad (3.14)$$

Uvjet izvedivosti (1) primjenjuje se i na ograničenja jednakosti i nejednakosti i jednostavno je pravilo da se ograničenja ne smiju kršiti u optimalnim uvjetima. Stanje gradijenta (2)

osigurava da ne postoji izvediv smjer koji bi mogao potencijalno poboljšati funkciju cilja. Zadnja dva uvjeta, (3) i (4) potrebna su samo s ograničenjima nejednakosti i proizvode pozitivni Langrange množitelj kada je ograničenje aktivno ( $= 0$ ) i nulti Langrange množitelj kada je ograničenje neaktivno ( $> 0$ ) [34].



Slika 5.1 KKT uvjeti.

Što se tiče našeg samog programa, ispitali smo tri slučaja:

1. Izjednačavanje po RGB gdje Matlab na početku pretvara originalni YCbCr 4:2:2 u RGB format,
2. Izjednačavanje samo po Y gdje Matlab učitava originalni YCbCr 4:2:2 i ne radi konverziju u RGB,
3. Izjednačavanje unutar YCbCr prostora pomoću RGB interno izračunatih koeficijenata gdje Matlab učitava originalni YCbCr 4:2:2 i ne radi konverziju u RGB.

Nakon početnog koraka moramo definirati izjednačavanje po RGB i Y komponenti. To radimo na način da sve R matrice zamijenimo sa Y. Kada taj korak smo napravili, treba razmisliti kako uključiti Cb i Cr matrice u program jer s njima trebamo imati različite veličine pristupa s obzirom da su dva puta manje po širini u odnosu na Y.

Kada smo definirali početne stavke, preostaje nam definirati sljedeće kamere, kutove, blendu, pozicije, ujednačavanje i finalni prikaz.:

- Cb, Cr kamera
- Generirati ROI za Y, lijeva kamera
- Generirati ROI za C, lijeva kamera
- Generirati ROI za C, desna kamera
- Gornji lijevi kut za blendu – matrica, c
- Gornji desni kut za blendu – matrica, c
- Donji desni kut za blendu – matrica, c
- Donji lijevi kut za blendu – matrica, c
- Središnja pozicija u finalnoj slici, c
- Ujednačavanje po Y, koje može biti samo po Y
- Složiti finalni prikaz videa
- Blendanje po-Y, Cb i Cr
- Blendanje gornjeg lijevog kuta
- Blendanje gornjeg desnog kuta
- Blendanje donjeg lijevog kuta
- Blendanje donjeg desnog kuta
- Spremiti na kraju izlazni video

#### Definirane vrijednosti kamera

```
%Y kamera
y_front_camera=[173 1; 173 640; 415 640; 415 1];
y_rear_camera=[762 1; 762 640; 1004 640; 1004 1];
y_left_camera=[174 963; 174 1216; 1005 1216; 1005 963];
y_left_camera_corner=[174 1131; 352 1216];
y_right_camera=[174 1347; 174 1600 ; 1005 1600 ; 1005 1347];
y_right_camera_corner=[174 1378; 294 1347];

%Cb, Cr kamera
c_front_camera=[173 1; 173 320; 415 320; 415 1];
c_rear_camera=[762 1; 762 320; 1004 320; 1004 1];
c_left_camera=[174 482; 174 608; 1005 608; 1005 482];
c_left_camera_corner=[174 566; 352 608];
c_right_camera=[174 674; 174 800 ; 1005 800 ; 1005 674];
c_right_camera_corner=[174 689; 294 674];
```

## Definiranje dimenzija kamera

```
Y_front=Y(y_front_camera(1,1):y_front_camera(3,1),y_front_camera(1,2)+1:2:y_f  
ront_camera(3,2));  
Cb_front=Cb(c_front_camera(1,1):c_front_camera(3,1),c_front_camera(1,2):c_fro  
nt_camera(3,2));  
Cr_front=Cr(c_front_camera(1,1):c_front_camera(3,1),c_front_camera(1,2):c_fro  
nt_camera(3,2));
```

```
Y_left=Y(y_left_camera(1,1):y_left_camera(3,1),y_left_camera(1,2)+1:2:y_left_  
camera(3,2));  
Cb_left=Cb(c_left_camera(1,1):c_left_camera(3,1),c_left_camera(1,2):c_left_ca  
mera(3,2));  
Cr_left=Cr(c_left_camera(1,1):c_left_camera(3,1),c_left_camera(1,2):c_left_ca  
mera(3,2));
```

```
Y_right=Y(y_right_camera(1,1):y_right_camera(3,1),y_right_camera(1,2)+1:2:y_r  
ight_camera(3,2));  
Cb_right=Cb(c_right_camera(1,1):c_right_camera(3,1),c_right_camera(1,2):c_rig  
ht_camera(3,2));  
Cr_right=Cr(c_right_camera(1,1):c_right_camera(3,1),c_right_camera(1,2):c_rig  
ht_camera(3,2));
```

```
Y_rear=Y(y_rear_camera(1,1):y_rear_camera(3,1),y_rear_camera(1,2)+1:2:y_rear_  
camera(3,2));  
Cb_rear=Cb(c_rear_camera(1,1):c_rear_camera(3,1),c_rear_camera(1,2):c_rear_ca  
mera(3,2));  
Cr_rear=Cr(c_rear_camera(1,1):c_rear_camera(3,1),c_rear_camera(1,2):c_rear_ca  
mera(3,2));
```

## Definiranje pretvorbe YCbCr u RGB i RGB u YCbCr

```
%%konverzije za RGB -> YCbCr i YCbCr ->RGB  
%u kodu se pretpostavljaju granice  
%Y...16/255-235/255, Cb/Cr...16/255-240/255, R/G/B...0-255/255  
  
%[y;Cb;Cr]=[16/255;128/255;128/255]+[0.257 0.504 0.098;-0.148 -0.291  
0.439;0.439 -0.368 -0.071]*[R;G;B];  
%and  
%[R;G;B]=[1.164 0 1.596; 1.164 -0.392 -0.813; 1.164 2.017 0]*[Y-  
16/255;Cb-128/255;Cr-128/255];  
  
rgb_ycbcr=[0.257 0.504 0.098;-0.148 -0.291 0.439;0.439 -0.368 -0.071];  
rgb_ycbcr_const=[16/255;128/255;128/255];  
  
ycbcr_rgb=[1.164 0 1.596; 1.164 -0.392 -0.813; 1.164 2.017 0];  
ycbcr_rgb_const=[-16/255;-128/255;-128/255];
```

## Blendanje

```
s1y=(Y_left(1:size(Y_front,1),:)).*upper_left_blend_c_inner;  
s2y=(Y_front(:,1:size(Y_left,2))).*upper_left_blend_c_inner;  
s3y=(Y_front(:,end-size(Y_right,2)+1:end)).*upper_right_blend_c_inner;  
s4y=(Y_right(1:size(Y_front,1),:)).*upper_right_blend_c_inner;  
s5y=(Y_right(end-size(Y_rear,1)+1:end,:)).*lower_right_blend_c_inner;  
s6y=(Y_rear(:,end-size(Y_right,2)+1:end)).*lower_right_blend_c_inner;  
s7y=(Y_rear(:,1:size(Y_left,2))).*lower_left_blend_c_inner;  
s8y=(Y_left(end-size(Y_rear,1)+1:end,:)).*lower_left_blend_c_inner;
```

```
[a1y,a2y,a3y,a4y]=optimisation6b_bez_u(s1y,s2y,s3y,s4y,s5y,s6y,s7y,s8y,sum(up  
per_left_blend_y_inner(:)),sum(upper_right_blend_y_inner(:)),sum(lower_right_  
blend_y_inner(:)),sum(lower_left_blend_y_inner(:)));
```

```
s1cb=(Cb_left(1:size(Cb_front,1),:)).*upper_left_blend_c_inner;  
s2cb=(Cb_front(:,1:size(Cb_left,2))).*upper_left_blend_c_inner;  
s3cb=(Cb_front(:,end-size(Cb_right,2)+1:end)).*upper_right_blend_c_inner;  
s4cb=(Cb_right(1:size(Cb_front,1),:)).*upper_right_blend_c_inner;  
s5cb=(Cb_right(end-size(Cb_rear,1)+1:end,:)).*lower_right_blend_c_inner;  
s6cb=(Cb_rear(:,end-size(Cb_right,2)+1:end)).*lower_right_blend_c_inner;  
s7cb=(Cb_rear(:,1:size(Cb_left,2))).*lower_left_blend_c_inner;  
s8cb=(Cb_left(end-size(Cb_rear,1)+1:end,:)).*lower_left_blend_c_inner;
```

```
s1cr=(Cr_left(1:size(Cr_front,1),:)).*upper_left_blend_c_inner;  
s2cr=(Cr_front(:,1:size(Cr_left,2))).*upper_left_blend_c_inner;  
s3cr=(Cr_front(:,end-size(Cr_right,2)+1:end)).*upper_right_blend_c_inner;  
s4cr=(Cr_right(1:size(Cr_front,1),:)).*upper_right_blend_c_inner;  
s5cr=(Cr_right(end-size(Cr_rear,1)+1:end,:)).*lower_right_blend_c_inner;  
s6cr=(Cr_rear(:,end-size(Cr_right,2)+1:end)).*lower_right_blend_c_inner;  
s7cr=(Cr_rear(:,1:size(Cr_left,2))).*lower_left_blend_c_inner;  
s8cr=(Cr_left(end-size(Cr_rear,1)+1:end,:)).*lower_left_blend_c_inner;
```

```
s1y=s1y+ycbcr_rgb_const(1)*upper_left_blend_c_inner;  
s2y=s2y+ycbcr_rgb_const(1)*upper_left_blend_c_inner;  
s3y=s3y+ycbcr_rgb_const(1)*upper_right_blend_c_inner;  
s4y=s4y+ycbcr_rgb_const(1)*upper_right_blend_c_inner;  
s5y=s5y+ycbcr_rgb_const(1)*lower_right_blend_c_inner;  
s6y=s6y+ycbcr_rgb_const(1)*lower_right_blend_c_inner;  
s7y=s7y+ycbcr_rgb_const(1)*lower_left_blend_c_inner;  
s8y=s8y+ycbcr_rgb_const(1)*lower_left_blend_c_inner;
```

```
s1cb=s1cb+ycbcr_rgb_const(2)*upper_left_blend_c_inner;  
s2cb=s2cb+ycbcr_rgb_const(2)*upper_left_blend_c_inner;  
s3cb=s3cb+ycbcr_rgb_const(2)*upper_right_blend_c_inner;  
s4cb=s4cb+ycbcr_rgb_const(2)*upper_right_blend_c_inner;  
s5cb=s5cb+ycbcr_rgb_const(2)*lower_right_blend_c_inner;  
s6cb=s6cb+ycbcr_rgb_const(2)*lower_right_blend_c_inner;  
s7cb=s7cb+ycbcr_rgb_const(2)*lower_left_blend_c_inner;
```



## Izjednačavanje kamera

%naprijed

```
Y_front2=rgb_ycbcr_const(1)+...
```

```
rgb_ycbcr(1,1)*a1r*(ycbcr_rgb(1,1)*(Y_front+ycbcr_rgb_const(1))+ycbcr_rgb(1,2)
)*(Cb_front+ycbcr_rgb_const(2))+ycbcr_rgb(1,3)*(Cr_front+ycbcr_rgb_const(3)))
+...
```

```
rgb_ycbcr(1,2)*a1g*(ycbcr_rgb(2,1)*(Y_front+ycbcr_rgb_const(1))+ycbcr_rgb(2,2)
)*(Cb_front+ycbcr_rgb_const(2))+ycbcr_rgb(2,3)*(Cr_front+ycbcr_rgb_const(3)))
+...
```

```
rgb_ycbcr(1,3)*a1b*(ycbcr_rgb(3,1)*(Y_front+ycbcr_rgb_const(1))+ycbcr_rgb(3,2)
)*(Cb_front+ycbcr_rgb_const(2))+ycbcr_rgb(3,3)*(Cr_front+ycbcr_rgb_const(3)))
```

```
;
```

```
Y_front2(Y_front2>235/255)=235/255;
```

```
Y_front2(Y_front2<16/255)=16/255;
```

```
Cb_front2=rgb_ycbcr_const(2)+...
```

```
rgb_ycbcr(2,1)*a1r*(ycbcr_rgb(1,1)*(Y_front+ycbcr_rgb_const(1))+ycbcr_rgb(1,2)
)*(Cb_front+ycbcr_rgb_const(2))+ycbcr_rgb(1,3)*(Cr_front+ycbcr_rgb_const(3)))
+...
```

```
rgb_ycbcr(2,2)*a1g*(ycbcr_rgb(2,1)*(Y_front+ycbcr_rgb_const(1))+ycbcr_rgb(2,2)
)*(Cb_front+ycbcr_rgb_const(2))+ycbcr_rgb(2,3)*(Cr_front+ycbcr_rgb_const(3)))
+...
```

```
rgb_ycbcr(2,3)*a1b*(ycbcr_rgb(3,1)*(Y_front+ycbcr_rgb_const(1))+ycbcr_rgb(3,2)
)*(Cb_front+ycbcr_rgb_const(2))+ycbcr_rgb(3,3)*(Cr_front+ycbcr_rgb_const(3)))
```

```
;
```

```
Cb_front2(Cb_front2>240/255)=240/255;
```

```
Cb_front2(Cb_front2<16/255)=16/255;
```

```
Cr_front2=rgb_ycbcr_const(3)+...
```

```
rgb_ycbcr(3,1)*a1r*(ycbcr_rgb(1,1)*(Y_front+ycbcr_rgb_const(1))+ycbcr_rgb(1,2)
)*(Cb_front+ycbcr_rgb_const(2))+ycbcr_rgb(1,3)*(Cr_front+ycbcr_rgb_const(3)))
+...
```

```
rgb_ycbcr(3,2)*a1g*(ycbcr_rgb(2,1)*(Y_front+ycbcr_rgb_const(1))+ycbcr_rgb(2,2)
)*(Cb_front+ycbcr_rgb_const(2))+ycbcr_rgb(2,3)*(Cr_front+ycbcr_rgb_const(3)))
+...
```

```
rgb_ycbcr(3,3)*a1b*(ycbcr_rgb(3,1)*(Y_front+ycbcr_rgb_const(1))+ycbcr_rgb(3,2)
)*(Cb_front+ycbcr_rgb_const(2))+ycbcr_rgb(3,3)*(Cr_front+ycbcr_rgb_const(3)))
```

```
;
```

```
Cr_front2(Cr_front2>240/255)=240/255;
```

```
Cr_front2(Cr_front2<16/255)=16/255;
```

## Finalni okvir

```
%finalni video frame
Y_final=128/255*ones(c_left_camera(4,1)-
c_left_camera(1,1)+1,c_front_camera(2,2)-c_front_camera(1,2)+1);
Cb_final=128/255*ones(c_left_camera(4,1)-
c_left_camera(1,1)+1,c_front_camera(2,2)-c_front_camera(1,2)+1);
Cr_final=128/255*ones(c_left_camera(4,1)-
c_left_camera(1,1)+1,c_front_camera(2,2)-c_front_camera(1,2)+1);

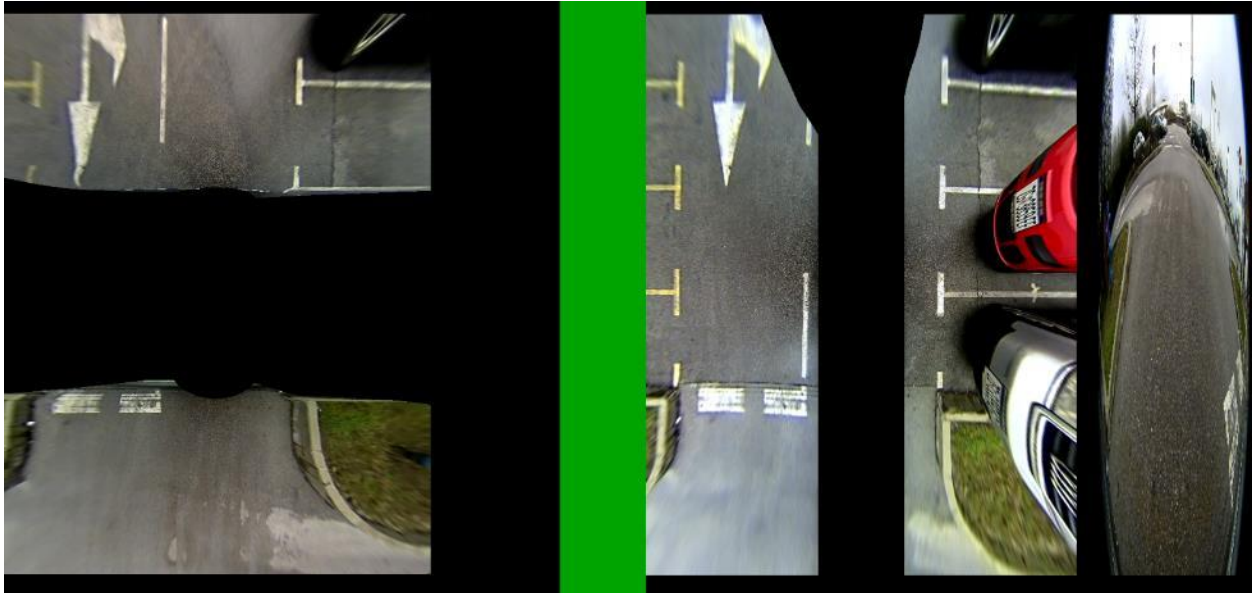
Y_final(1:size(Y_front,1),1:size(Y_front,2))=Y_front2;
Y_final(1:size(Y_left,1),1:size(Y_left,2))= Y_left2;
Y_final(1:size(Y_right,1),end-size(Y_right,2)+1:end)=Y_right2;
Y_final(end-size(Y_rear,1)+1:end,end-size(Y_rear,2)+1:end)=Y_rear2;

Cb_final(1:size(Cb_front,1),1:size(Cb_front,2))=Cb_front2;
Cb_final(1:size(Cb_left,1),1:size(Cb_left,2))= Cb_left2;
Cb_final(1:size(Cb_right,1),end-size(Cb_right,2)+1:end)=Cb_right2;
Cb_final(end-size(Cb_rear,1)+1:end,end-size(Cb_rear,2)+1:end)=Cb_rear2;

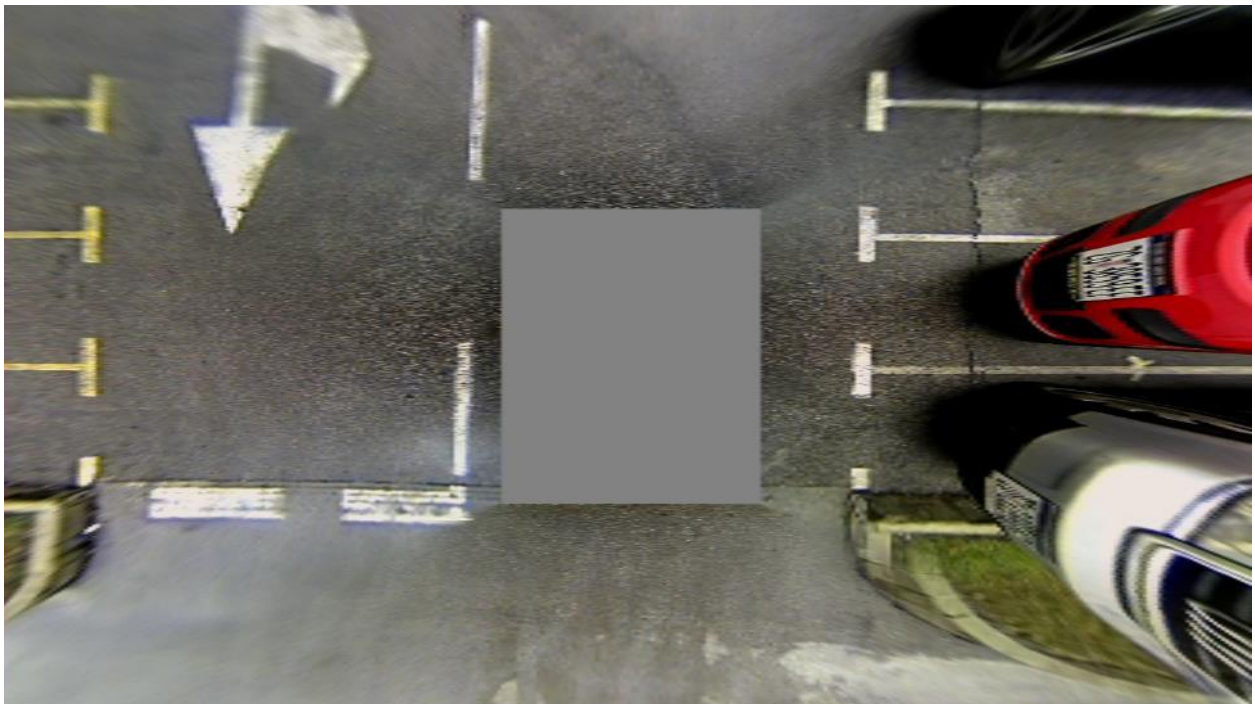
Cr_final(1:size(Cr_front,1),1:size(Cr_front,2))=Cr_front2;
Cr_final(1:size(Cr_left,1),1:size(Cr_left,2))= Cr_left2;
Cr_final(1:size(Cr_right,1),end-size(Cr_right,2)+1:end)=Cr_right2;
Cr_final(end-size(Cr_rear,1)+1:end,end-size(Cr_rear,2)+1:end)=Cr_rear2;
```

Možemo pretpostaviti da je isječak koda poprilično nejasan, no zbog toga u nastavku se mogu vidjeti rezultati na izlaznim slikama.

Nadalje, priložit ću sliku ulazne datoteke, odnosno sliku svih ulaznih kamera ptičjeg oka i finalnu sliku koja se dobije pomoću našeg algoritma. Točnije dobit će se jedna jasna i intuitivna slika.



*Slika 5.2 Ulazna slika, pogled od svih četiri kamera.*



*Slika 5.3 Izlazna slika, dobivena našim algoritmom.*

## 6. Rezultati provedenog testiranja

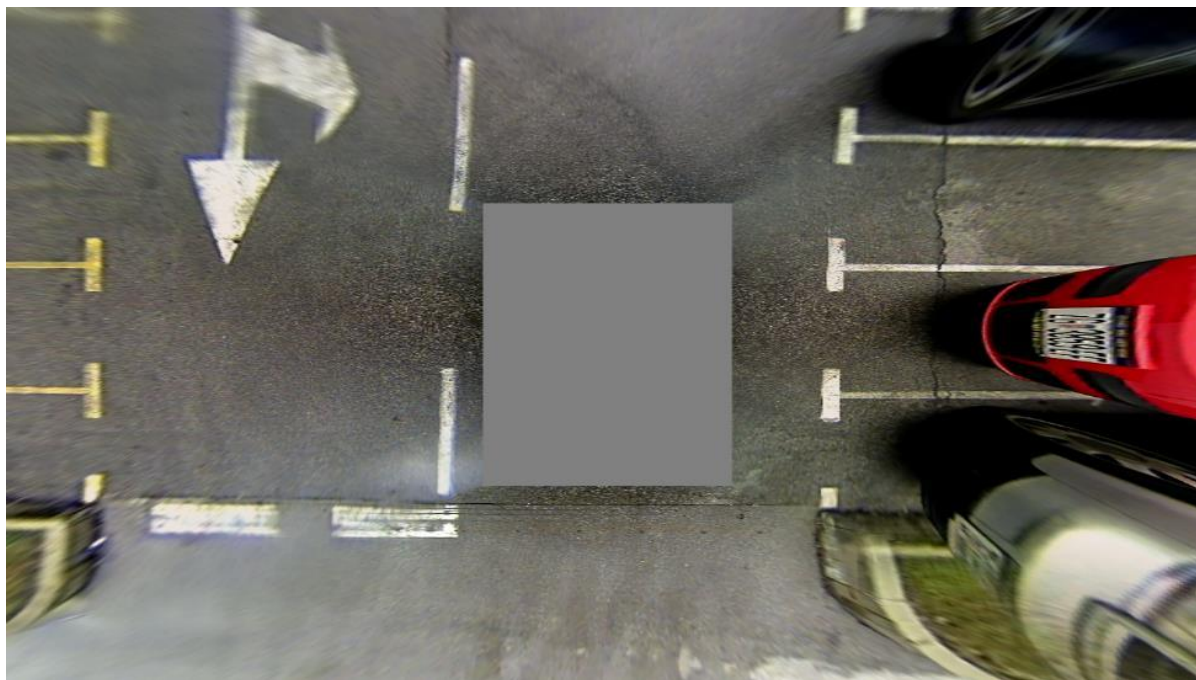
Kako bi zaokružili cjelinu te ujedno i „preveli“ sve ove silne formule, algoritme i programski kod na krajnjem korisniku prihvatljiv jezik, u nastavku ćemo napraviti analizu između RGB i Y odnosno YCbCr komponente.

### 6.1. Izjednačavanje RGB i Y komponente

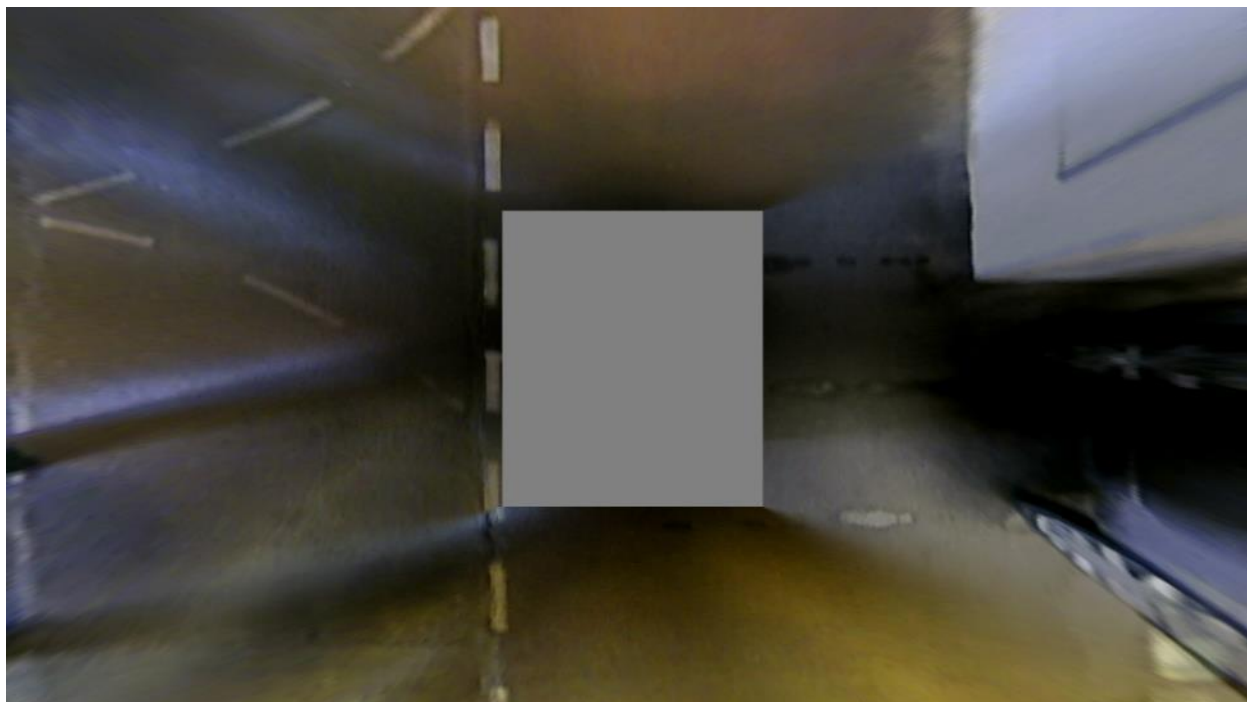
Ovdje ćemo prikazati izjednačavanje videozapisa na RGB i Y komponente. Izvrtjet ćemo nekoliko videozapisa i staviti rezultate dobivenih slika.

#### 6.1.1. RGB komponenta

U nastavku slijede četiri slike po RGB izjednačavanju, te nakon toga kratka usporedba. Prvo smo stavili sve slike kako bi se mogli vidjeti različiti vremenski uvjeti, točnije kako se kamera ponaša u nekoliko vremenskih uvjeta.



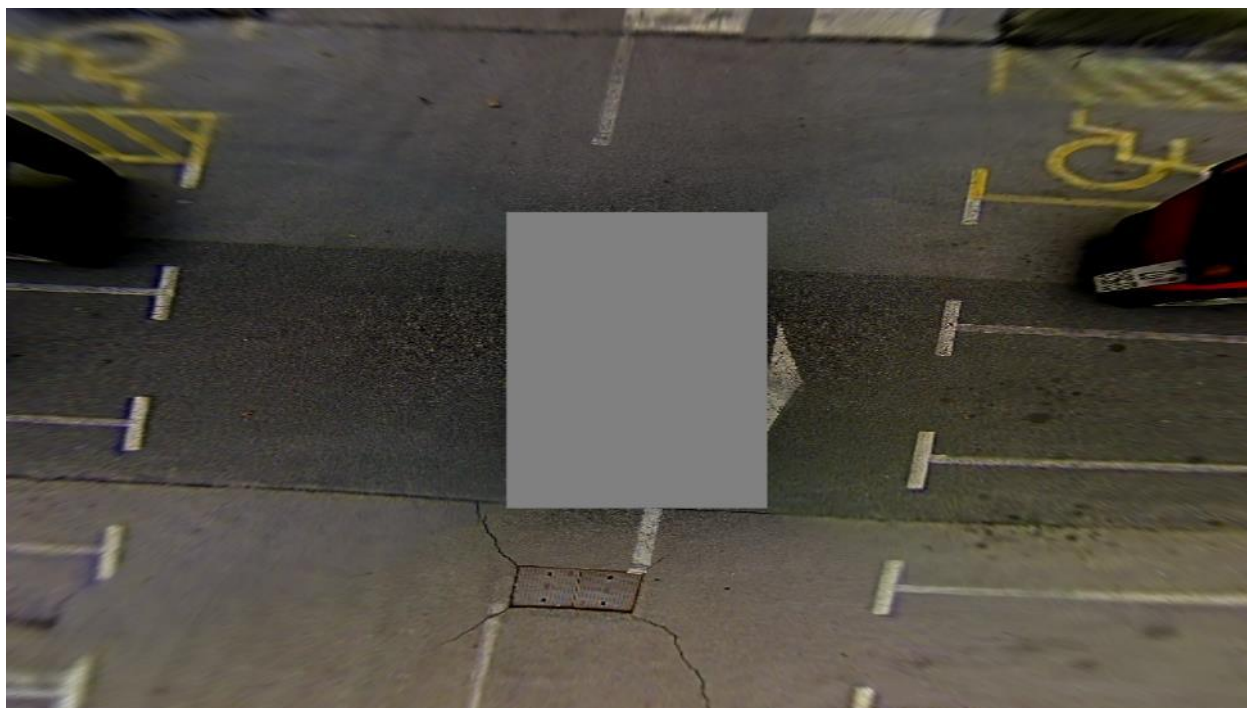
*Slika 6.1 RGB rezultat 1. videozapisa, normalni dnevni uvjeti.*



*Slika 6.2 RGB rezultat 2. videozapisa, noćni uvjeti.*



*Slika 6.3 RGB rezultat 3. videozapisa, jutarnji uvjet.*

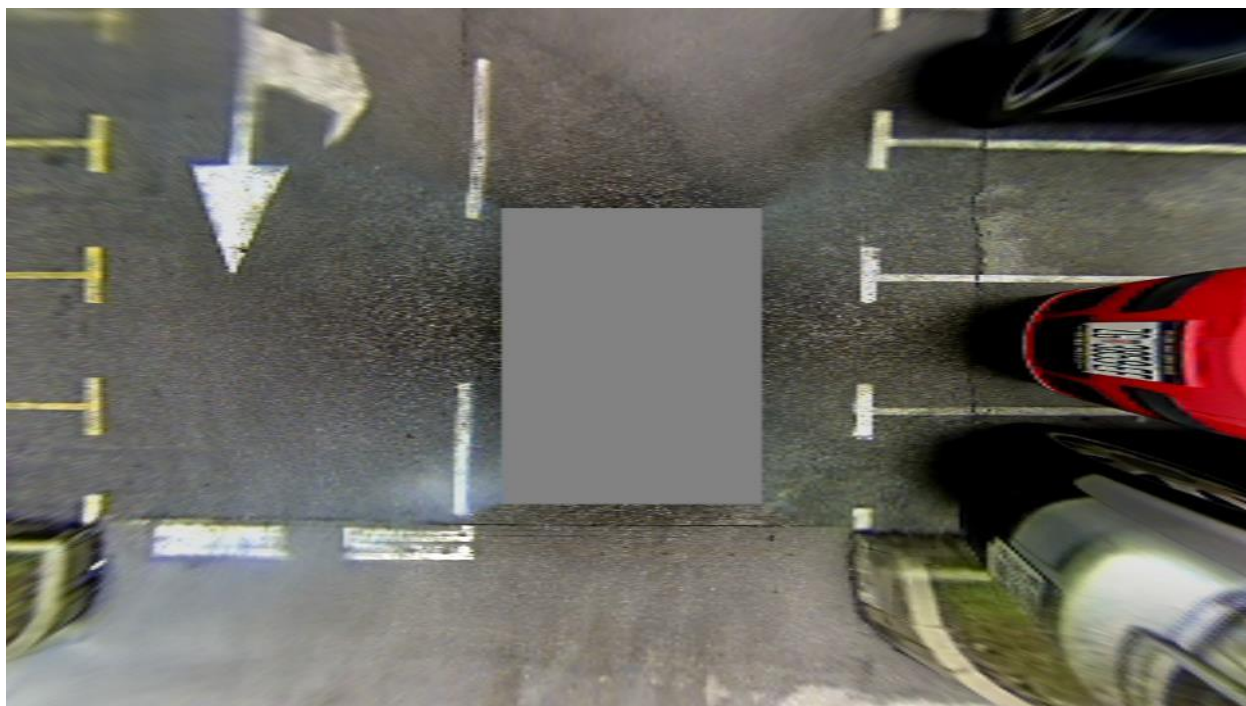


*Slika 6.4 RGB rezultat 4. videozapisa, uvjeti pred zalazak sunca.*

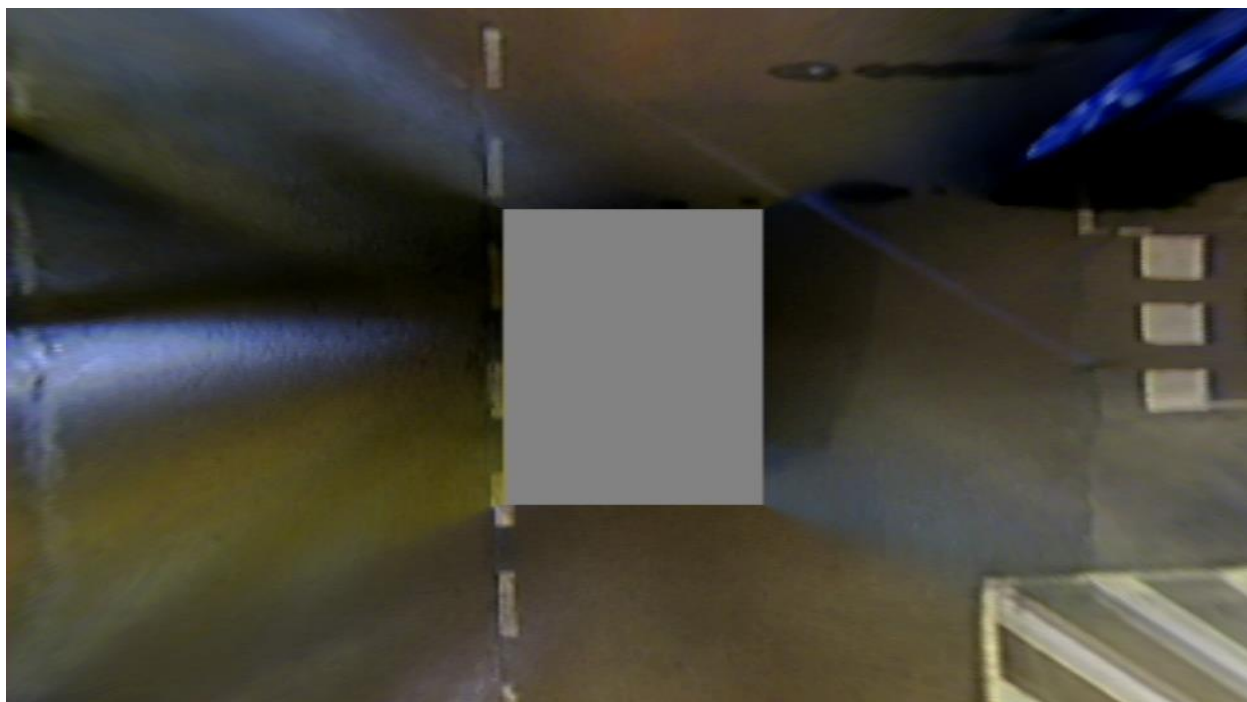
Iz prvog priloženog seta od četiri slike možemo vidjeti što će nam biti početna „baza“, odnosno kako će slike izgledati prema RGB komponenti. Set se sastoji od četiri različita videa, te sukladno tome imamo i četiri slike.

### 6.1.2. Y komponenta

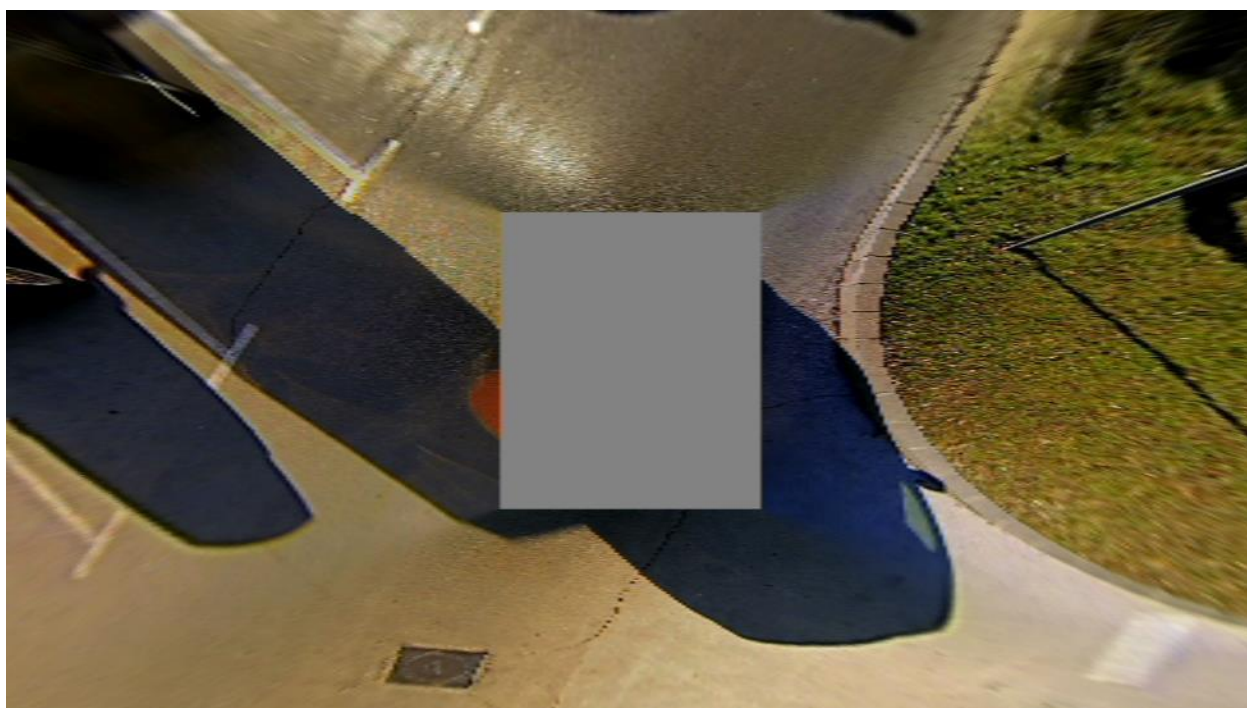
Kao što smo već naveli kod RGB komponente, u nastavku slijede četiri slike po Y izjednačavanju, te nakon toga kratka usporedba. Prvo smo stavili sve slike kako bi se mogli vidjeti različiti vremenski uvjeti, točnije kako se kamera ponaša u nekoliko vremenskih uvjeta.



*Slika 6.5 Y rezultat 1. videozapisa, normalni dnevni uvjeti.*

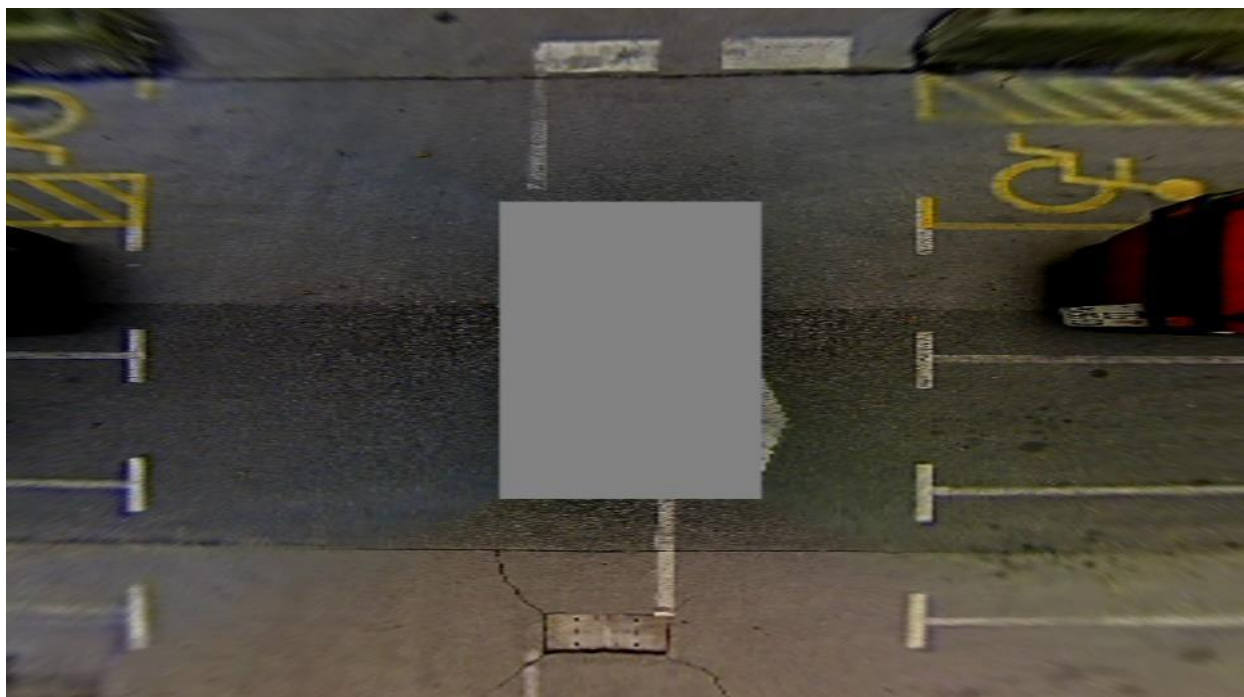


*Slika 6.6 Y rezultat 2. videozapisa noćni uvjeti .*



*Slika 6.7 Y rezultat 3. videozapisa, jutarnji uvjet.*





*Slika 6.8 Y rezultat 4. videozapisa, uvjeti pred zalazak sunca..*

Također kao što smo naveli kod RGB komponente, set se sastoji od četiri različita videa, te sukladno tome imamo i četiri slike. Ovdje se radi o Y komponenti, odnosno Y izjednačavanje i na prvi pogled slike su poprilično slične.

Daljnjom usporedbom između RGB i Y usrednjavanja vidimo da su slike od Y komponente nešto više blijede odnosno sivo plave.

Prema priloženim slikama možemo vidjeti da je RGB komponenta dosta oštrija i kvalitetnija za razliku od Y komponente. Također to se može vidjeti prema samoj veličini datoteke.

Kod RGB-a veličina datoteke je od 2 do 7 GB, dok kod Y komponente je od 1.5 do 2.5 GB.

Uz kvalitetu slike i veličine, također se treba uzeti u obzir renderiranja koje je u Y komponenti znatno manje i brže nego kod RGB-a. U RGB komponenti vrijeme renderiranja je od 3 minute pa sve do 5 minuta, dok kod Y komponente je u prosjeku oko 3 minute za sva četiri videa. Također treba imati na umu da ovdje spominjemo vrijeme renderiranja samo po Y komponenti, tako da Cb i Cr komponente dodatno troše vrijeme renderiranja koje smo u ovom slučaju isključili.

Naravno, treba uzeti u obzir da se radi o drugačijim veličinama ulaznih datoteka kao i njihovim trajanjem.

Sve u svemu možemo zaključiti da je RGB bolji jer je slika jasnija, boje su izjednačene dok je kod Y komponente samo svjetlina izjednačena.

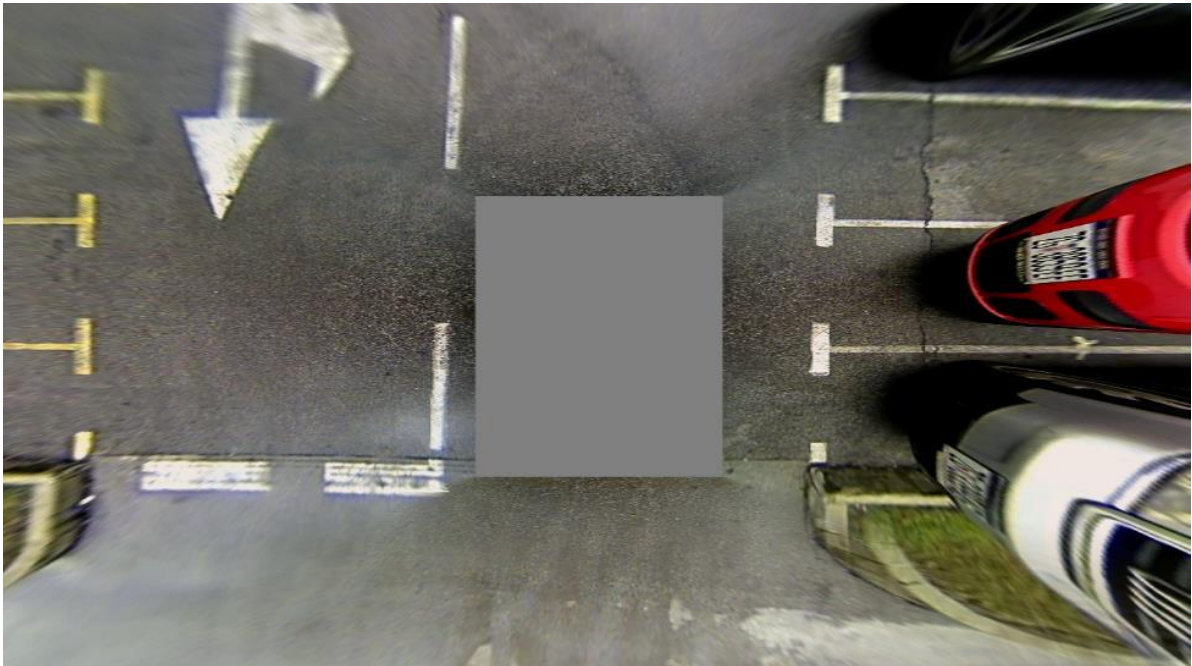
## 6.2. Rezultati različitih granica KKT-a

Ovdje ćemo provjeriti rezultate za različite granice KKT optimizacije. Donja granica (konst1 = 1) će ostati fiksna, dok ćemo gornju granicu (konst2 = 3, 4, 5) povećavati za jedan.

### 6.2.1. RGB komponenta

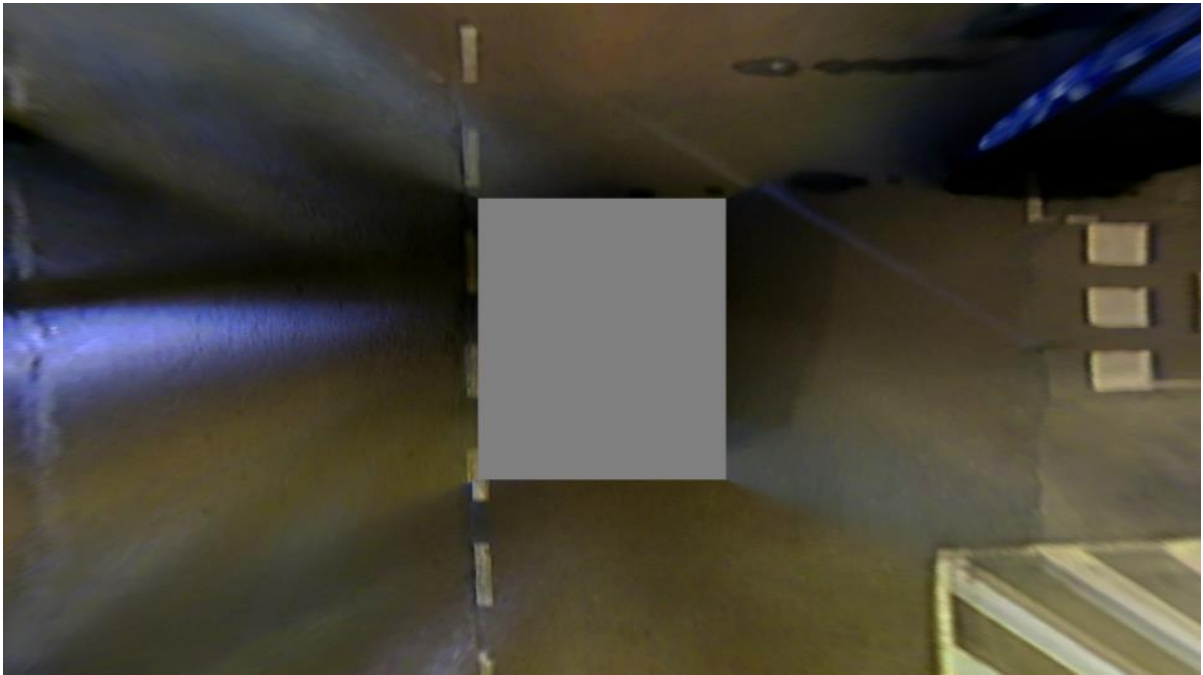
Za prvi slučaj ćemo koristiti sljedeće postavke:

- Konst1 = 1,
- Konst2 = 3.



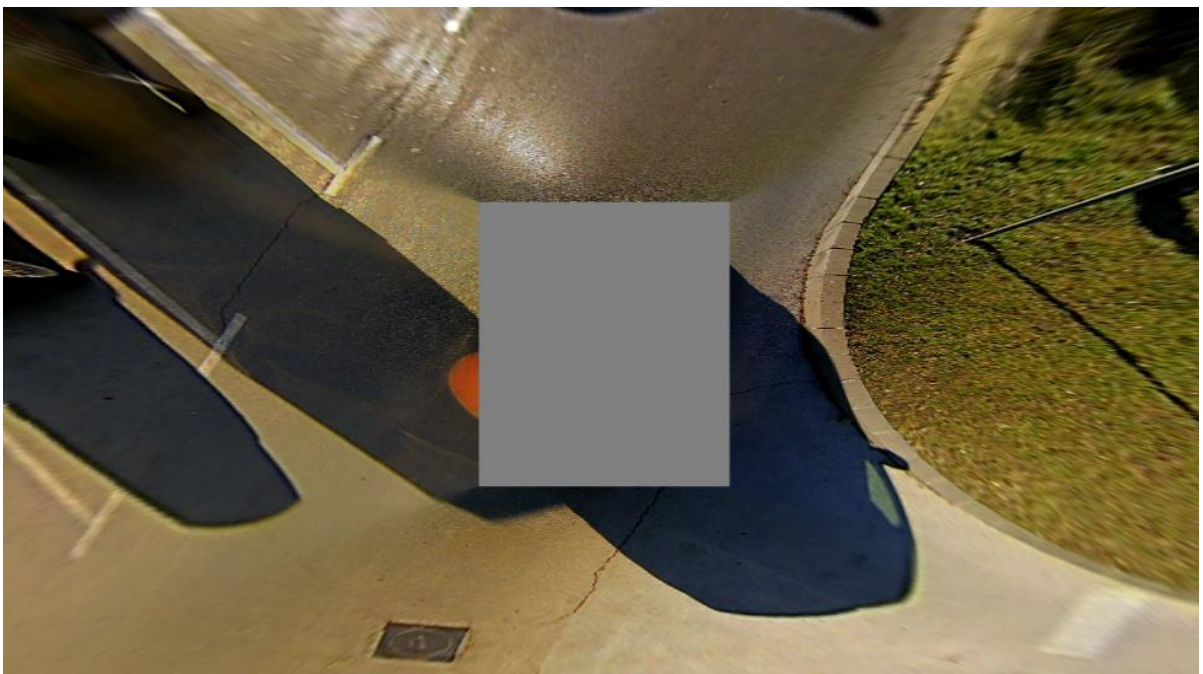
*Slika 6.9 RGB rezultat 1. videozapisa.*

Prva slika nam je svjetlija, jasnija i uočljivija



*Slika 6.10 RGB rezultat 2. videozapisa.*

Isti slučaj kao i kod prve slike, slika je jasnija po noćnim uvjetima.



*Slika 6.11 RGB rezultat 3. videozapisa.*

Kod treće slike možemo vidjeti da je dosta zasićenija i „življa“ nego u početnim postavkama.

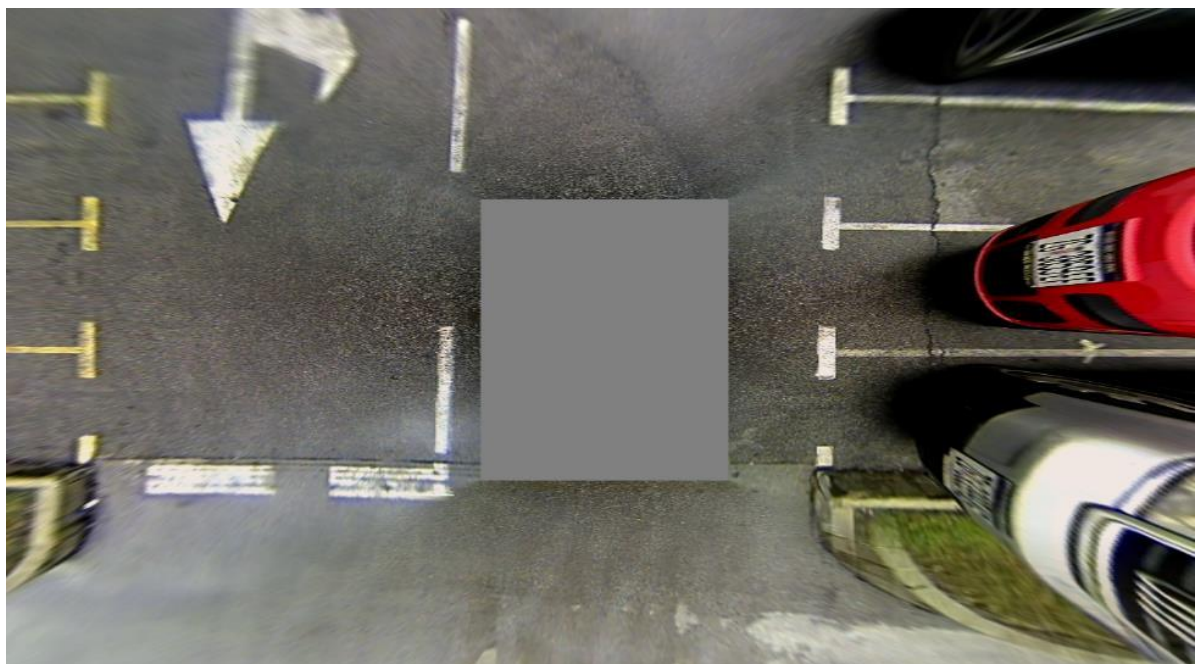


*Slika 6.12 RGB rezultat 4. videozapisa.*

Kod zadnje slike u ovom setu primjećujemo izuzetno čistu sliku u odnosu na ostale tri.

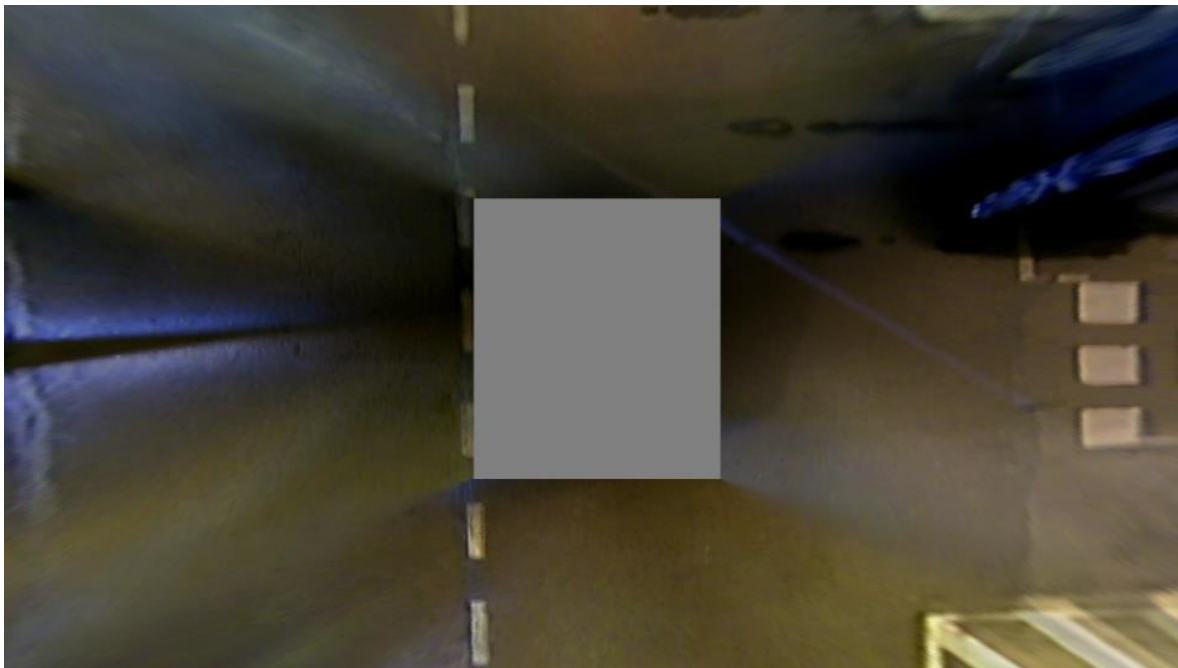
Za drugi slučaj ćemo koristiti sljedeće postavke:

- Konst1 = 1,
- Konst2 = 4.



*Slika 6.13 RGB rezultat 1. videozapisa.*

Sa povećanjem gornje granice vidimo da slika počinje gubiti na čistoći i već se može primijetiti izbijanje sive boje oko sredine slike.



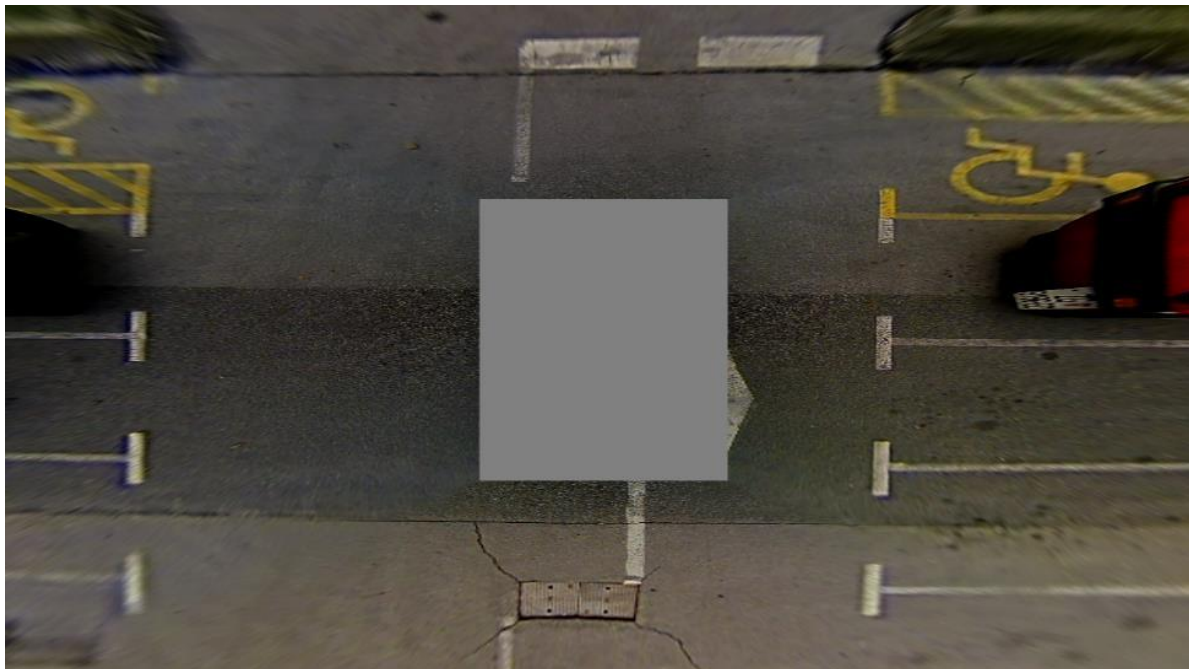
*Slika 6.14 RGB rezultat 2. videozapisa.*

Noćni uvjeti nisu nešto previše izgubili povećanjem gornje granice, slika je i dalje jasna a razlike u odnosu na prvi set gotovo da i nema.



*Slika 6.15 RGB rezultat 3. videozapisa.*

Kod jutarnjih uvjeta možemo vidjeti da slika više nije čista i zasićena kao i u prvom setu slika, već dolazi do mutnoće. To najbolje vidimo na području trave koja više nije oštra. Naravno to možemo i prepisati kameri u pokretu.

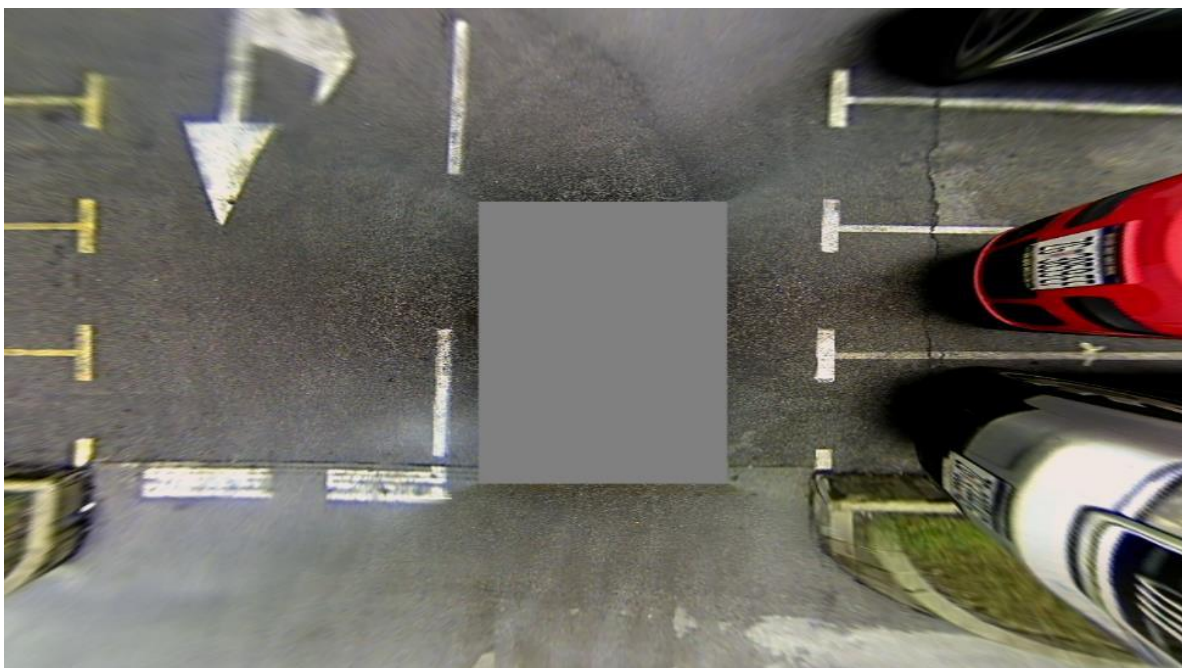


*Slika 6.16 RGB rezultat 4. videozapisa.*

Zadnja slika ovog seta nam je i dalje čista, razlike su minimalne (oštrina).

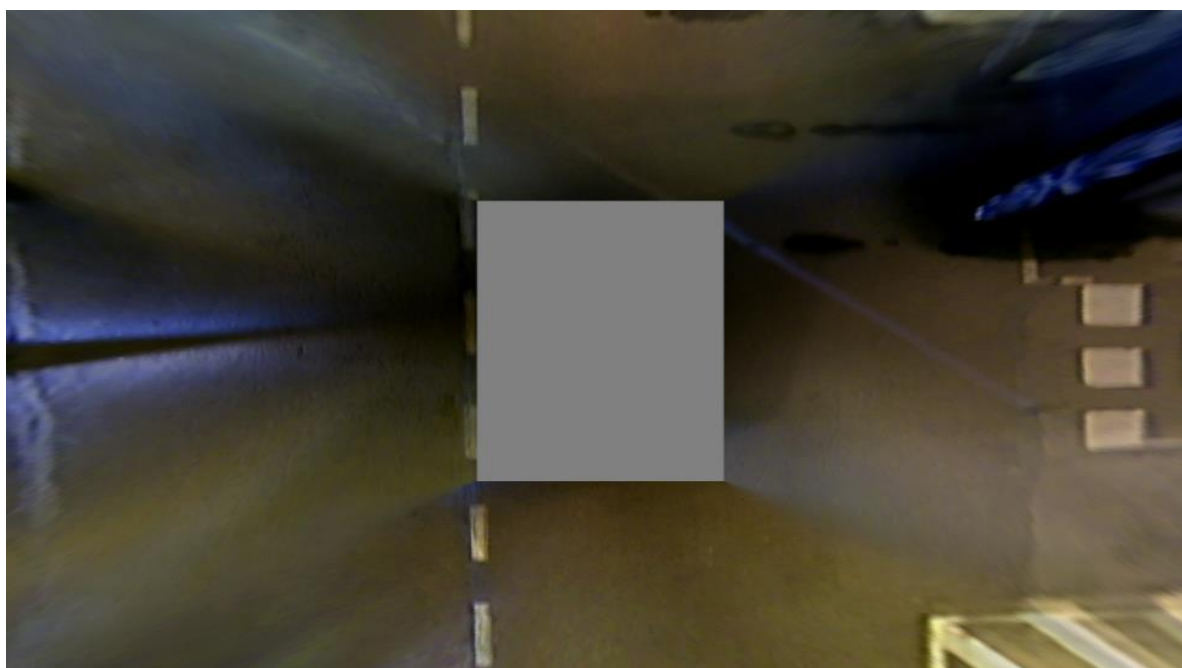
Za treći slučaj ćemo koristiti sljedeće postavke:

- Konst1 = 1,
- Konst2 = 5.



*Slika 6.17 RGB rezultat 1.videzapisa.*

Ovdje već vidimo izbijanje bijele boje.



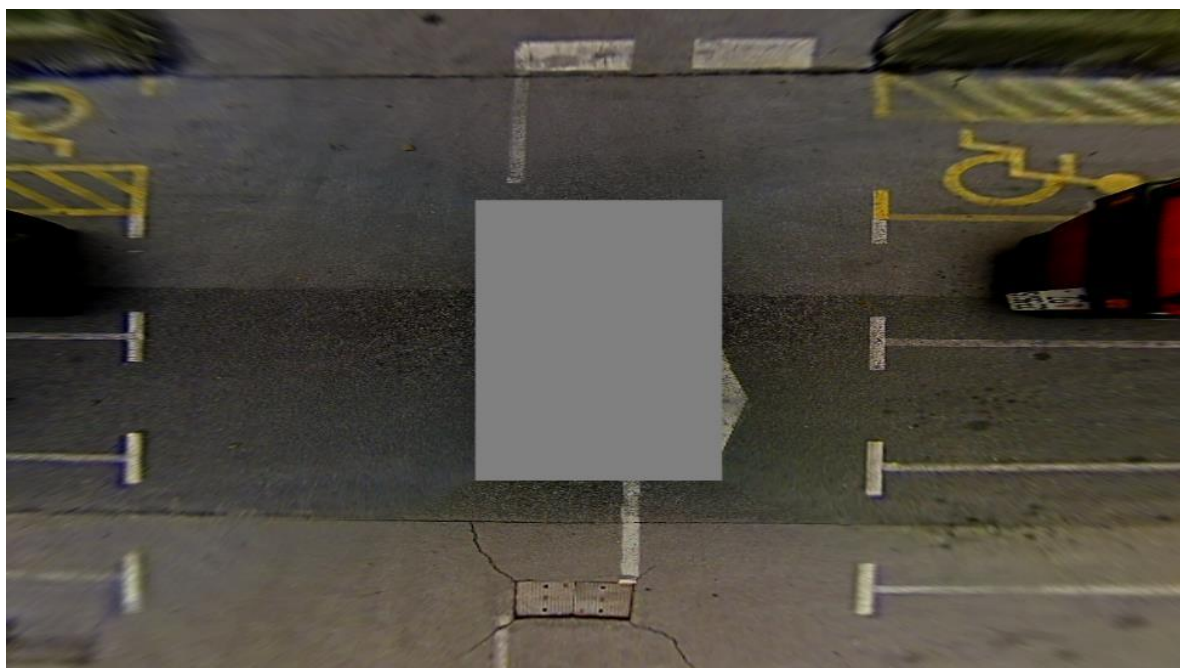
*Slika 6.18 RGB rezultat 2. videozapisa.*

U noćnim uvjetima vidimo da slika postaje nešto mutnija.



*Slika 6.19 RGB rezultat 3. videozapisa.*

Kod rano jutarnjih uvjeta vidimo da slika i dalje nema značajnijih promjena.



*Slika 6.20 RGB rezultat 4. videozapisa.*

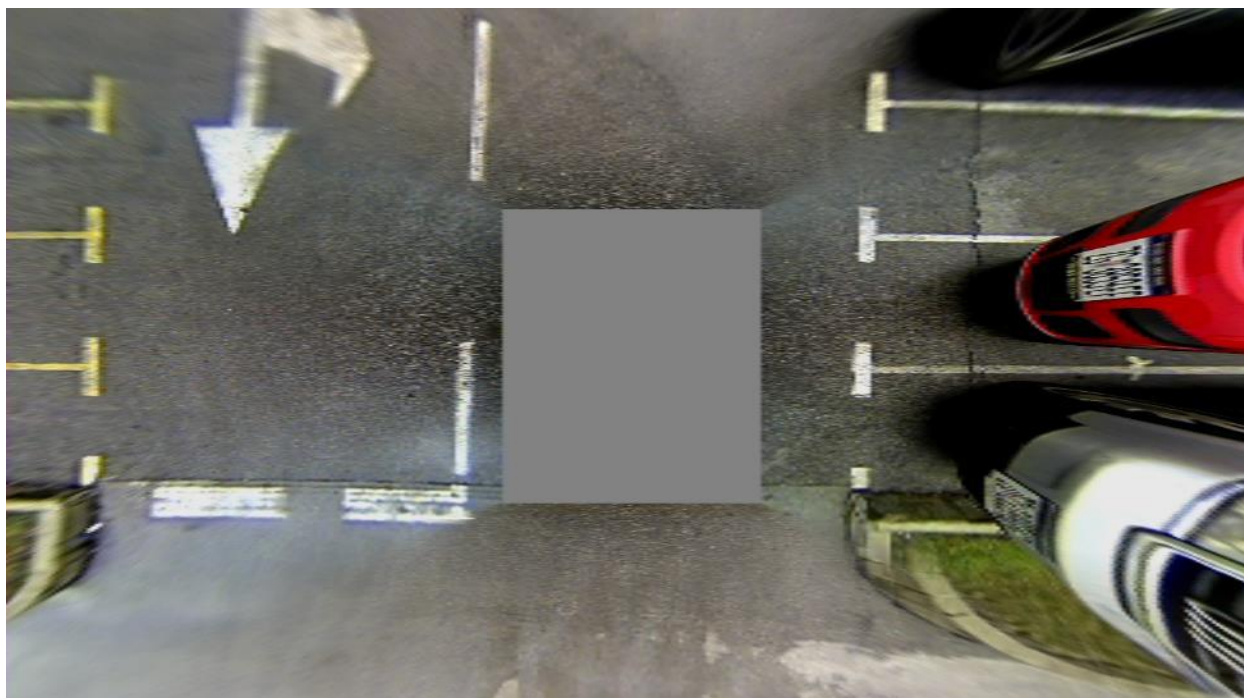
Isti slučaj kao i kod rano jutarnjih uvjeta, ne vide se značajnije promjene.



### 6.2.2. Y komponenta

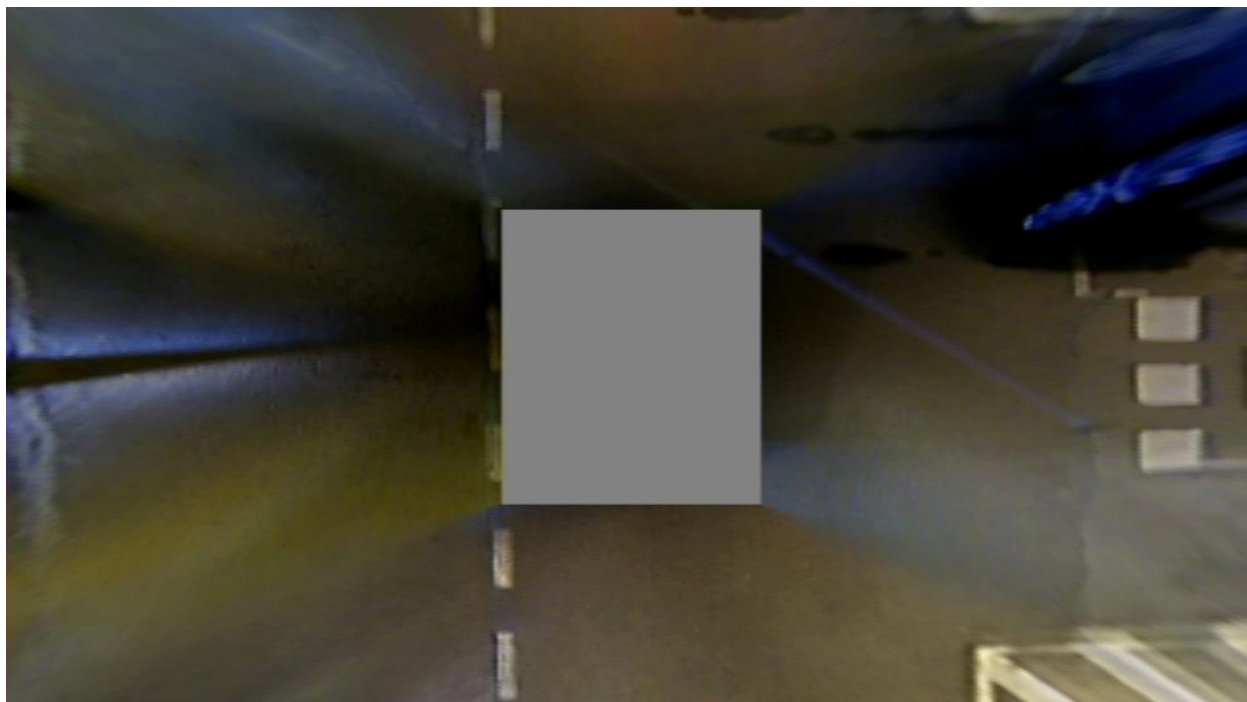
Za prvi slučaj ćemo koristiti sljedeće postavke:

- $Konst1 = 1$ ,
- $Konst2 = 3$ .



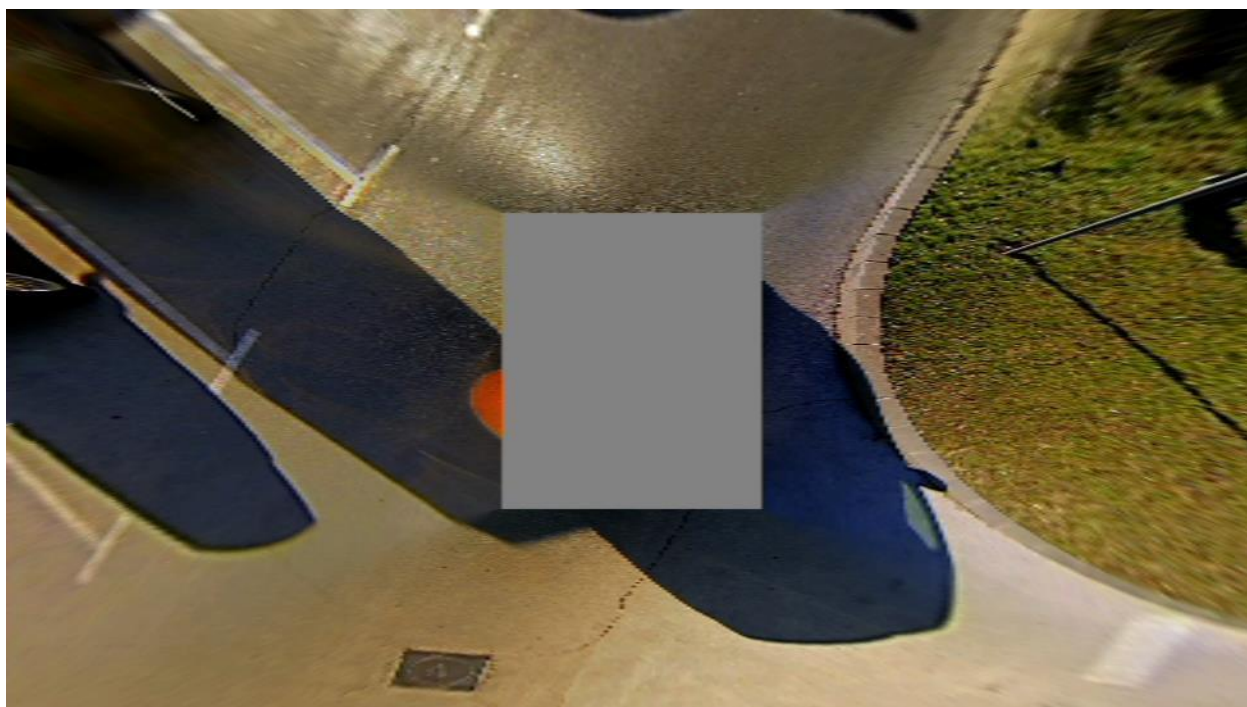
*Slika 6.21 Y rezultat 1. videozapisa.*

Ovdje kod Y komponente na prvu možemo uočiti izbijanje sivo plave boje.



*Slika 6.22 Rezultat Y rezultat 2. videozapisa*

U noćnim uvjetima izbijanje plave boje dolazi do izražaja, pogotovo na području gdje se spajaju sve kamere.



*Slika 6.23 Y rezultat 3. videozapisa.*

Rano jutarnji uvjeti se i dalje dobro drže kod Y komponente.

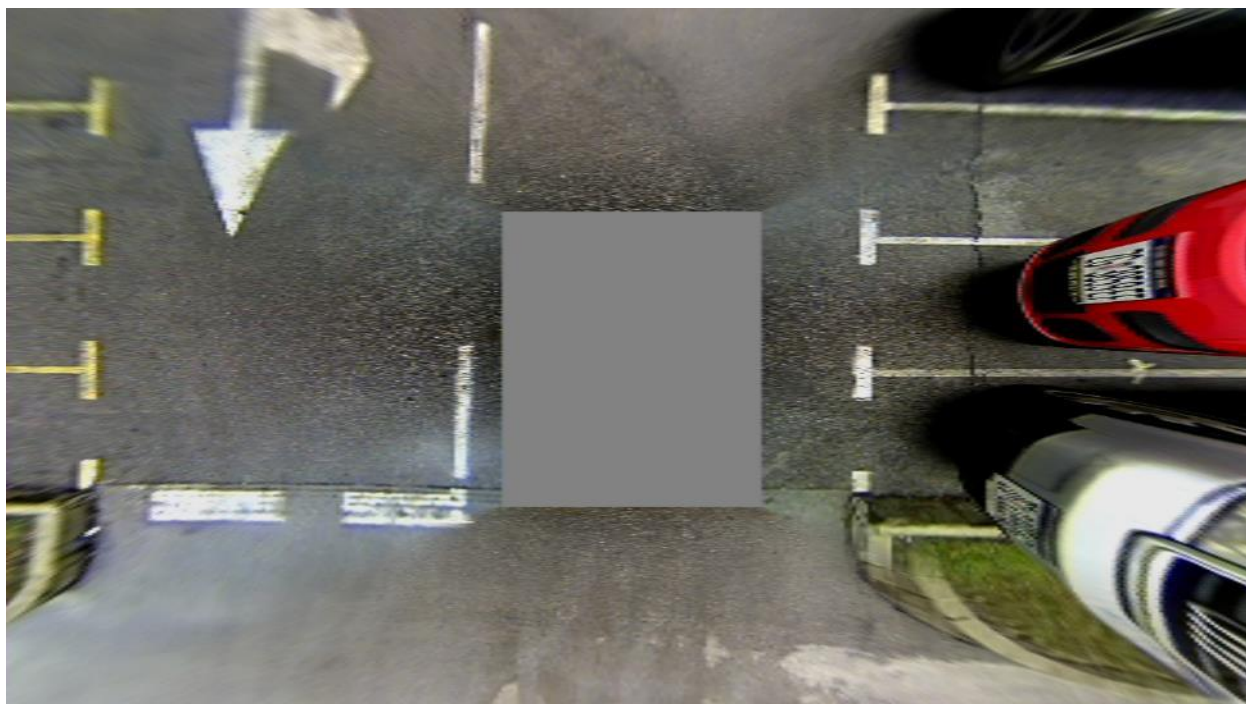


*Slika 6.24 Y rezultat 4. videozapisa.*

Večernji uvjeti su nešto svjetliji nego kao što je to slučaj sa RGB komponentom.

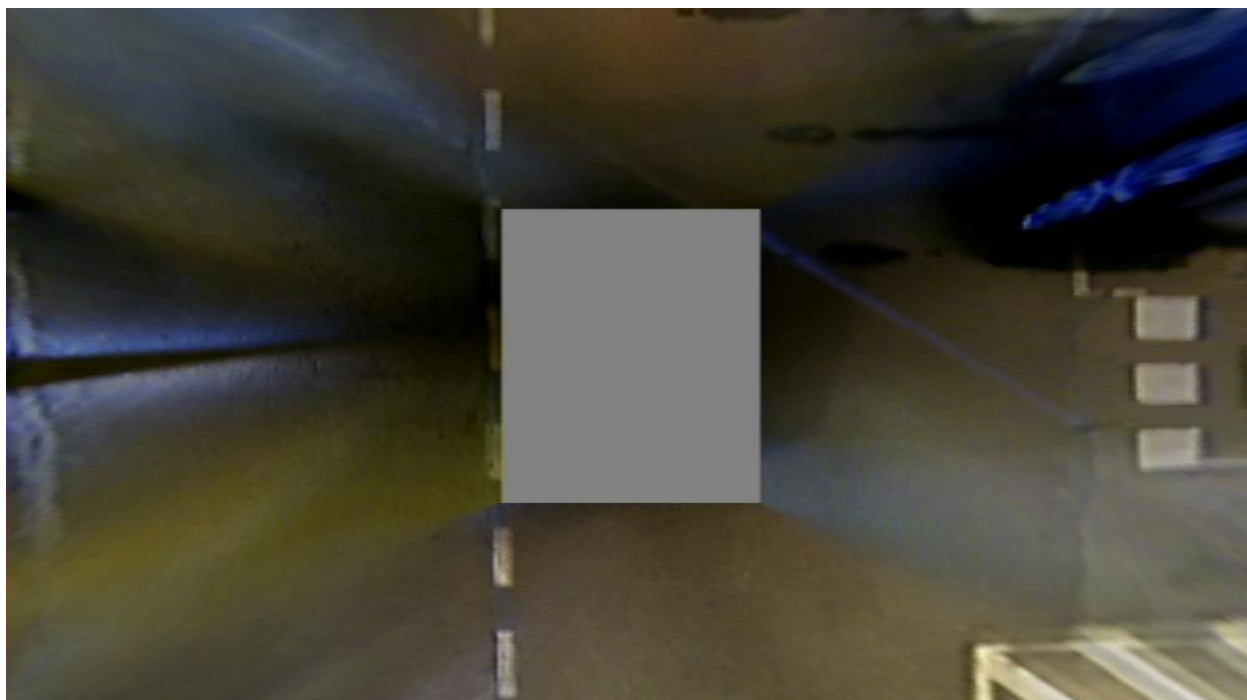
Za drugi slučaj ćemo koristiti sljedeće postavke:

- $Konst1 = 1$ ,
- $Konst2 = 4$ .



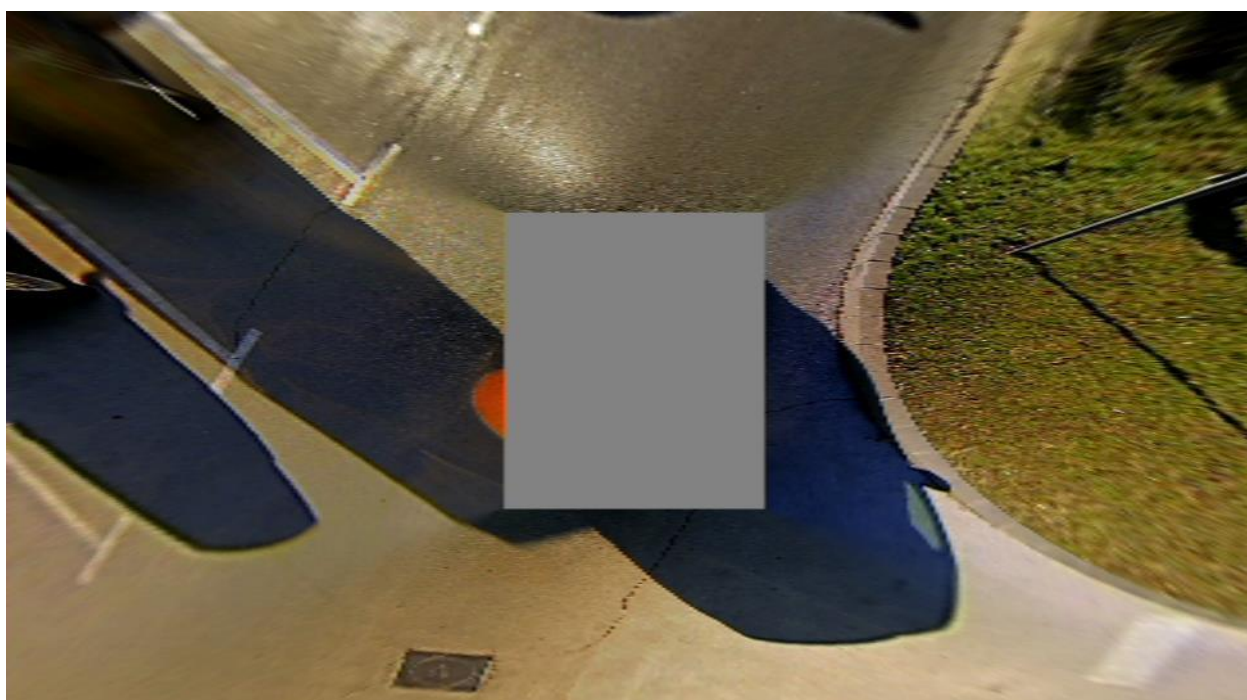
*Slika 6.25 Y rezultat 1. videozapisa.*

Sa povećanjem gornje granice slika se dodatno izbjeljuje. Jasnoća je poprilično ista kao i kod manje gornje granice.



*Slika 6.26 Y rezultat 2. videozapisa.*

Noćni uvjeti su svjetliji kod povećanja gornje granice.



*Slika 6.27 Y rezultat 3. videozapisa.*

U ranim jutarnjim uvjetima promjene su teže uočljive jer je slika sama po sebi oštra, no možemo vidjeti da je nešto svjetlija u odnosu na početnu sliku.

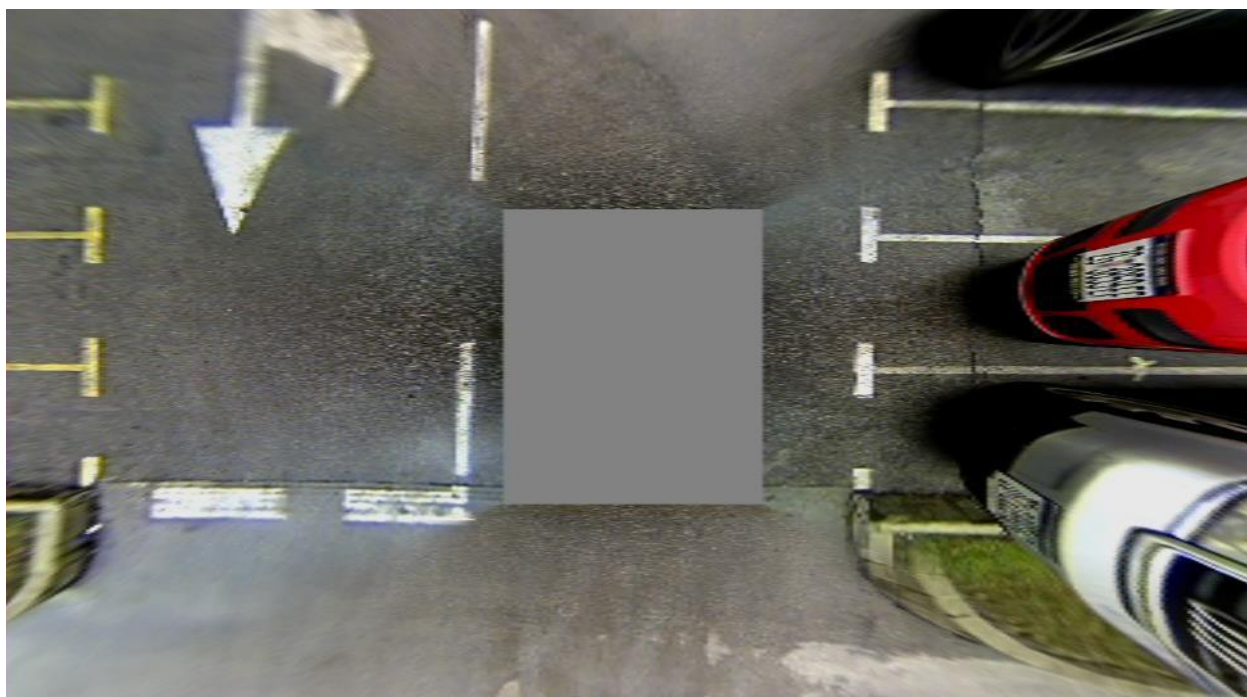


*Slika 6.28 Y rezultat 4. videozapisa.*

Večernji uvjeti su i dalje svjetliji. Ne gubi se na ošttrini.

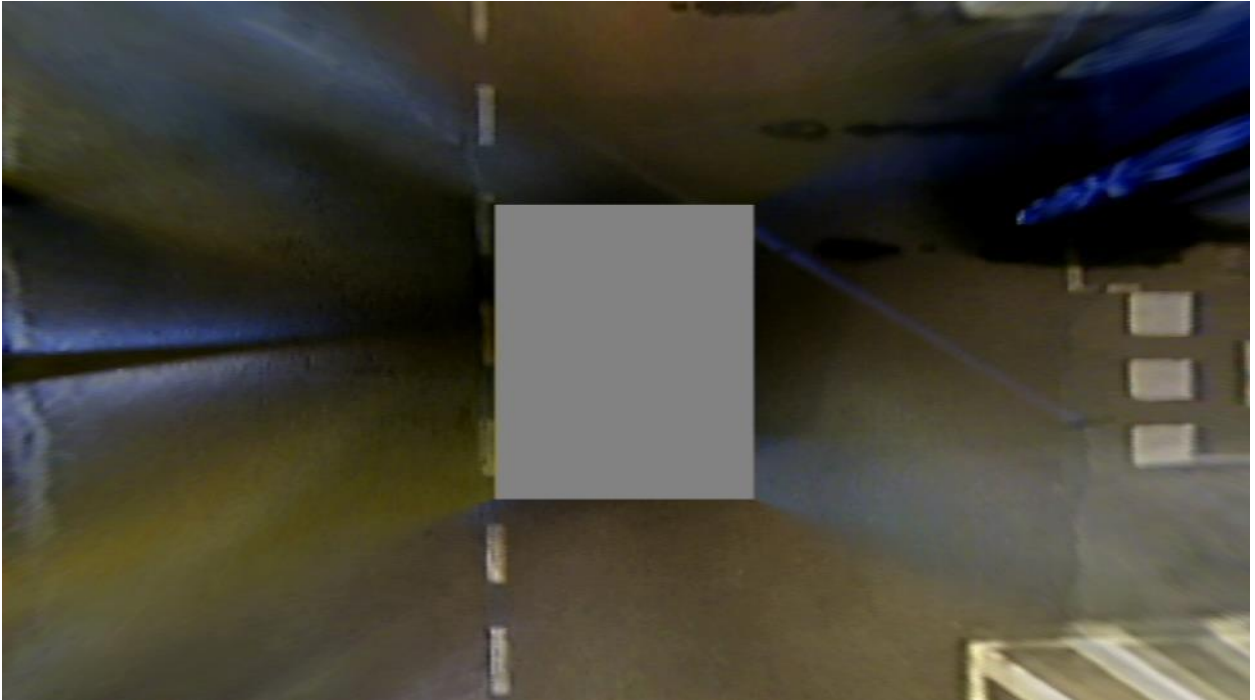
Za treći slučaj ćemo koristiti sljedeće postavke:

- $\text{Konst1} = 1$ ,
- $\text{Konst2} = 5$ .



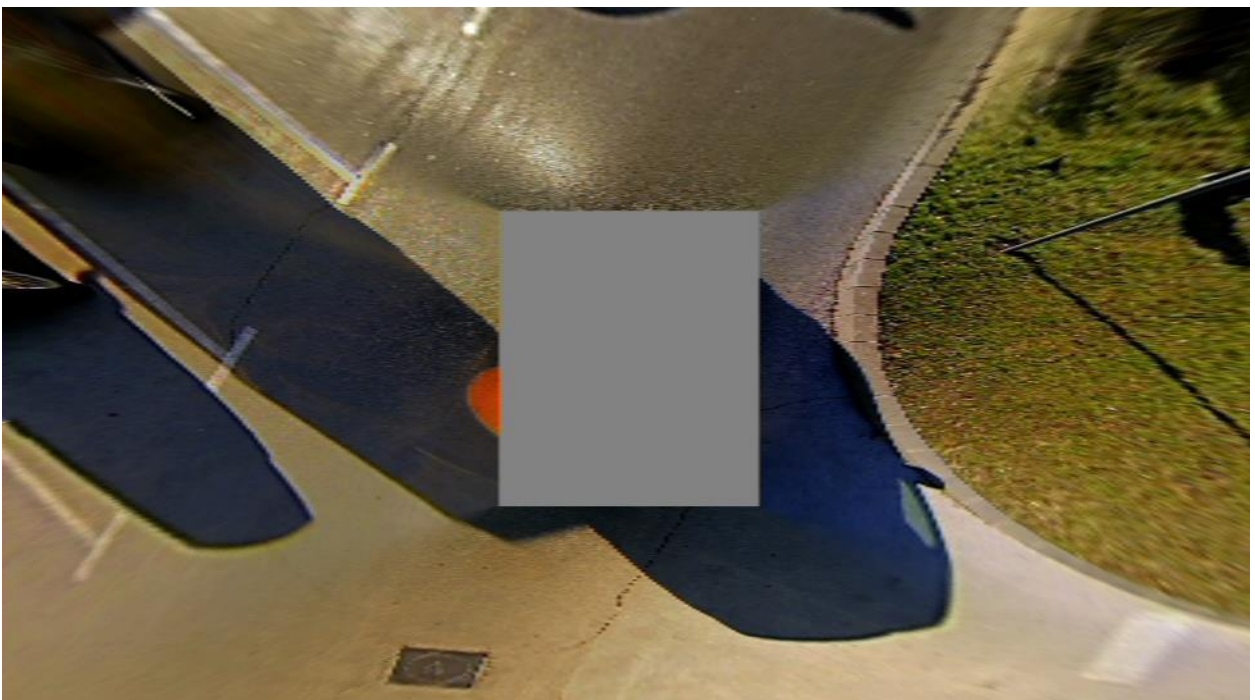
*Slika 6.29 Y rezultat 1. videozapisa.*

Najveća gornja granica dodatno prosvjetljuje sliku, možemo vidjeti izbijanje bijele boje u donjem lijevom kutu. Vidi se to i u prijašnjim rezultatima no sad je nešto uočljivije.



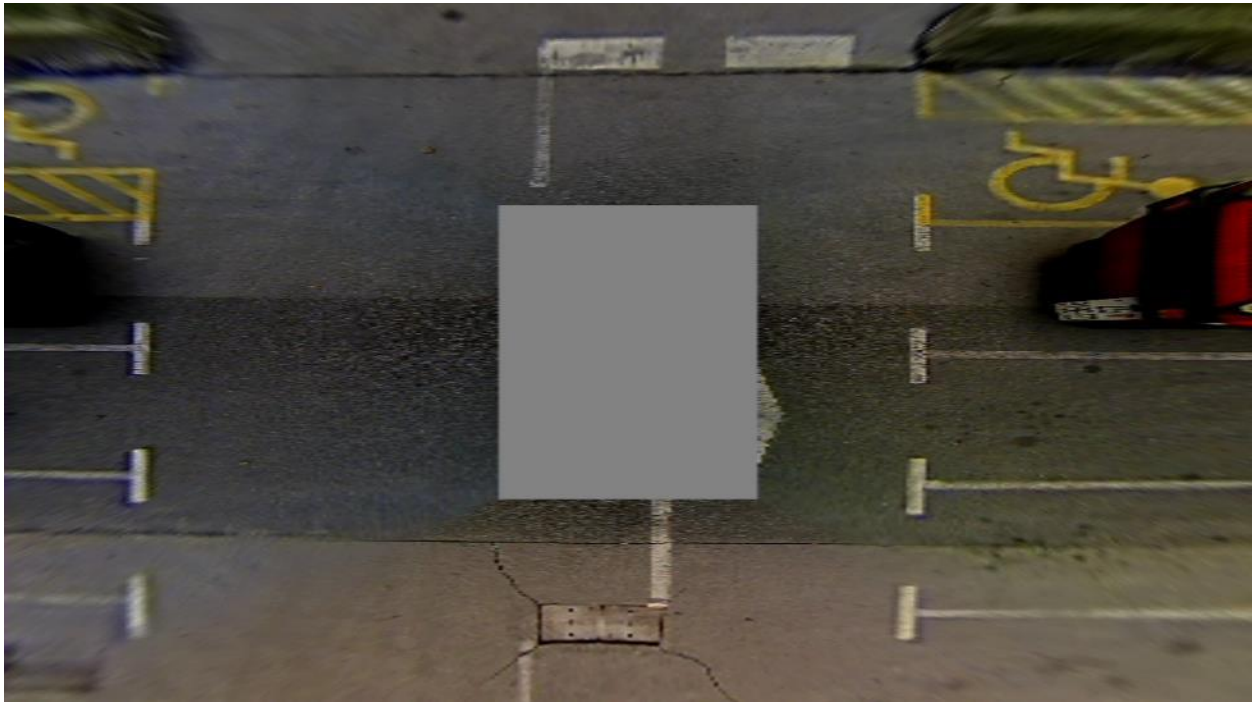
*Slika 6.30 Y rezultat 2. videozapisa.*

Noćni uvjeti nisu dobili neku značajniju promjenu.



*Slika 6.31 Y rezultat 3. videozapisa.*

Kod rano jutarnjih uvjeta možemo vidjeti da sunčeva svjetlost ne izbija na ulici, odnosno ne reflektira toliko kao i na prethodnim slikama.



*Slika 6.32 Y rezultat 4. videozapisa.*

Večernji uvjeti su najviše profitirali od povećanja gornje granice, jer mrak se dodatno osvjetljuje tako da se lakše uoči okolina.

Kao što je slučaj u prijašnjoj usporedbi, Y komponenta se znatno brže renderira i manje prostora zauzima. Također prema priloženim slikama možemo vidjeti razliku u odnosu na rastuće vrijednosti gornje granice pojačanja.

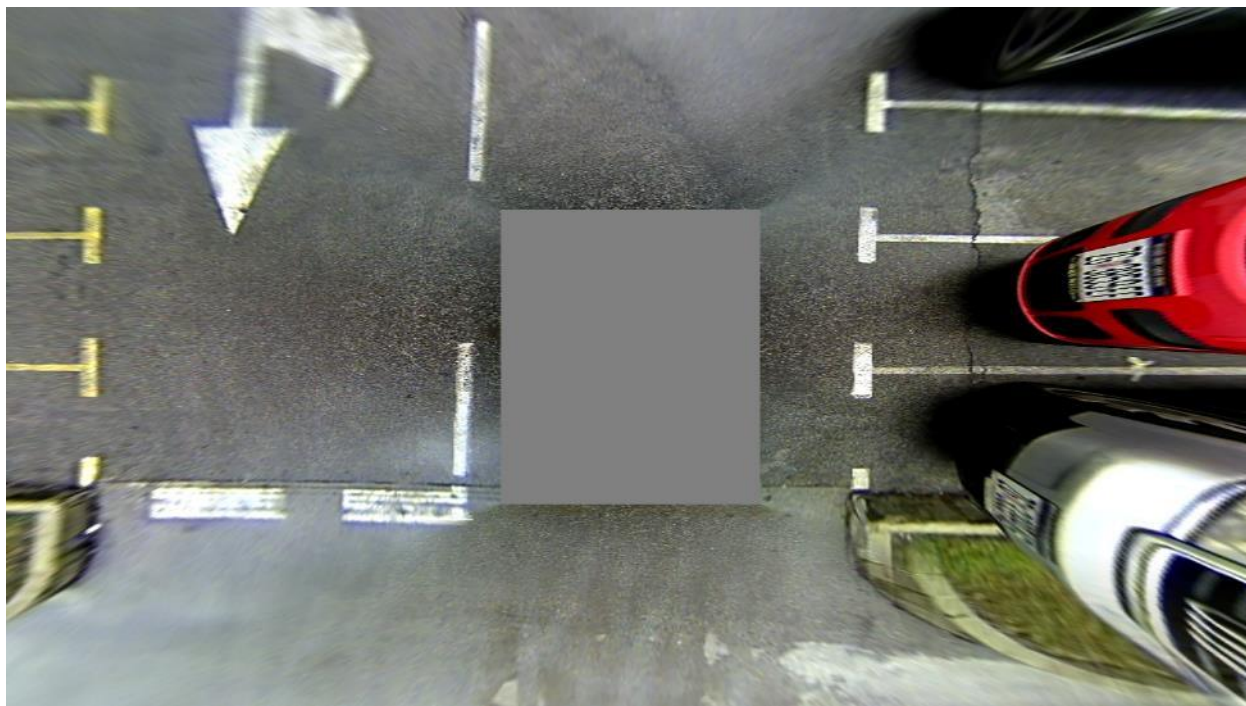
### **6.3. Podešavanje svjetline slike**

U ovom slučaju ćemo promijeniti vrijednost `norm_rgb` za RGB i `norm_bright` za YCbCr sa 1 u 0 i time vidjeti da li je videozapis bolji dok je svjetliji.

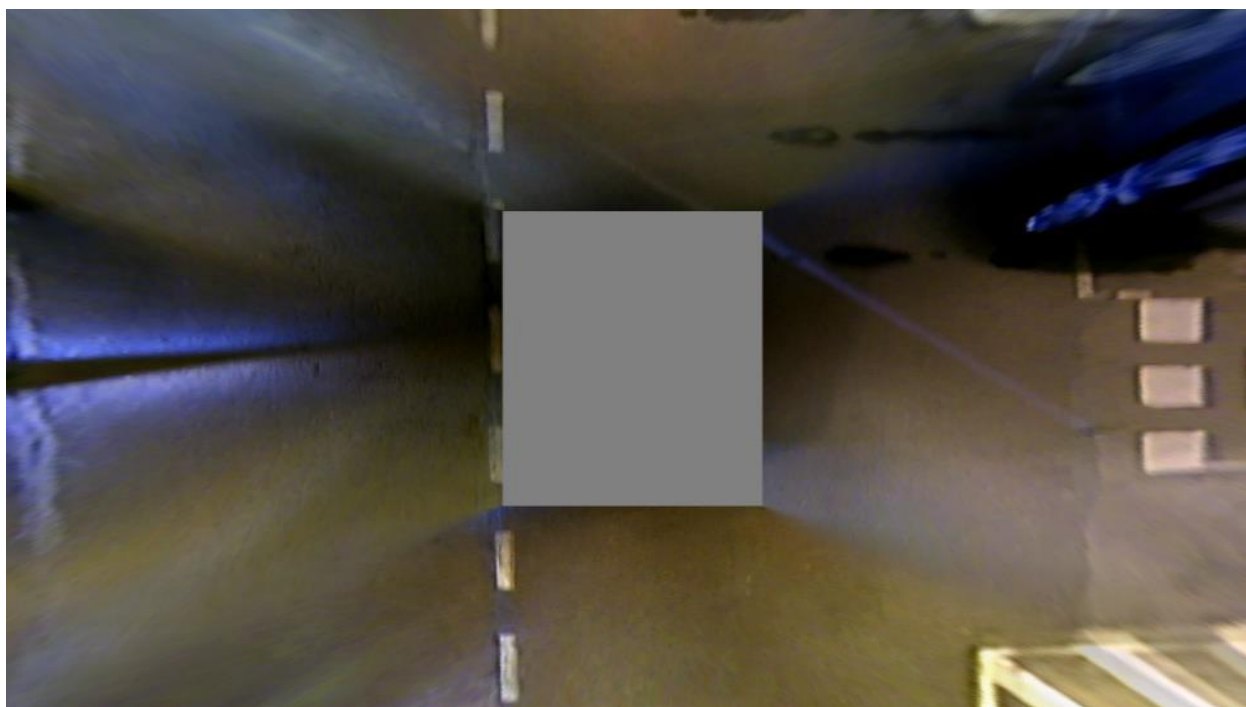


### 6.3.1. RGB komponenta

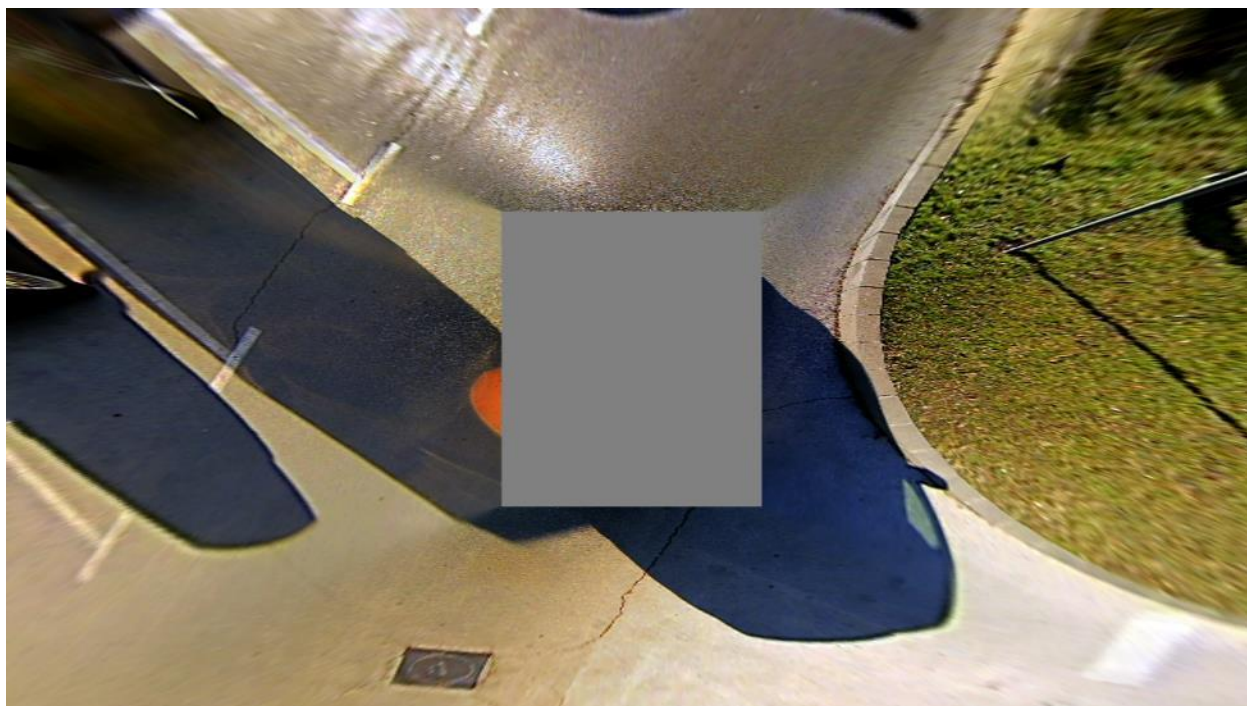
Za RGB komponentu nam vrijedi da je `norm_rgb` promijenjeno u nulu sa početne vrijednosti koja je bila jedan.



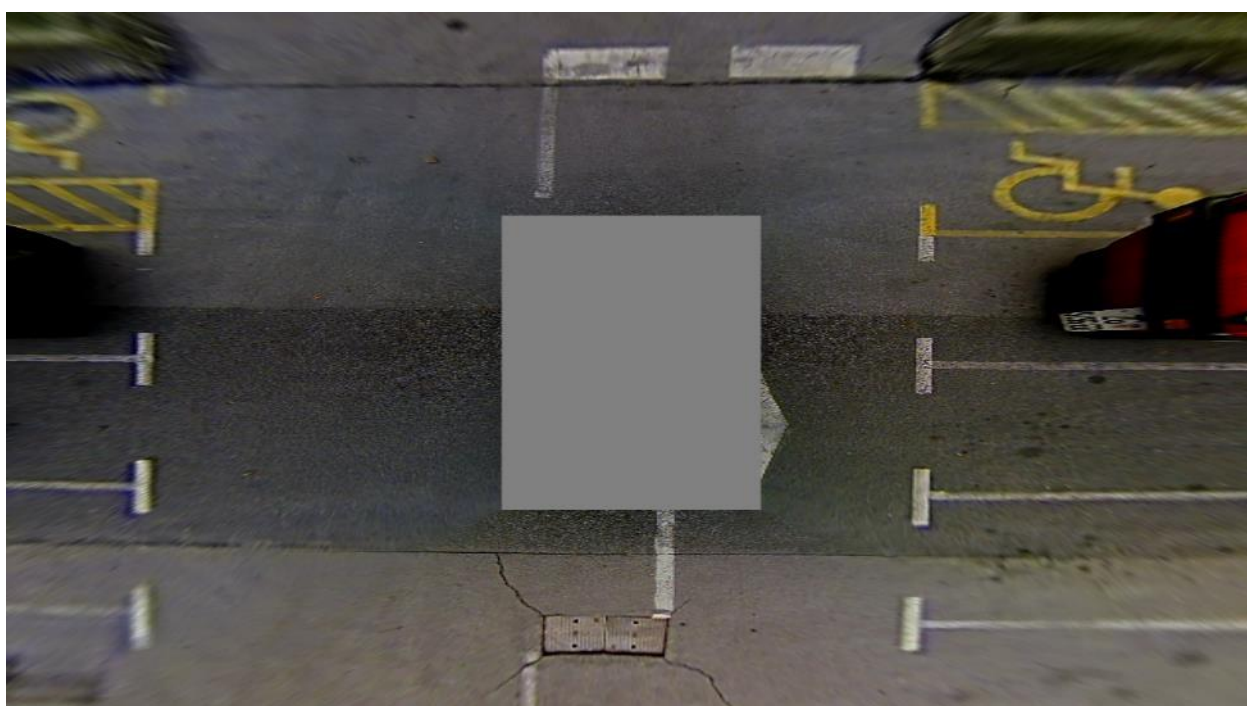
*Slika 6.33 RGB rezultat 1. videozapisa.*



*Slika 6.34 RGB rezultat 2. videozapisa.*



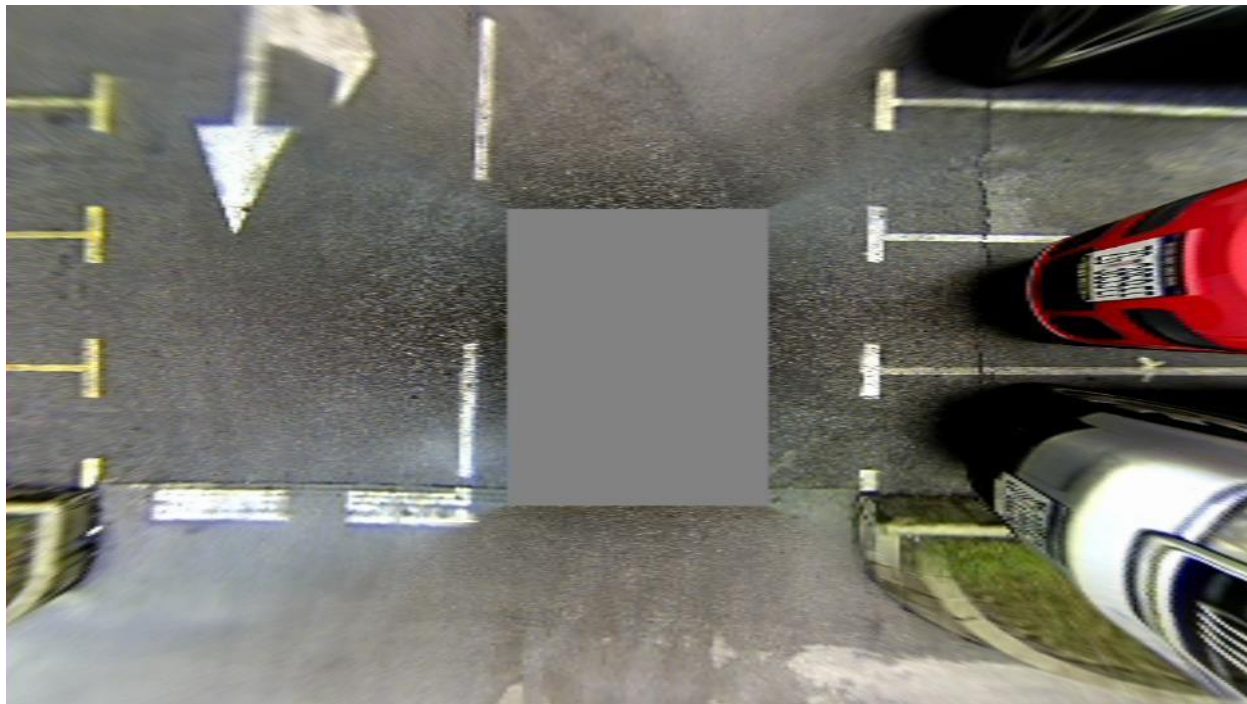
*Slika 6.35 RGB rezultat 3. videozapisa.*



*Slika 6.36 RGB rezultat 4. videozapisa.*

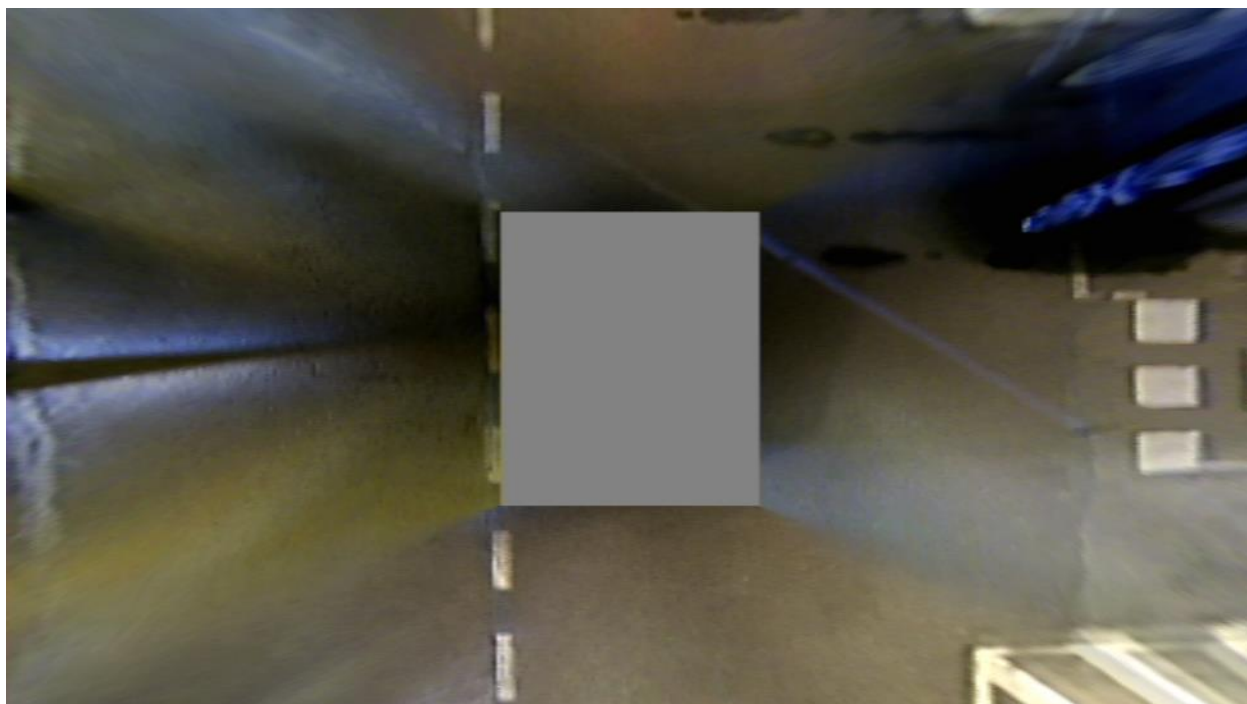
### 6.3.2. Y komponenta

Za Y komponentu nam vrijedi da je `norm_bright` promijenjeno u nulu sa početne vrijednosti koja je bila jedan.



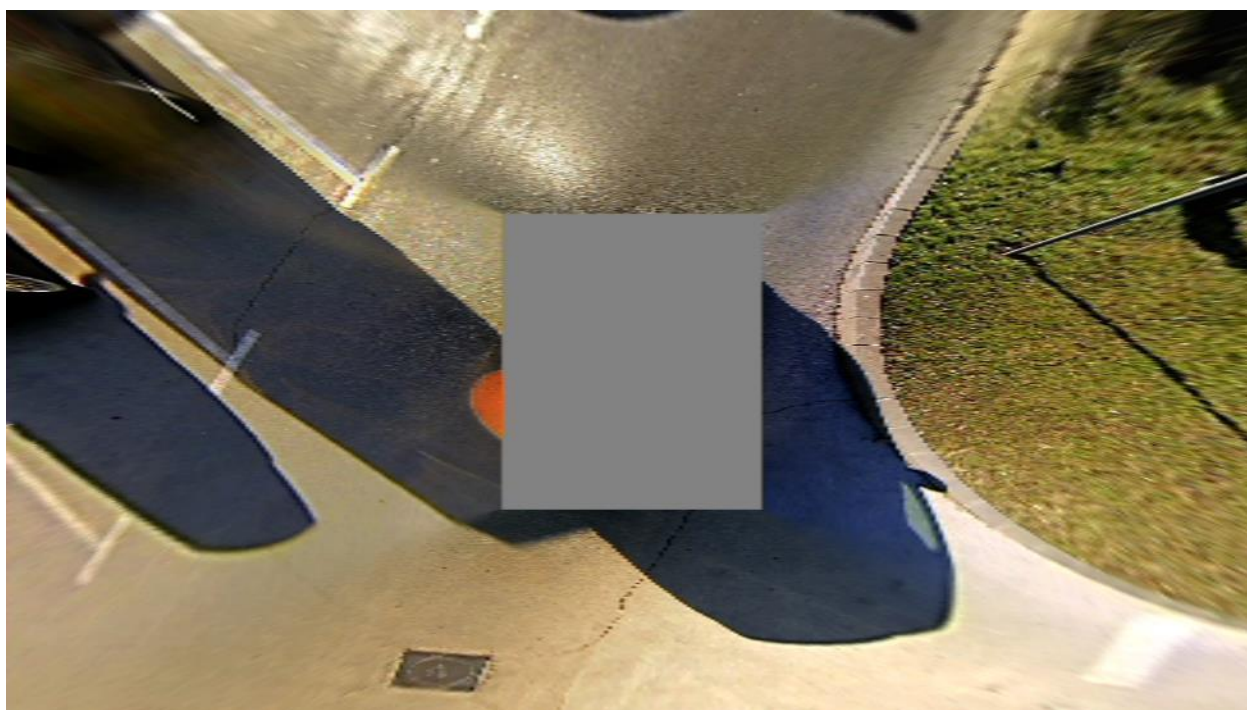
*Slika 6.37 Y rezultat 1. videozapisa.*

Ovdje možemo vidjeti izbijanje plave boje oko auta.



*Slika 6.38 Y rezultat 2. videozapisa.*

Također isti slučaj kao i kod prethodne slike, vidimo izbijanje plave boje i promjenjivu deformaciju slike u pokretu. To se najbolje vidi u gornjem desnom kutu gdje je parkiran auto te je u ovom slučaju izobličen do te mjere da se ne prepozna.



*Slika 6.39 Y rezultat 3. videozapisa.*

Dosta velika promjena kod jačeg sunca, vidimo blijede boje.

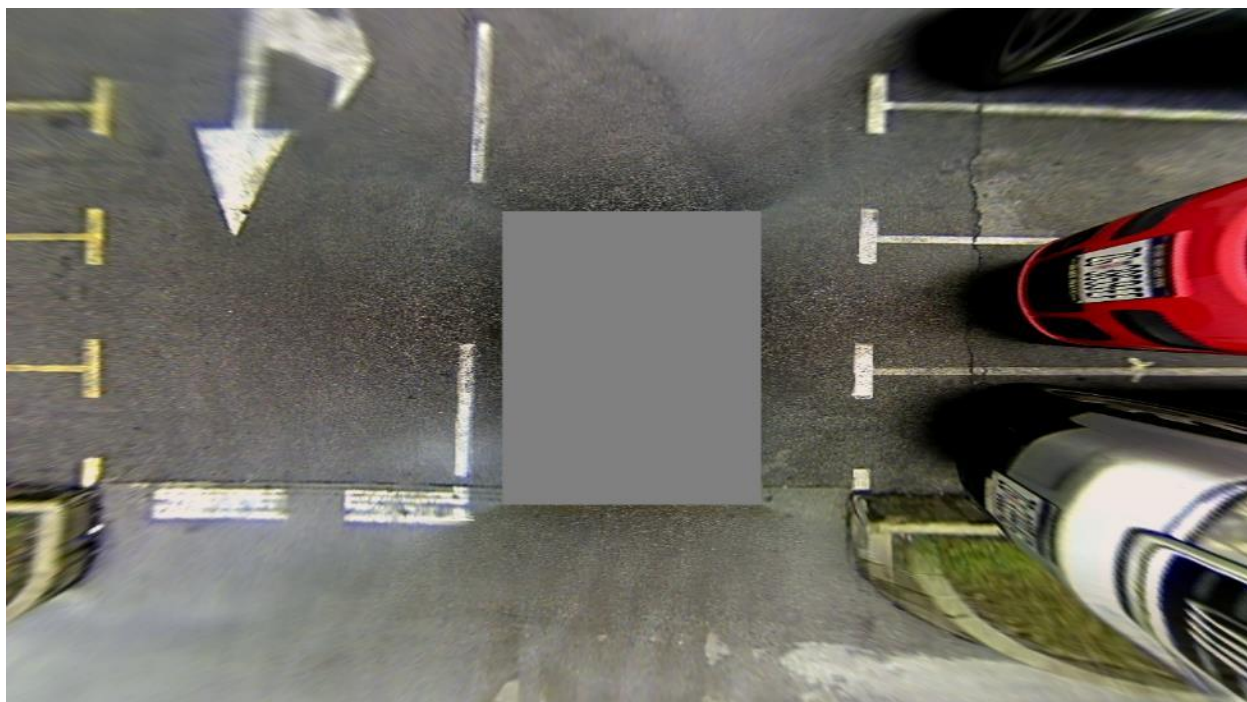


*Slika 6.40 Y rezultat 4. videozapisa.*

Večernji uvjeti su i dalje svjetliji ali slika nije zamagljena.

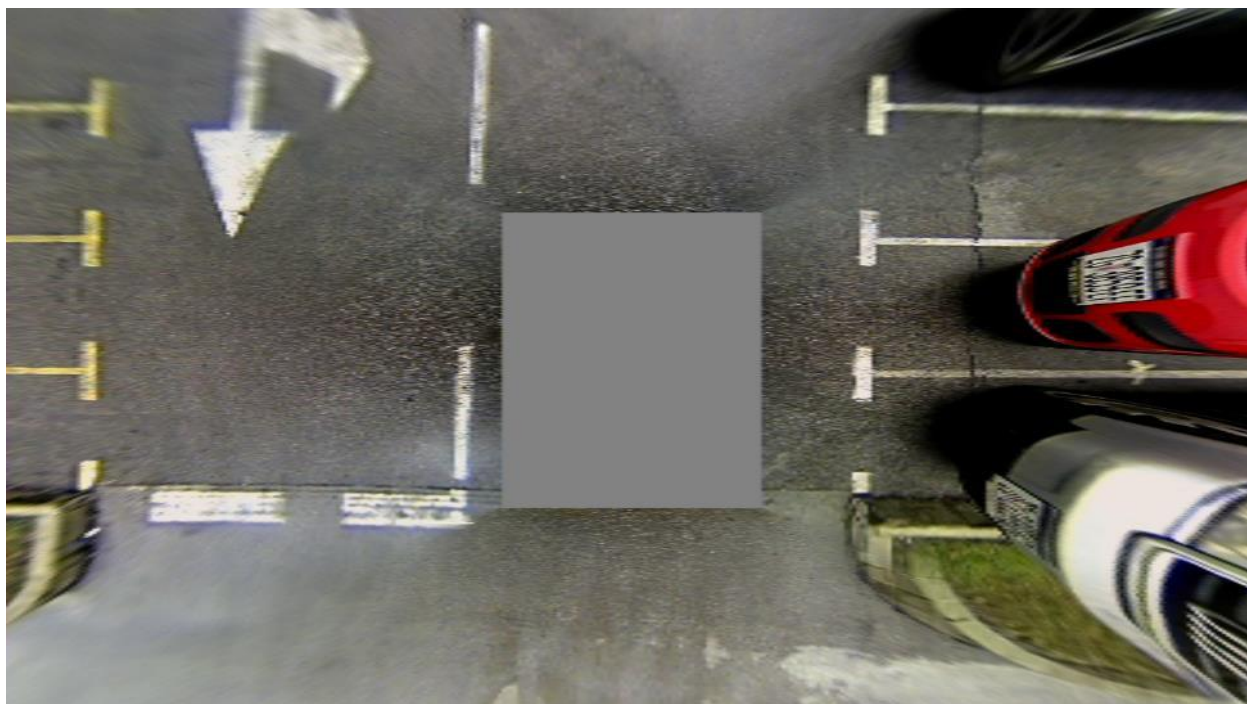
#### **6.4. Usporedba optimizacija po RGB koristeći YCbCr**

Slike dobivene Y optimizacijom su duplo uže u odnosu na one dobivene RGB-om, no za potrebe ovog rada smo ih proširili na istu veličinu kao i RGB kako bi se lakše uočile razlike.



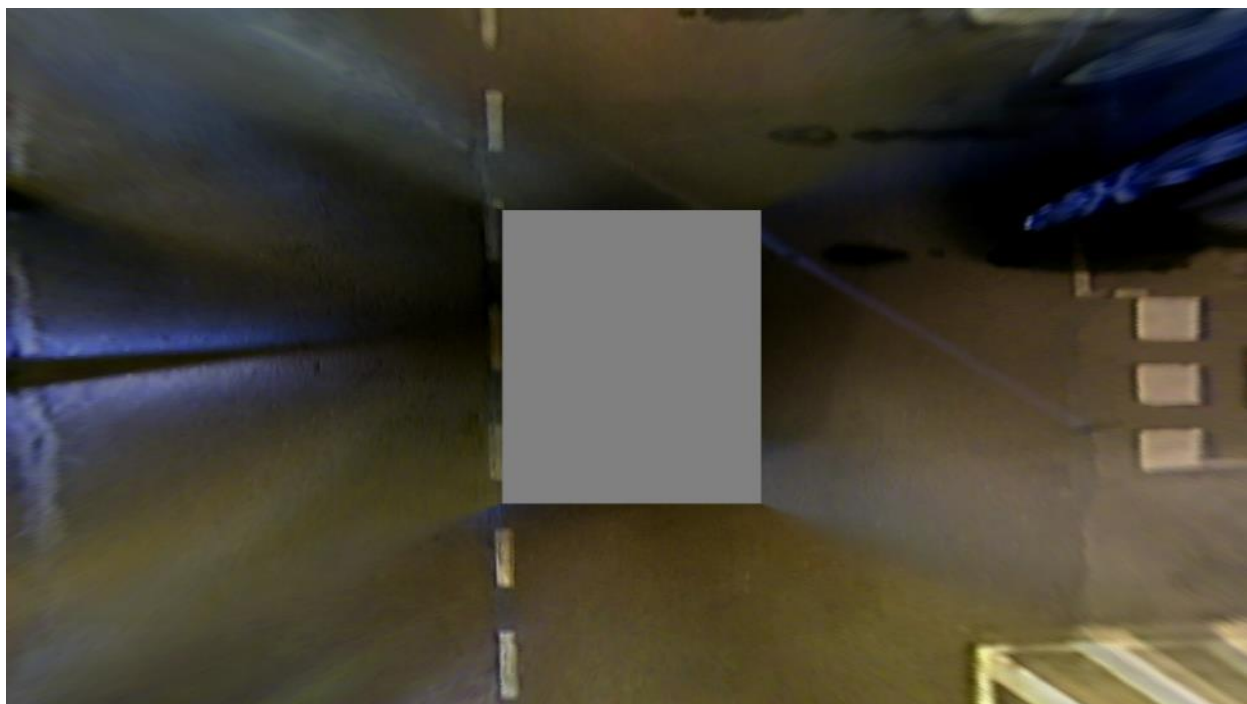
*Slika 6.41 RGB rezultat 1. videozapisa.*

Slika je jasna i oštra.



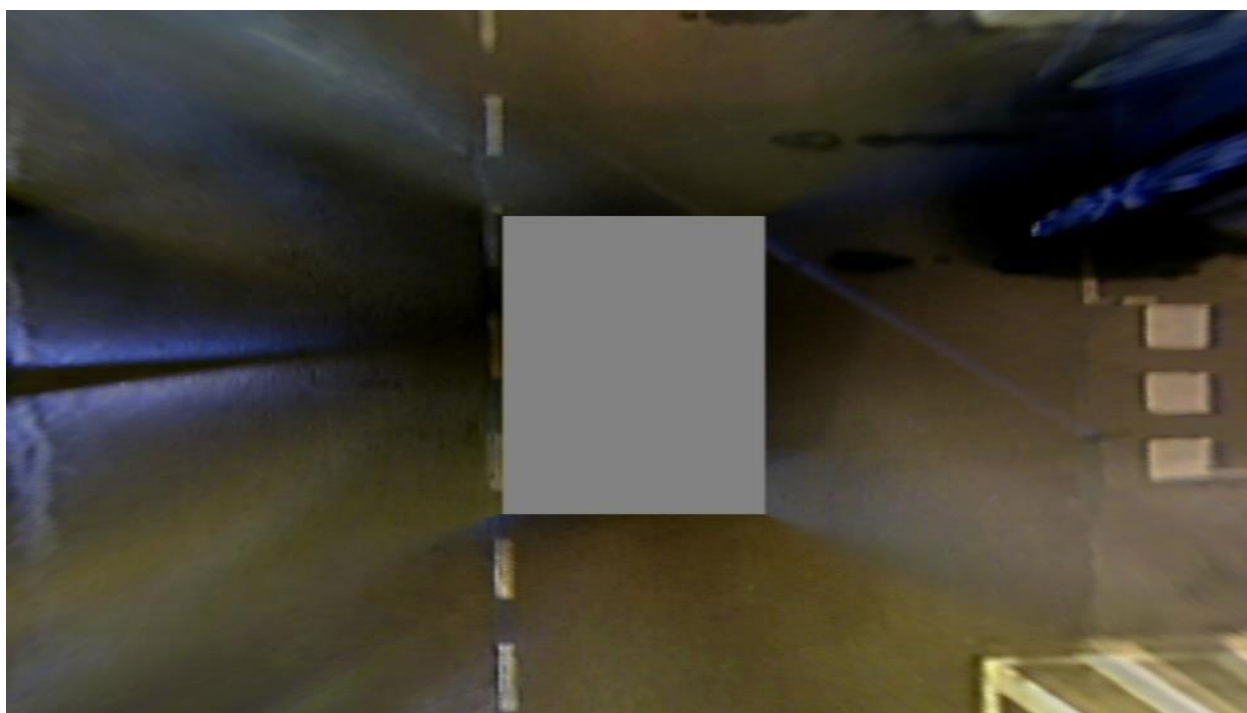
*Slika 6.42 YCbCr rezultat 1. videozapisa.*

Kod YCbCr vidimo da se gubi jasnoća i da je slika nešto mutnija nego prethodna.



*Slika 6.43 RGB rezultat 2. videozapisa.*

Klasični noćni uvjeti. Slika nije najoštrija te izbijaju plava, bijela i siva boja.



*Slika 6.44 YCbCr rezultat 2. videozapisa.*

YCbCr slika je nijansu mutnija no na prvi pogled gotovo neprimjetno s obzirom da oba dvije komponente imaju problema u noćnim uvjetima.



*Slika 6.45 RGB rezultat 3. videozapisa.*

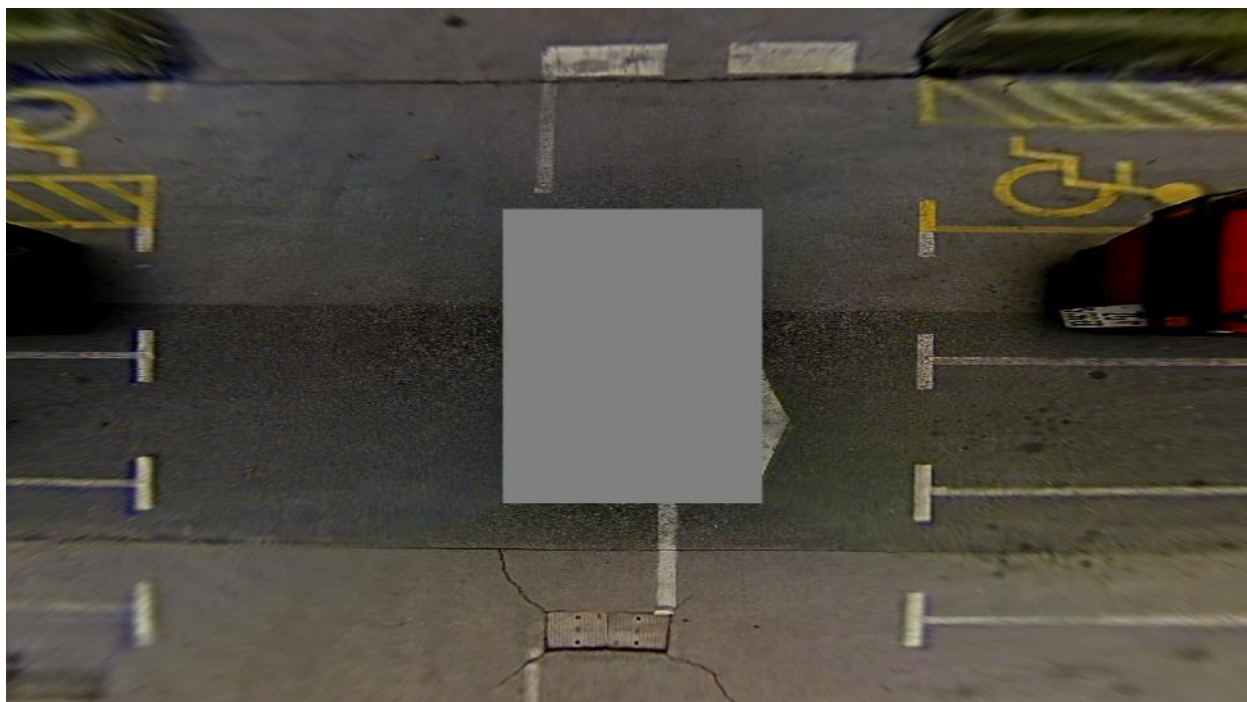
Jutarnji uvjeti, slika je zasićena i vidi se veća saturacija narančaste boje.



*Slika 6.46 YCbCr rezultat 3. videozapisa.*

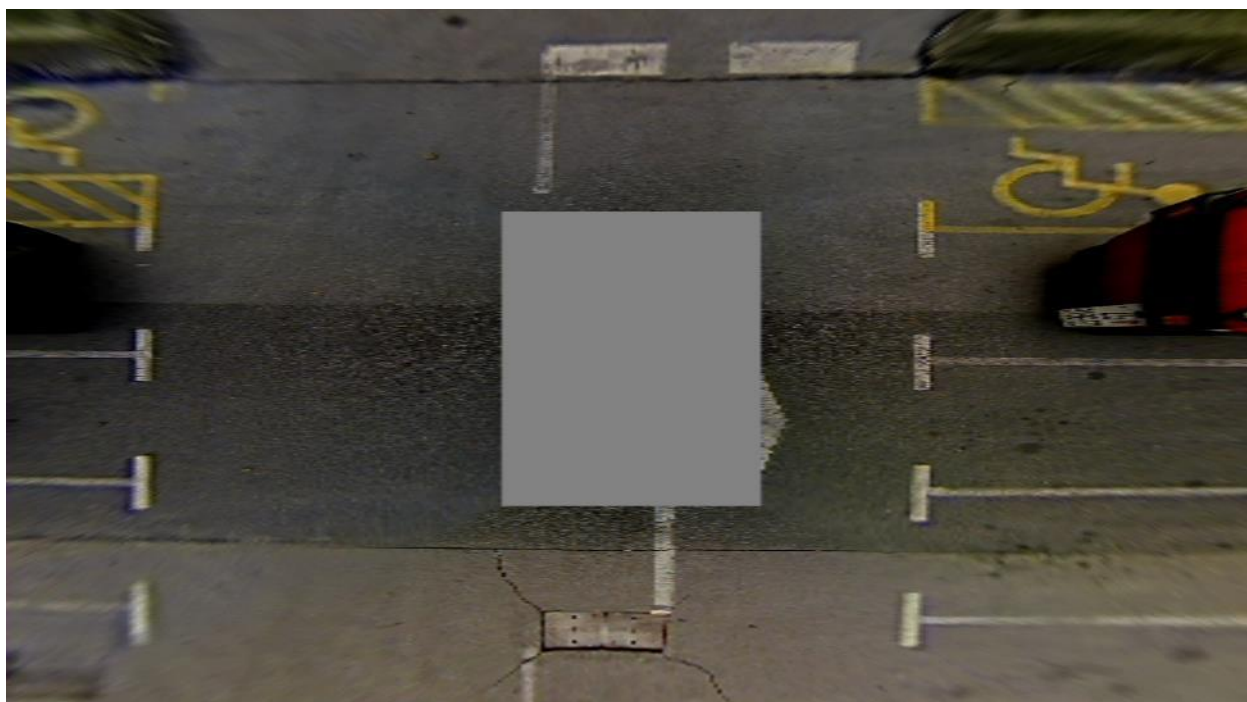
Kod YCbCr slika balansira zasićenje narančaste boje, no možemo uočiti izbijanje plave i zelene boje oko auta.





*Slika 6.47 RGB rezultat 4. videozapisa.*

Slika je jasna i oštra, ne vide se neka značajnija izbijanja drugih boja.



*Slika 6.48 YCbCr rezultat 4. videozapisa.*

Slika je nešto mutnija nego kod RGB komponente no i dalje je jasna. Golim okom neprimjetna razlika ukoliko ne idemo u daljnju analizu.

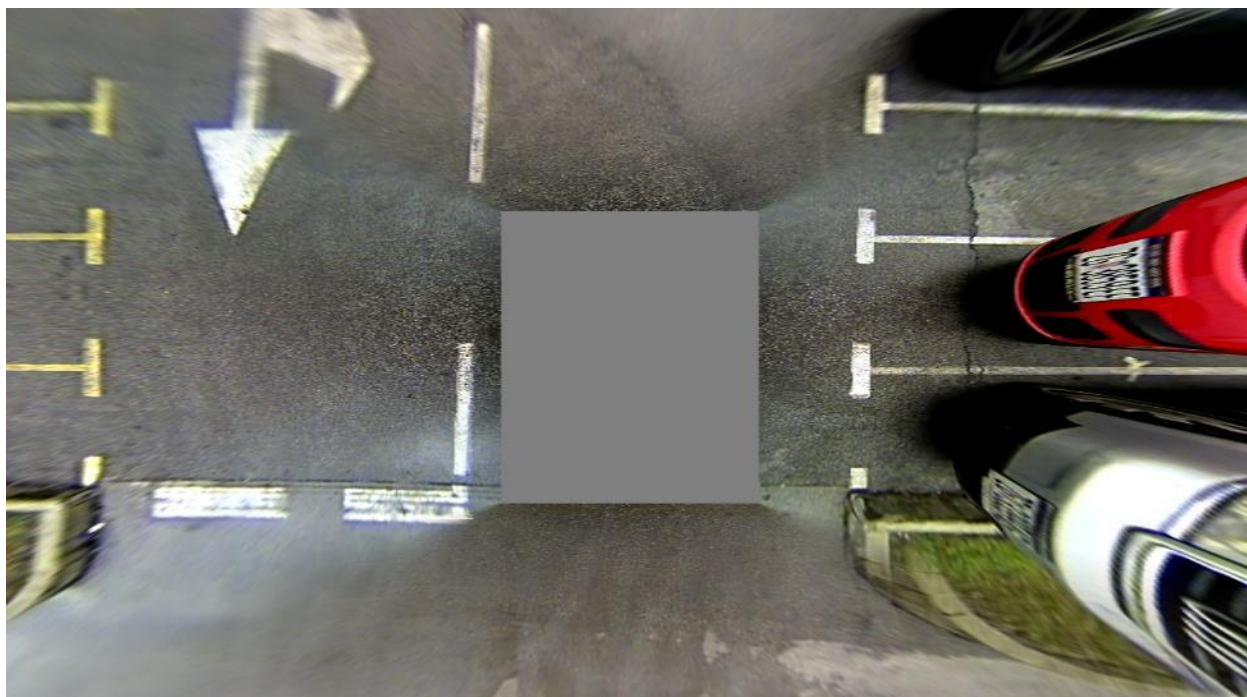
Nakon što smo prošli sve slučajeve, odnosno slike, možemo vidjeti da su slike kod Y komponente nešto blijede i sivo – plave u odnosu na RGB komponentu. Također možemo uočiti izbijanje sive i bijele boje na nekim slikama Y komponente u slučaju povećanja gornje granice.

Ne gubimo nešto previše na kvaliteti, jasno je da se malo jasnoće izgubi no ništa previše da bi značajno utjecalo na sami pregled okoline.

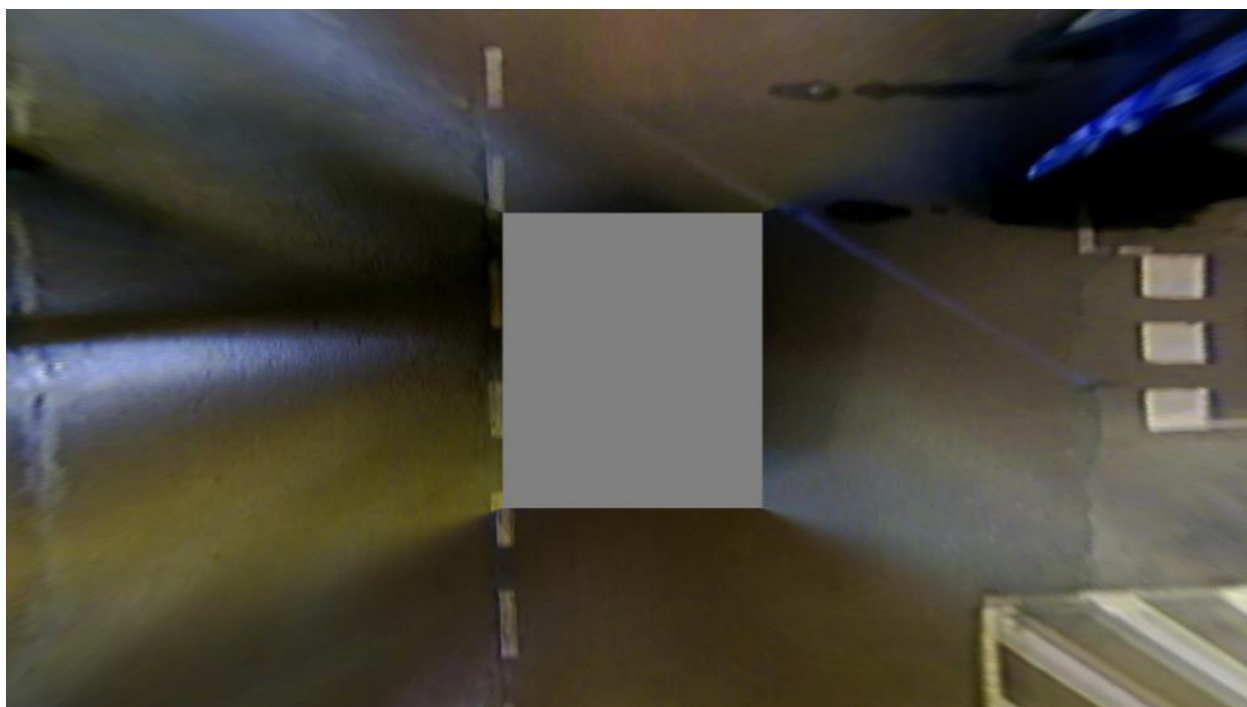
Koristeći YCbCr optimizaciju, koja interno računa koeficijente optimizacije po R, G i B komponentama, možemo vidjeti da nam je krajnji cilj zadovoljen jer pretvorene YCbCr slike izgledaju poprilično identično kao i kod RGB komponente. Mogli bi se i postići drugačiji rezultati, odnosno nešto bolji rezultati da smo na početku računali interpolaciju Cb i Cr komponenti na istu veličinu kao Y komponenta (umjesto smanjivanja Y na veličinu Cb i Cr komponenti). Na taj način ne bi izgubili informacije u ulaznoj Y komponenti a i krajnja veličina slike bi ostala nepromijenjena. No samim time bi cijeli proces računanja koeficijenata pojačanja bio sporiji.

## **6.5. Zapis bez izjednačavanja**

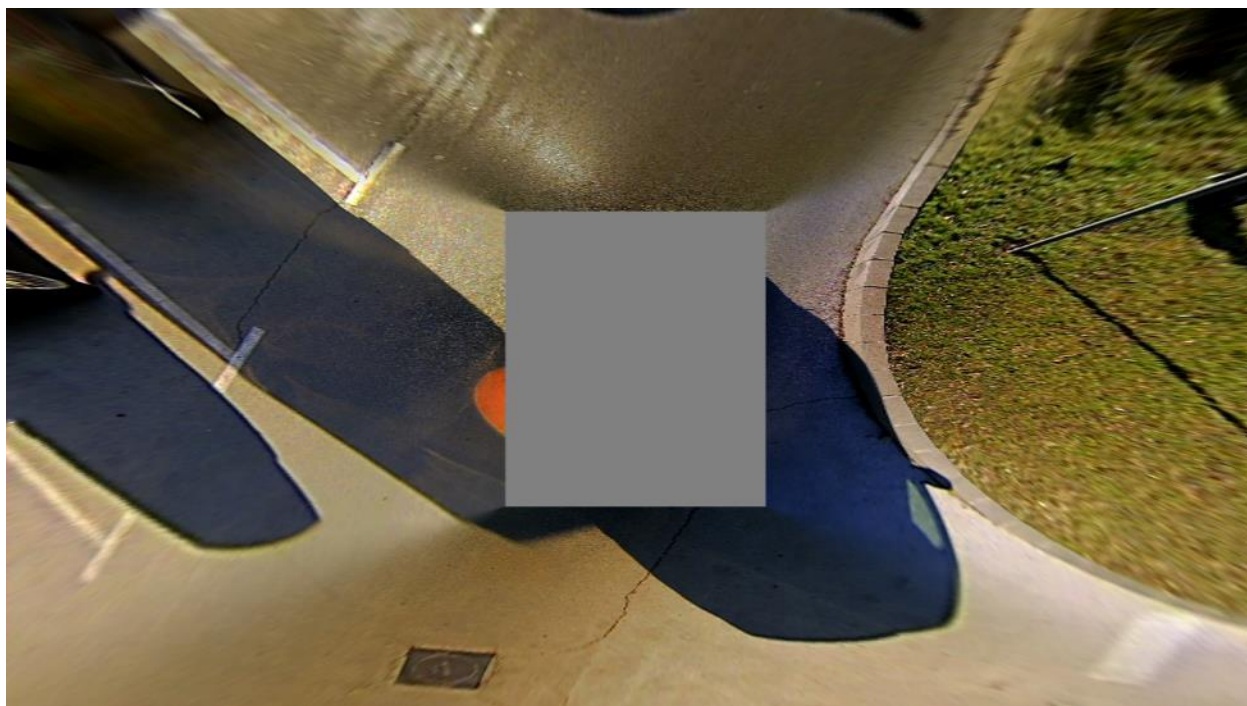
Nakon svih promjena, stavit ćemo obične slike. Slike su stavljene na kraj analize iz razloga da se ne vraćamo desetak stranica kako bi vidjeli kako slike izgledaju bez izjednačavanja. Tako će se moći vidjeti krajnji rezultat prije samog zaključka.



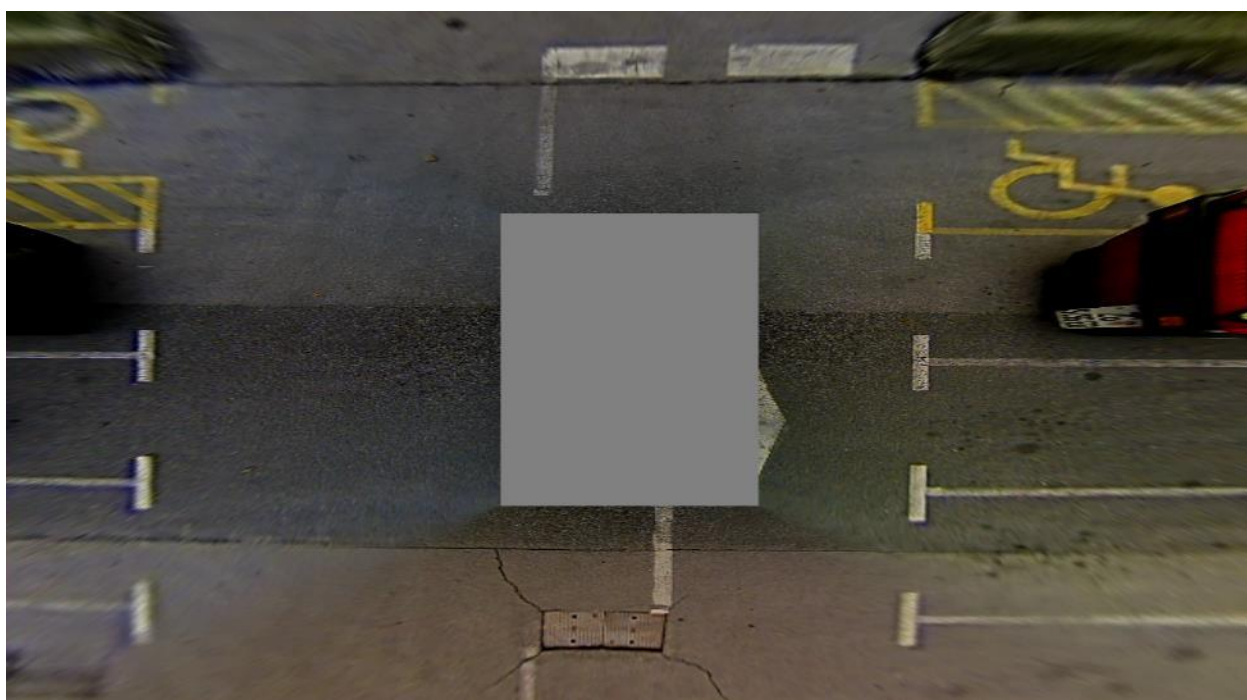
*Slika 6.49 Bez izjednačavanja, dnevni uvjeti.*



*Slika 6.50 Bez izjednačavanja, noćni uvjeti.*



*Slika 6.51 Bez izjednačavanja, jutarnji uvjeti.*



*Slika 6.52 Bez izjednačavanja, večernji uvjeti.*

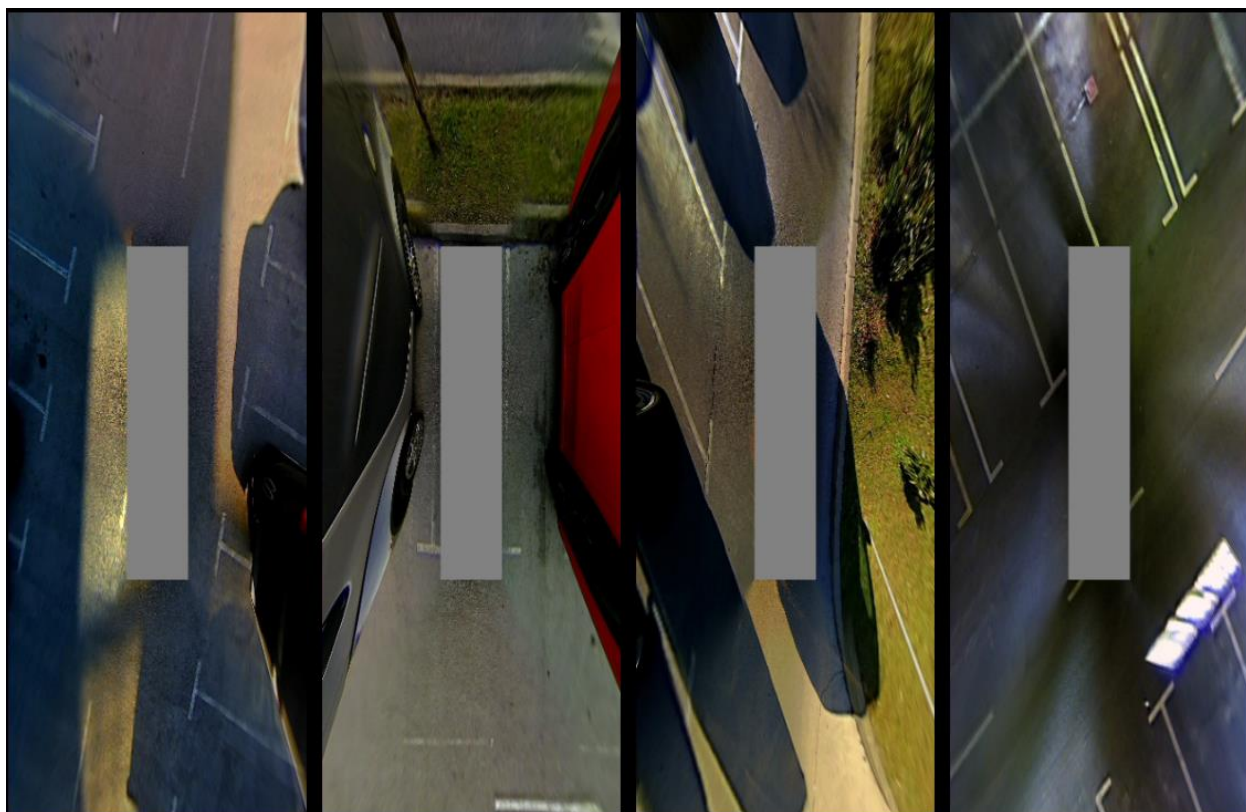
Sada na kraju možemo vidjeti kako bi video, odnosno slike, izgledali bez izjednačavanja. Izjednačavanje radimo kako bi dodatno „namjestili“ sliku, ne u smislu manipulacije iskrivljenja slike kao što je to slučaj kod osobnih slika, već kako bi vozaču dali što više je moguće jasniju sliku okoline.

Samim time što je čovjeku jasnija slika, postoji manja šansa od nejasnoća, artefakta i konfuzije. Sve to utječe na sigurnost vožnje, a time ujedno i na sigurnost života samih vozača.

## 6.6. Direktna usporedba slika

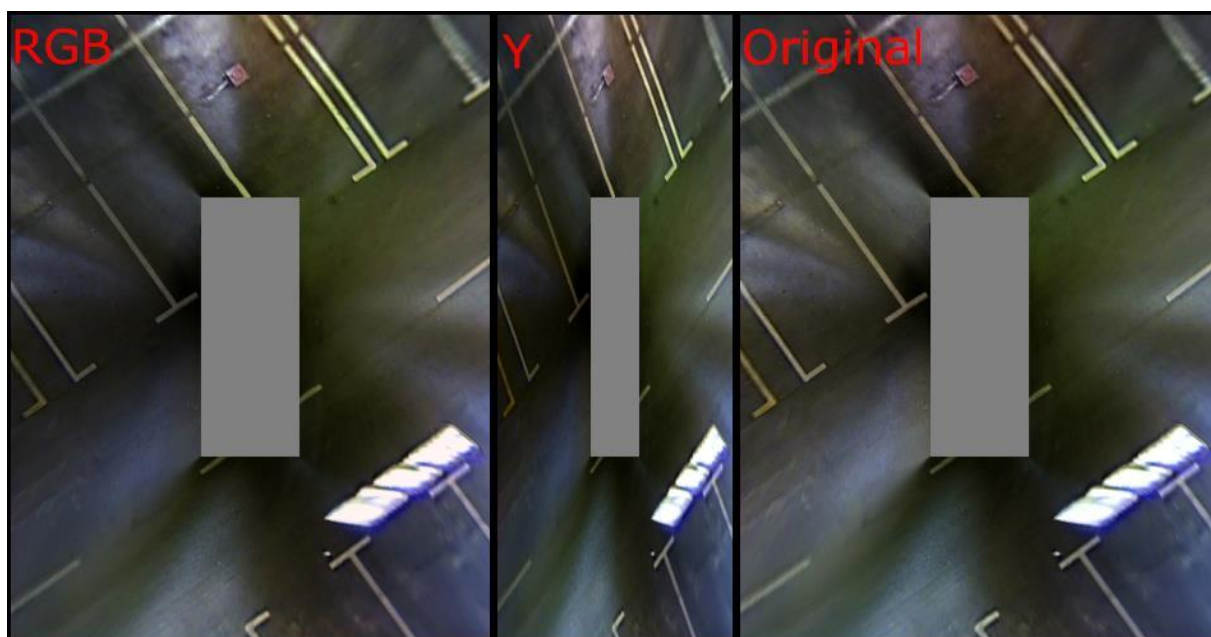
Kako bi krajnjem korisniku što točnije mogli predočiti i objasniti o čemu se tu točno radi, spojili smo sve komponente (original, Y i RGB) u jednu sliku. Ono što moramo imati na umu je to da Y komponenta će biti uža zbog formata 4:2:2, dok će original i RGB slike biti normalne veličine.

Za početak prikazat ćemo sve slike YCbCr po RGB-u. Slike su iste kvalitete, samo su duplo uže.



*Slika 6.53 YCbCr po RGB.*

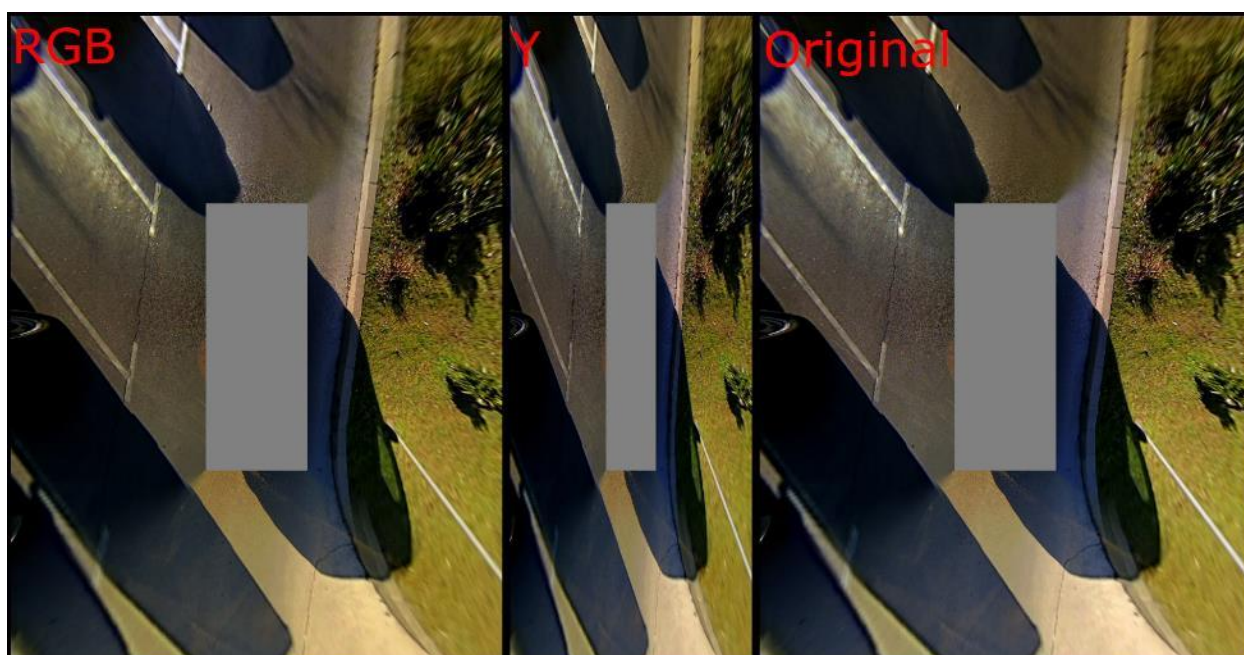
Prva slika su nam noćni uvjeti vožnje.



*Slika 6.54 Direktna usporedba slika po noćnim uvjetima.*

Kod noćnih uvjeta možemo najbolje vidjeti razliku o čemu se radi. U odnosu na originalnu sliku, RGB je nešto svjetlija i blijeda. Y slika osim što je uža, za nijansu je svjetlija od originalne no nije ispijena kao što je to slučaj sa RGB slikom.

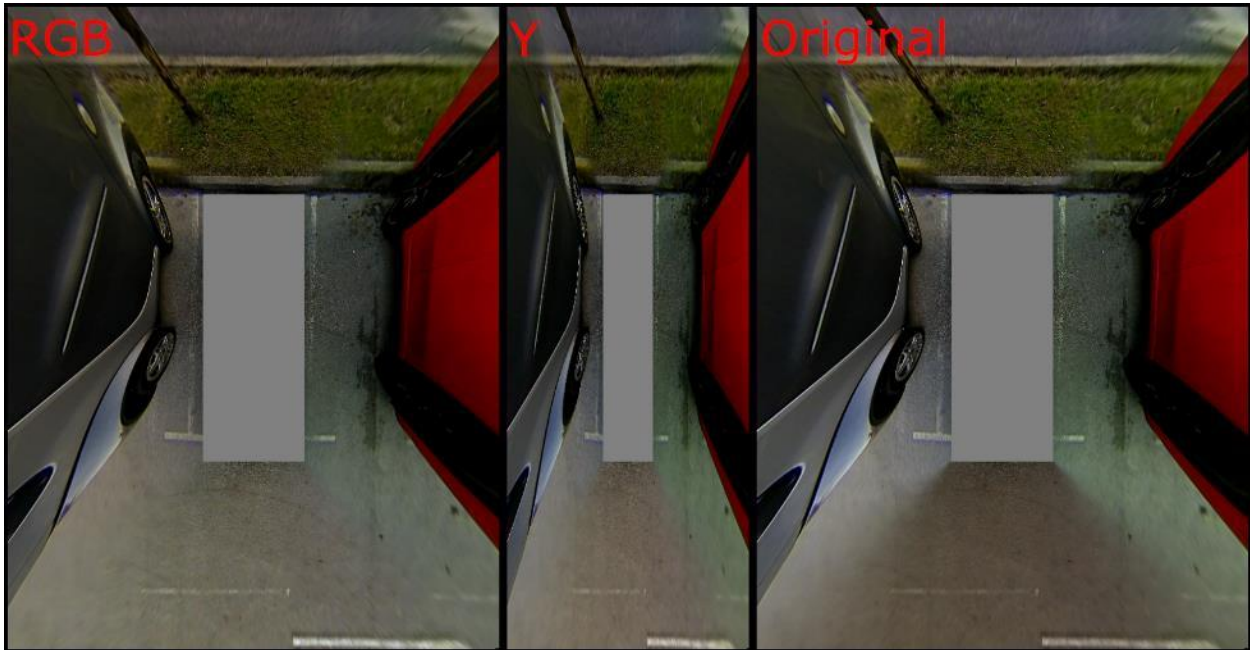
Druga slika su jutarnji uvjeti.



*Slika 6.55 Direktna usporedba slika po jutarnjim uvjetima.*

Jutarnji uvjeti, dok sunce tek izlazi i dosta je intenzivno, možemo vidjeti da RGB i u ovom slučaju izbljeđuje sliku. U nekim dijelovima je također i mutna, pogotovo na rubovima, no u ovom slučaju to izbjeljivanje je dosta poboljšalo vidljivost same slike. Y slika je skoro pa identična originalnoj slici, te nemamo problem sa blijedim bojama.

Treća slika su večernji uvjeti.

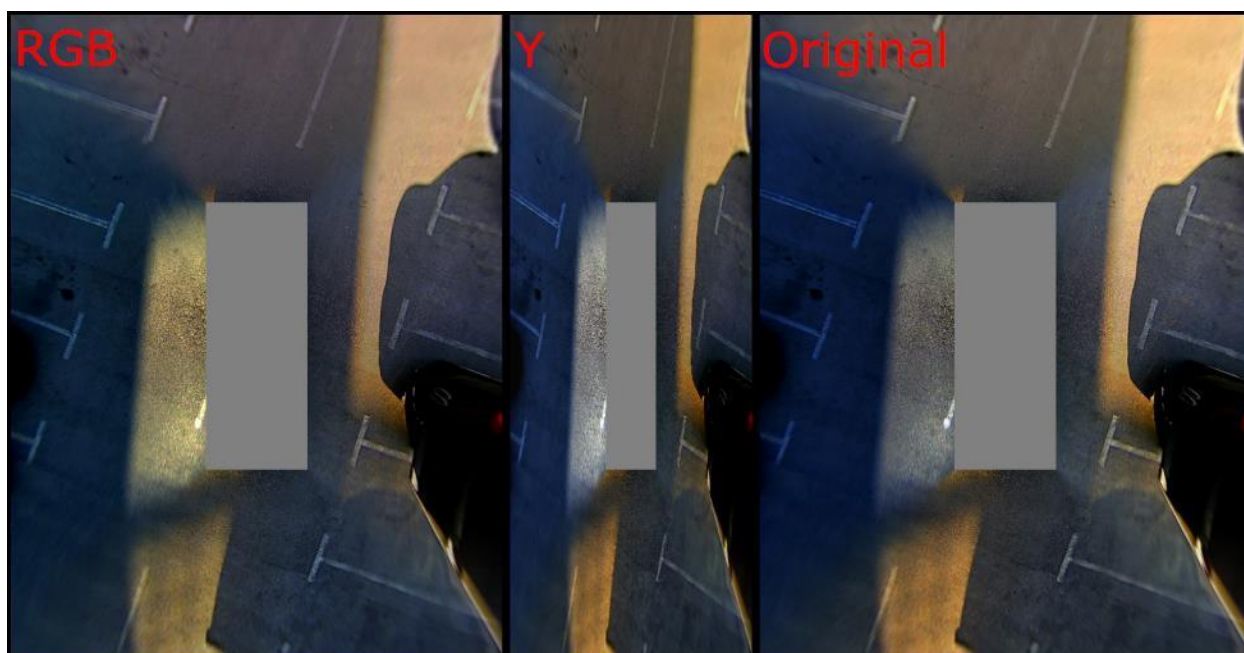


*Slika 6.56 Direktna usporedba slika po večernjim uvjetima.*

U slučaju kod večernjih uvjeta, razlika se najmanje uoči. Samim time što svjetlost više nije intenzivna, možemo vidjeti da kod RGB slike ne izbija toliko plava boja kao kod Y slike. Boje su nešto sive u odnosu na original te također možemo primijetiti da RGB slika nema vidljive sjene dok Y slika ima nešto malo vidljive u odnosu na originalnu sliku.



Zadnja slika, odnosno četvrta, su dnevni uvjeti.



*Slika 6.57 Direktna usporedba slika po dnevnim uvjetima.*

Normalni dnevni uvjeti smatraju se da je vidljivost dobra no ne i intenzivna kao što je kod jutarnjih uvjeta specifično. RGB slika nije zasićena toplom bojom te vidimo izbijanje plavo zelene boje. Svjetlost je nešto slabija. Kod Y slike boje su naglašenije i jasnije u dnevnim uvjetima. Sunčeva svjetlost nije intenzivno narančasta niti izbijaju plave boje oko auta.

## 7. Zaključak

Ovaj rad predstavlja napredni sustav pomoći vozaču temeljen na 3D tehnologiji pogleda okruženja. Sa četiri kamere ribljeg oka koje su montirane na vozilu, provodimo cijeli sustav za pogled okruženja, koji uključuje posebnu kalibraciju, 3D model konstrukcije, mapiranje tekstura i slično. Riješeni program je vrlo učinkovit i može se primijeniti u ugrađenim sustavima. Cijeli sustav može se dobro prilagoditi promjenama u okolini i prebaciti se na bilo koji željeni prikaz u realnom vremenu. Rezultati rada pokazuju da kalibracijski algoritam dobiva preciznije rezultate u odnosu na tradicionalni algoritam. Štoviše, umjeravanje zasnovano na jednoj slici omogućuje da se ovaj sustav ne upotrebljava samo u naprednim sustavima pomoći vozaču, nego i u video nadzorima i aplikacijama gdje je potrebna potražnja prikaza okruženja u realnom vremenu.

ADAS različitih proizvođača koji su opisani polako postaje standardna oprema i sada se proizvode za skuplja vozila ili luksuzna vozila. Iako upotreba ADAS-a uvelike povećava sigurnost na cesti, vozačima je važno naučiti kako se to radi i kako se koristi. Naravno to može predstavljati probleme, no korisnici će vremenom naučiti i početi vjerovati tim sustavima. Važno je da ADAS proizvođači predvide kako se ti sustavi koriste u brojnim situacijama na cesti.

Uz testiranje pojedinih sustava, sve je važnije pripremiti testove za sve sustave koji rade paralelno. Upotreba sve većeg broja senzora i ADAS-a dodatno omogućava primjenu naprednih sustava zaključivanja.

Također omogućuje izgradnju ovih sustava temeljenih na umjetnoj inteligenciji, što je korak prema automatiziranoj vožnji. Napredak se može primijetiti kroz publikacije, natječaje i medije. Pojednostosti o takvim rješenjima obično se dobro čuvaju u tajnosti i malo je takvih podataka dostupno širokoj javnosti.

Također u ovom radu smo predstavili cjeloviti sustav okruženja u stvarnom vremenu koje je spremno za masovno korištenje. Uz to smo naveli i opisali tri glavne komponente za razlučivanje okoline, geometrijsko poravnanje, fotometrijsko poravnanje i kompozitni prikaz sinteze. Opisali smo tehnike koje se koriste za optimizaciju našeg rješenja za performanse u stvarnom vremenu. Predloženim rješenjem postigli smo visokokvalitetni HD video izlaz pri 30 sličica u sekundi.

Iako obećanje automatizirane vožnje privlači i već se naveliko prodaje potrošačima, u istraživanju provedenom za ovaj rad, pokazalo se da postoje i neki nedostaci u ADAS-u. Još uvijek se ne slažu optimalni modaliteti za lokalizaciju, mapiranje i percepciju, algoritmi i dalje nemaju 100%tnu točnost i učinkovitost, a potreba za ispravnom internetskom vezom i procjenom postala je potrebna i očita.

Manje od idealnih uvjeta na cesti još uvijek su problem koji se rješava, kao i rješavanje vremenskih nepogoda. Komunikacija između vozila i vozila još je u povojima, dok je centralizirano upravljanje informacijama utemeljeno na oblaku još uvijek uvedeno zbog složene potrebne infrastrukture. Ljudsko strojna interakcija još je nedovoljno istraženo polje s mnogo otvorenih problema koje treba usavršiti.

Razvoj automatiziranih upravljačkih sustava ovisi o napretku znanstvenih disciplina i novih tehnologija. Kao takve tehnologije, tek ubuduće će vjerojatno imati značajne utjecaje na tehnologiju automatizirane vožnje bilo prevladavanjem slabosti ostalih metoda ili alternativnih.

Ovaj rad i istraživanje koje se napravilo za njegove potrebe, pokazalo se interdisciplinarnim pristupom između korisnika, struke i podrškom autoindustrije mogu se riješiti izazovi ADAS-a. Usmjerenim naporima ka osiguravanju sigurnosti vozača te na svim razinama automatiziranih vozačkih sustava, sigurne, učinkovite i idealne ceste su još uvijek izvan realnih mogućnosti. Možda se u bližoj budućnosti sve to promjeni.

U Varaždinu, srpanj 2020.

Tomislav Levak



IZJAVA O AUTORSTVU  
I  
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, Tomislav Levak (ime i prezime) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog (obrisati nepotrebno) rada pod naslovom Fetometrijska isplodnaost videozapisa iz različitih kama (upisati naslov) te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:  
(upisati ime i prezime)

Tomislav Levak  
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnosti i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, Tomislav Levak (ime i prezime) neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (obrisati nepotrebno) rada pod naslovom Fetometrijska isplodnaost videozapisa iz različitih kama (upisati naslov) čiji sam autor/ica.

Student/ica:  
(upisati ime i prezime)

Tomislav Levak  
(vlastoručni potpis)

## 8. Literatura

Časopisi:

- [1] CC Lin and MS Wang, “A vision based top-view transformation model for a vehicle parking assistant”, *Sensors*, vol. 12, pp. 4431–4446, 2012.
- [2] Yu-Chih Liu, Kai-Ying Lin, and Yong-Sheng Chen, “Birds-eye view vision system for vehicle surrounding monitoring”, in *International Workshop on Robot Vision*, 2008, pp. 207–218.
- [3] Frank Nielsen, “Surround video: a multihead camera approach”, *Vis. Comput.*, vol. 21, pp. 92–103, 2005.
- [4] Din Chang Tseng, Tat Wa Chao, and Jiun Wei Chang, “Image-based parking guiding using ackermann steering geometry”, *Appl. Mech. Mater.*, vol. 437, pp. 823–826, 2013.
- [5] Matthew Brown and David G. Lowe, “Automatic panoramic image stitching using invariant features”, *International Journal of Computer Vision*, vol. 74, pp. 59–73, January 2007.
- [6] P. Koopman and M. Wagner, “Autonomous vehicle safety: An inter-disciplinary challenge”, *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 1, pp. 90–96, Jan. 2017.
- [7] Y. Yuan, D. Wang, and Q. Wang, “Anomaly detection in traffic scenes via spatial-aware motion reconstruction”, *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1198–1209, Mar. 2017.
- [8] B. Zhanget al., “A surround view camera solution for embedded systems”, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 676–681.
- [9] M. Yu and G. Ma, “360° surround view system with parking guidance”, *Driver Assist. Syst.*, vol. 7, no. 1, pp. 19–24, 2014.
- [10] Z. Hu, Y. Li, and Y. Wu, “Radial distortion invariants and lens evaluation under a single-optical-axis omnidirectional camera”, *Comput. Vis. Image Understand.*, vol. 126, no. 2, pp. 11–27, 2014
- [11] C. S. Fraser, “Automatic camera calibration in close range photogrammetry”, *Photogramm. Eng. Remote Sens.*, vol. 79, no. 4, pp. 381–388, 2013.
- [12] Z. Zhang, “A flexible new technique for camera calibration”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.
- [13] Z. Zhang, “Camera calibration with one-dimensional objects”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 7, pp. 892–899, Jul. 2004.
- [14] Q. Wang, C. Zou, Y. Yuan, H. Lu, and P. Yan, “Image registration by normalized mapping”, *Neurocomputing*, vol. 101, pp. 181–189, Feb. 2013.
- [15] E. A. Bier and K. R. Sloan, “Two-part texture mappings”, *IEEE Comput. Graph. Appl.*, vol. 6, no. 9, pp. 40–53, Sep. 1986.
- [16] P. J. Besl, „Geometric modeling and computer vision”, *Proc. IEEE*, vol. 76, no. 8, pp. 936–958, Aug. 1988.
- [17] M. A. Ruzon and C. Tomasi, “Alpha estimation in natural images”, in *Proc. CVPR*, vol. 1, 2000, pp. 18–25.
- [18] A. Levin, D. Lischinski, and Y. Weiss, “A closed-form solution to natural image matting”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 228–242, Feb. 2008.

- [19] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, 'Three Decades of Driver Assistance Systems: Review and Future Perspectives', *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 6–22, 2014.
- [20] A. Ziebinski and S. Swierc, 'Soft Core Processor Generated Based on the Machine Code of the Application', *Journal of Circuits, Systems and Computers*, vol. 25, no. 04, p. 1650029, Apr. 2016.
- [21] J. Han, D. Kim, M. Lee, and M. Sunwoo, 'Road boundary detection and tracking for structured and unstructured roads using a 2D lidar sensor', *International Journal of Automotive Technology*, vol. 15, no. 4, pp. 611–623, Jun. 2014.
- [22] H. G. Jung, Y. H. Lee, H. J. Kang, and J. Kim, 'Sensor fusion-based lane detection for LKS+ACC system', *International Journal of Automotive Technology*, vol. 10, no. 2, pp. 219–228, Apr. 2009.
- [23] J. Arrósipide and L. Salgado, 'On-road visual vehicle tracking using Markov chain Monte Carlo particle filtering with metropolis sampling', *International Journal of Automotive Technology*, vol. 13, no. 6, pp. 955–961, Oct. 2012.
- [24] S. Budzan and J. Kasprzyk, 'Fusion of 3D laser scanner and depth images for obstacle recognition in mobile applications', *Optics and Lasers in Engineering*, vol. 77, pp. 230–240, Feb. 2016.
- [25] M. Błachuta, R. Czyba, W. Janusz, and G. Szafranski, 'Data Fusion Algorithm for the Altitude and Vertical Speed Estimation of the VTOL Platform', *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1–2, pp. 413–420, Apr. 2014.
- [26] A. Barth and U. Franke, 'Estimating the Driving State of Oncoming Vehicles From a Moving Platform Using Stereo Vision', *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 560–571, Dec. 2009.
- [27] D. Pamuła and A. Ziębiński, 'Securing video stream captured in real time', *Przegląd Elektrotechniczny*, vol. R. 86, nr 9, pp. 167–169, 2010.
- [28] A. Ziebinski, R. Cupek, H. Erdogan, and S. Waechter, 'A Survey of ADAS Technologies for the Future Perspective of Sensor Fusion', presented at the International Conference on Computational Collective Intelligence, 2016, pp. 135–146.

Radovi na konferenciji:

- [29] S.T.Y. Suen, E.Y. Lam, and K.K.Y. Wong, "Digital photograph stitching with optimized matching of gradient and curvature", in *Proc. SPIE - Int. Soc. Opt. Eng.*, 2006.
- [30] M. Uyttendaele, A. Eden, and R. Skeliski, "Eliminating ghosting and exposure artifacts in image mosaics", in *CVPR 2001*, 2001.
- [31] Seung-Ryong Han, Jongsul Min, Taesung Park, and Yongje Kim, "Photometric and geometric rectification for stereoscopic images", in *Proc. SPIE - Int. Soc. Opt. Eng.*, 2012.
- [32] M. Lin, G. Xu, X. Ren and K. Xu, "Cylindrical panoramic image stitching method based on multi-cameras", in *Proc. IEEE Int. Conf. Cyber Technol. Autom., Control, Intell. Syst.*, Jun. 2015, pp. 1091–1096.
- [33] M. Salvi and K. Vaidyanathan, "Multi-layer alpha blending", in *Proc. Meet. ACM SIGGRAPH Symp. Interact. 3D Graph. Games*, 2014, pp. 151–158.

Knjige:

- [34] Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [35] Richard L. Burden and J. Douglas Faires, *Numerical Analysis*, Brooks/Cole, 2004.
- [36] “360° wrap-around video imaging technology ready for integration with fujitsu graphics SoCs”, Fujitsu Microelectron. America, Inc., Sunnyvale, CA, USA, Tech. Rep., Feb. 2011. [Online].  
Raspoloživo: [https://www.fujitsu.com/us/Images/360\\_OmniView\\_AppNote.pdf](https://www.fujitsu.com/us/Images/360_OmniView_AppNote.pdf)
- [37] M. Schönbein, T. Strauß, and A. Geiger, “Calibrating and centering quasicentral catadioptric cameras”, in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 4443–4450.
- [38] H.-T. Chen, “Geometry-based camera calibration using five point correspondences from a single image”, *IEEE Trans. Circuits Syst. Video Technol.*, to be published.
- [39] T. Stathaki, *Image Fusion: Algorithms and Applications*. San Francisco, CA, USA: Academic, 2008.

## Popis slika

Slika 1.1 Automobilski sustav za prikaz okruženja. U ovom primjeru prikazane su četiri širokokutne kamere (koje predstavljaju riblje oko) oko automobila, te je svaka od kamera okrenuta u drugom smjeru kako bi pokrila različite kutove. ....	2
Slika 1.2 Ilustracija sustava za prikaza okruženja sa i bez fotometrijskog poravnanja. ....	3
Slika 2.1 Slika preklapajućeg područja i presjeka blokova. Pod pretpostavkom da je geometrijsko poravnanje već izvedeno na svakom je prikazu $\Omega$ skup piksela iz preklapajućeg područja. $\Omega_{m,n}$ sadrži skup piksela iz preklapajućeg područja. $\Omega_{m,n}$ sadrži piksele iz prikaza. ....	5
Slika 2.2 Ilustracija krivulje tona preslikavanja i sidrenih točaka. ....	8
Slika 3.1 Izlazni rezultat postojećeg sustava. Lijevo je 2D prikaz, dok je desno 2D prikaz sa spojenim slikama iz svih kutnih kamera. ....	11
Slika 3.2 Slika zaslona sustava. ....	11
Slika 3.3 Prikaz sustava unutar vozila. ....	12
Slika 3.4 Prikazuje dijagram ključnih koraka za dobivanje slike okruženja. ....	13
Slika 3.5 Prikaz modela Broda. ....	13
Slika 3.6 Pogled u presjeku 3D modela broda. ....	14
Slika 3.7 Prikaz konstrukcije modela. ....	14
Slika 3.8 Specijalna kalibracijska ploča (označena crvenim linijama). ....	16
Slika 3.9 Perspektiva transformacije procesa. [6] ....	17
Slika 3.10 Proces transformacije koordinata. ....	20
Slika 3.11 Mapiranje u dva koraka. ....	21
Slika 3.12 O preslikavanje. ....	22
Slika 3.13 Projekcija modela u prvom kvadrantu. ....	23
Slika 3.14 Usporedba dva algoritma, (a) je tradicionalan algoritam a (b) je predloženi algoritam. ....	24
Slika 3.15 Prikaz slika uzetih s lećama ribljih očiju. (a) prednji pogled, (b) stražnji pogled, (c) desni pogled i (d) lijevi pogled ....	25
Slika 3.16 Prikaz okruženja u podzemnoj garaži. (a) prednji pogled, (b) stražnji pogled, (c) desni pogled i (d) lijevi pogled. ....	25
Slika 3.17 Usporedba s drugim sustavima. (a) Fujitsu sustav, (b) Delphi sustav i (c) moderni sustav. ....	26
Slika 5.1 KKT uvjeti. ....	34
Slika 5.2 Ulazna slika, pogled od svih četiri kamera. ....	40
Slika 5.3 Izlazna slika, dobivena našim algoritmom. ....	40



Slika 6.1 RGB rezultat 1. videozapisa, normalni dnevni uvjeti. ....	41
Slika 6.2 RGB rezultat 2. videozapisa, noćni uvjeti. ....	42
Slika 6.3 RGB rezultat 3. videozapisa, jutarnji uvjet. ....	42
Slika 6.4 RGB rezultat 4. videozapisa, uvjeti pred zalazak sunca. ....	43
Slika 6.5 Y rezultat 1. videozapisa, normalni dnevni uvjeti. ....	44
Slika 6.6 Y rezultat 2. videozapisa noćni uvjeti . ....	45
Slika 6.7 Y rezultat 3. videozapisa, jutarnji uvjet. ....	45
Slika 6.8 Y rezultat 4. videozapisa, uvjeti pred zalazak sunca. ....	46
Slika 6.9 RGB rezultat 1. videozapisa. ....	47
Slika 6.10 RGB rezultat 2. videozapisa. ....	48
Slika 6.11 RGB rezultat 3. videozapisa. ....	48
Slika 6.12 RGB rezultat 4. videozapisa. ....	49
Slika 6.13 RGB rezultat 1. videozapisa. ....	49
Slika 6.14 RGB rezultat 2. videozapisa. ....	50
Slika 6.15 RGB rezultat 3. videozapisa. ....	50
Slika 6.16 RGB rezultat 4. videozapisa. ....	51
Slika 6.17 RGB rezultat 1. videzapisa. ....	52
Slika 6.18 RGB rezultat 2. videozapisa. ....	52
Slika 6.19 RGB rezultat 3. videozapisa. ....	53
Slika 6.20 RGB rezultat 4. videozapisa. ....	53
Slika 6.21 Y rezultat 1. videozapisa. ....	54
Slika 6.22 Rezultat Y rezultat 2. videozapisa. ....	55
Slika 6.23 Y rezultat 3. videozapisa. ....	55
Slika 6.24 Y rezultat 4. videozapisa. ....	56
Slika 6.25 Y rezulztat 1. videozapisa. ....	57
Slika 6.26 Y rezultat 2. videozapisa. ....	58
Slika 6.27 Y rezultat 3. videozapisa. ....	58
Slika 6.28 Y rezultat 4. videozapisa. ....	59
Slika 6.29 Y rezultat 1. videozapisa. ....	59
Slika 6.30 Y rezultat 2. videozapisa. ....	60
Slika 6.31 Y rezultat 3. videozapisa. ....	60
Slika 6.32 Y rezultat 4. videozapisa. ....	61
Slika 6.33 RGB rezultat 1. videozapisa. ....	62
Slika 6.34 RGB rezultat 2. videozapisa. ....	62
Slika 6.35 RGB rezultat 3. videozapisa. ....	63

Slika 6.36 RGB rezultat 4. videozapisa. ....	63
Slika 6.37 Y rezultat 1. videozapisa. ....	64
Slika 6.38 Y rezultat 2. videozapisa. ....	65
Slika 6.39 Y rezultat 3. videozapisa. ....	65
Slika 6.40 Y rezultat 4. videozapisa. ....	66
Slika 6.41 RGB rezultat 1. videozapisa. ....	67
Slika 6.42 YCbCr rezultat 1. videozapisa. ....	67
Slika 6.43 RGB rezultat 2. videozapisa. ....	68
Slika 6.44 YCbCr rezultat 2. videozapisa. ....	68
Slika 6.45 RGB rezultat 3. videozapisa. ....	69
Slika 6.46 YCbCr rezultat 3. videozapisa. ....	69
Slika 6.47 RGB rezultat 4. videozapisa. ....	70
Slika 6.48 YCbCr rezultat 4. videozapisa. ....	70
Slika 6.49 Bez izjednačavanja, dnevni uvjeti. ....	72
Slika 6.50 Bez izjednačavanja, noćni uvjeti. ....	72
Slika 6.51 Bez izjednačavanja, jutarnji uvjeti. ....	73
Slika 6.52 Bez izjednačavanja, večernji uvjeti. ....	73
Slika 6.53 YCbCr po RGB. ....	75
Slika 6.54 Direktna usporedba slika po noćnim uvjetima. ....	76
Slika 6.55 Direktna usporedba slika po jutarnjim uvjetima. ....	76
Slika 6.56 Direktna usporedba slika po večernjim uvjetima. ....	77
Slika 6.57 Direktna usporedba slika po dnevnim uvjetima. ....	78