

# Programska podrška za snimanje video sadržaja za verifikaciju ADAS algoritama na strani PC računala

---

**Benke, Ivan**

**Master's thesis / Diplomski rad**

**2018**

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:167344>

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

*Download date / Datum preuzimanja:* **2025-02-26**

*Repository / Repozitorij:*

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



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I**

**INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni studij**

**PROGRAMSKA PODRŠKA ZA SNIMANJE VIDEO  
SADRŽAJA ZA VERIFIKACIJU ADAS ALGORITAMA  
NA STRANI PC RAČUNALA**

**Diplomski rad**

**Ivan Benke**

**Osijek, 2018.**

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

Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada

Osijek, 22.09.2018.

Odboru za završne i diplomske ispite

**Imenovanje Povjerenstva za obranu diplomskog rada**

<b>Ime i prezime studenta:</b>	Ivan Benke
<b>Studij, smjer:</b>	Diplomski sveučilišni studij Računarstvo
<b>Mat. br. studenta, godina upisa:</b>	D 751 R, 22.09.2017.
<b>OIB studenta:</b>	05394896456
<b>Mentor:</b>	Doc.dr.sc. Ratko Grbić
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	Matija Pul
<b>Predsjednik Povjerenstva:</b>	Izv. prof. dr. sc. Marijan Herceg
<b>Član Povjerenstva:</b>	Izv. prof. dr. sc. Mario Vranješ
<b>Naslov diplomskog rada:</b>	Programska podrška za snimanje video sadržaja za verifikaciju ADAS algoritama na strani PC računala
<b>Znanstvena grana rada:</b>	<b>Informacijski sustavi (zn. polje računarstvo)</b>
<b>Zadatak diplomskog rada:</b>	ADAS (engl. Advanced Driver Assistance System) algoritmi se sve više testiraju u laboratorijima s unaprijed snimljenim podacima (uključujući video podatke). U tu svrhu se razvijaju uređaji koji podatke snimaju s kamera i dostavljaju do PC računala koje ih skladište na disk. U okviru ovog zadatka potrebno je napraviti programsku podršku za snimanje video sadržaja na strani PC računala koja komunicira i dobavlja sadržaj s namjenskog uređaja za snimanje. Potrebno je realizirati protokol za komunikaciju preko PCIe sabirnice između PC računala i uređaja za snimanje video sadržaja - AMV Grabber. Ova komponenta treba podržavati kontrolne poruke, slanje zahtjeva za video sadržajem kao i prihvatanje video sadržaja preko PCIe sabirnice i njegovo smještavanje na disk u
<b>Prijedlog ocjene pismenog dijela ispita (diplomskog rada):</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene mentora:</b>	22.09.2018.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

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

Osijek, 24.09.2018.

**Ime i prezime studenta:**

Ivan Benke

**Studij:**

Diplomski sveučilišni studij Računarstvo

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

D 751 R, 22.09.2017.

**Ephorus podudaranje [%]:**

2

Ovom izjavom izjavljujem da je rad pod nazivom: **Programska podrška za snimanje video sadržaja za verifikaciju ADAS algoritama na strani PC računala**

izrađen pod vodstvom mentora Doc.dr.sc. Ratko Grbić

i sumentora

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

Potpis studenta:

# SADRŽAJ

1. UVOD .....	1
2. AUTONOMNA VOŽNJA .....	3
2.1. ADAS algoritmi.....	3
2.2. Pregled postojećih rješenja za snimanje video sadržaja za kasniju primjenu.....	5
3. PROJEKTIRANJE I IMPLEMENTACIJA PROGRAMSKE PODRŠKE ZA SNIMANJE VIDEO SADRŽAJA .....	8
3.1. Opis korištene hardverske platforme .....	8
3.2. Zahtjevi za izradu vlastitog rješenja .....	9
3.3. Korištene tehnologije i alati.....	10
3.4. Koncept rješenja .....	11
3.5. Implementacija rješenja .....	15
4. EVALUACIJA VLASTITOG RJEŠENJA.....	18
4.1. Brzina prijenosa preko PCIe sabirnice .....	18
4.2. Brzina zapisivanja na disk .....	20
4.3. Problemi prilikom rada aplikacije .....	24
5. ZAKLJUČAK .....	27
LITERATURA.....	28
SAŽETAK.....	30
ABSTRACT .....	31
ŽIVOTOPIS .....	32

## 1. UVOD

Razvojem automobilske industrije došlo je do potrebe za razvojem potpuno autonomnih vozila kojih je na cestama sve više. Takva vozila sadrže ugrađena računala koja primaju podatke sa senzora i kamera postavljenih po cijelom vozilu te na temelju tih podataka donose odluke. Kako bi ugrađena računala donosila ispravne odluke potrebno je izraditi odgovarajuće računalne algoritme te ih na temelju njihovih izlaza naučiti kako reagirati u određenim situacijama tijekom vožnje. Za proces učenja koriste se ADAS (engl. *Advanced Driver Assistance Systems*) algoritmi koji se najčešće temelje na metodama strojnog učenja. Kako bi ADAS algoritmi imali vrlo visoku točnost i pouzdanost u stvarnim uvjetima rada, potrebno ih je prvo istrenirati. Treniranje se vrši tako da im se zada skup podataka, najčešće video sadržaj koji prikazuje stvarne situacije u prometu. Sličan video sadržaj koristi se onda i u verifikaciji ADAS algoritama.

Budući da su rizici i troškovi treniranja ADAS algoritama veliki jer većinom zahtijevaju video sadržaj koji se može snimiti samo u stvarnim uvjetima u prometu, razvila se potreba za razvojem platforme koja omogućuje snimanje i spremanje video sadržaja za kasniju primjenu.

Zadatak diplomskog rada izrada je programske podrške za snimanje video sadržaja na strani računala koja komunicira i dobavlja sadržaj s namjenskog uređaja za snimanje – AMV *Grabber* (engl. *Automotive Machine Vision Grabber*). AMV *Grabber* uređaj predstavlja više-kanalni video uređaj za istovremeno preuzimanje i reprodukciju video sadržaja. Omogućuje prikupljanje i reprodukciju video signala s devet kamera koje su raspoređene po vozilu. Također podržava prijenos podataka na računalo preko PCIe (engl. *Peripheral Component Interconnect Express*) sabirnice za spremanje na trajnu memoriju računala. Spremljeni podaci, odnosno video sadržaj, se zatim šalje s računala preko AMV *Grabber* uređaja na uređaje čija je namjena treniranje i verificiranje ADAS algoritama.

Diplomskim će radom biti objašnjen koncept vlastitog rješenja na problem pohrane video sadržaja za kasniju primjenu. Također će biti opisana implementacija vlastitog rješenja i provedena mjerenja koja dokazuju koliko je rješenje uspješno implementirano korištenjem AMV *Grabber* uređaja.

Za uspješnu izradu diplomskog rada potrebno je realizirati protokol za komunikaciju računala s namjenskim uređajem preko PCIe sabirnice. Protokol treba podržavati kontrolne poruke, slanje zahtjeva za video sadržajem te prihvatanje video sadržaja preko PCIe sabirnice i njegovo smještanje na disk u stvarnom vremenu za podesiv broj kamera (do devet kamera).

Budući da je ovaj diplomski rad dio jedne veće cjeline, tj. on je jedan modul unutar projekta *AMV Grabber*, za uspješnu izradu modula, odnosno diplomskog rada, potrebno je definirati i vlastitu aplikativnu programsku spregu (engl. *Application Programming Interface* – API). Svrha API-ja je da izrađene funkcionalnosti budu dostupne i ostalim modulima unutar projekta. Također, nakon izrade i verifikacije rada modula, potrebno je navedeni modul optimizirati, odnosno poboljšati mu performanse kako bi se zadovoljili uvjeti za rad u stvarnom vremenu.

U drugom poglavlju dan je uvod u nastojanja automobilske industrije za napretkom što dovodi do stvaranja novih tehnologija, kao i pregled već dostupnih rješenja na problem prijenosa video sadržaja na računalo. Trećim poglavljem opisani su korišteni alati i tehnologije, korištena hardverska platforma te koncept vlastitog rješenja i njegova implementacija. U četvrtom poglavlju provedena je evaluacija rješenja u obliku mjerenja brzine prijenosa podataka na računalo i brzine zapisivanja podataka na disk. Zatim je dana diskusija rezultata i analizirani su problemi koji su se javili prilikom testiranja rješenja. U završnom, petom, poglavlju dan je zaključak i pregled svega ostvarenog u diplomskom radu.

## 2. AUTONOMNA VOŽNJA

Automobilska industrija jedna je od najznačajnijih industrija današnjice. Uz vojnu, automobilska industrija ima najveći utjecaj na razvoj tehnologija koje, uz manje ili veće preobrazbe, naposljetku budu implementirane u uređaje namijenjene običnim korisnicima. Kako bi ostala jedna od vodećih industrija, automobilska industrija mnogo sredstava ulaže u postupak istraživanja i razvoja, tzv. R'n'D (engl. *Research and Development*). Istraživanjem se došlo do zaključka da je sljedeći korak u razvoju automobilske industrije stvaranje potpuno autonomnog vozila namijenjenog običnom korisniku [1]. Takvo bi vozilo bilo potpuno kontrolirano ugrađenim računalom koje bi donosilo odluke umjesto vozača, tj. vozač više ne bi imao utjecaja na vozilo tijekom vožnje.

### 2.1. ADAS algoritmi

Autonomnost vozila, prema SAE *International* (engl. *Society of Automotive Engineers International*), podijeljena je na 6 razina [2]. Kao što je prikazano tablicom 2.1., razine su u rasponu od nulte do pete. Razina 0 omogućuje vozilima davanje upozorenja vozaču, ali nema kontrolu nad vozilom. Razinom 1 automatizirani sustav i vozač dijele kontrolu nad vozilom što je vidljivo iz primjera pomoći u parkiranju gdje sustav preuzima kontrolu nad upravljanjem vozilom dok vozač ima kontrolu nad brzinom vozila. Razina 2 prepušta potpunu kontrolu vozila automatiziranom sustavu (ubrzavanje, kočenje, upravljanje), ali vozač mora biti spreman na intervenciju u bilo kojem trenutku ako sustav ne uspije reagirati pravovremeno. Razina 3 omogućuje vozaču da posveti pozornost nečemu drugome tijekom vožnje, npr. razgovor sa suputnicima, a sustav će reagirati u situacijama kada je potrebna intervencija. Vozač i dalje mora biti spreman na intervenciju unutar ograničenog vremenskog perioda. Razina 4 ista je kao i razina 3, ali pozornost vozača uopće nije potrebna, što znači da vozač slobodno može zaspati ili se maknuti s vozačeva sjedišta. Autonomna vožnja je podržana samo na određenim prostorima ili situacijama, poput gužvi u prometu, a vozač može preuzeti kontrolu nad vozilom kad poželi. Razina 5 omogućuje sustavu potpunu kontrolu nad vozilom što znači da intervencija vozača nije nikad potrebna, tj. više ne postoji vozač, nego svi pojedinci u vozilu postaju putnici. Volan za upravljanje vozilom može biti ugrađen, ali nije potreban [3].



**Tab. 2.1.** Razine autonomnosti vozila prema SAE International [2].

RAZINA	OPIS
0	Vozilo nema nikakvu autonomnost, vozač izvršava sve zadatke vezane uz vožnju.
1	Vozač upravlja vozilom, ali neka svojstva autonomnosti mogu biti dio dizajna vozila.
2	Vozilo posjeduje kombinirane automatizirane funkcije poput upravljanja i ubrzavanja, ali vozač mora ostati prisutan u procesu vožnje.
3	Vozač je nužan, ali nije potrebno da motri okolinu. Vozač mora biti spreman preuzeti kontrolu nad vozilom u bilo kojem trenutku.
4	Vozilo je sposobno obavljati sve zadatke vezane uz vožnju u određenim uvjetima. Vozač može imati opciju da kontrolira vozilo.
5	Vozilo je sposobno obavljati sve zadatke vezane uz vožnje u svim uvjetima. Vozač može imati opciju da kontrolira vozilo.

U novije vrijeme, pojavio se trend razvoja ADAS algoritama. To su algoritmi namijenjeni za pomaganje vozačima tijekom vožnje. Primjena ADAS algoritama je raznovrsna, od pomoći vozačima u parkiranju do praćenja stanja vozača tijekom vožnje. ADAS algoritme prvenstveno se želi iskoristiti kako bi se omogućila autonomna vožnja, odnosno vožnja u kojoj vozači ne bi upravljali vozilom. Umjesto vozača, računalo u vozilu (engl. *Electronic Control Unit* – ECU) bi primalo podatke sa senzora postavljenih po vozilu, uključujući kamere, GPS, žiroskope i slično. Računalo bi zatim te podatke obrađivalo i pretvaralo u korisne informacije koje su potrebne ADAS algoritmima za donošenje odluka. Na temelju donesenih odluka, računalo bi dalo naredbu vozilu, tj. upravljalo bi njime bez potrebe za reakcijom vozača [3, 4].

Budući da je za razvoj i treniranje ADAS algoritama potreban nekakav skup podataka, najčešće skup slika ili video sadržaj koji prikazuju vožnju vozilom, javlja se potreba za razvijanjem nove tehnologije koja omogućuje snimanje, tj. prikupljanje takvih skupova podataka unaprijed. Osim što unaprijed prikupljeni skupovi podataka omogućuju lakši razvoj i testiranje algoritama, također smanjuju troškove, i vremenske i financijske, jer se ne mora svaki put voziti vozilom kada se želi istrenirati algoritme.

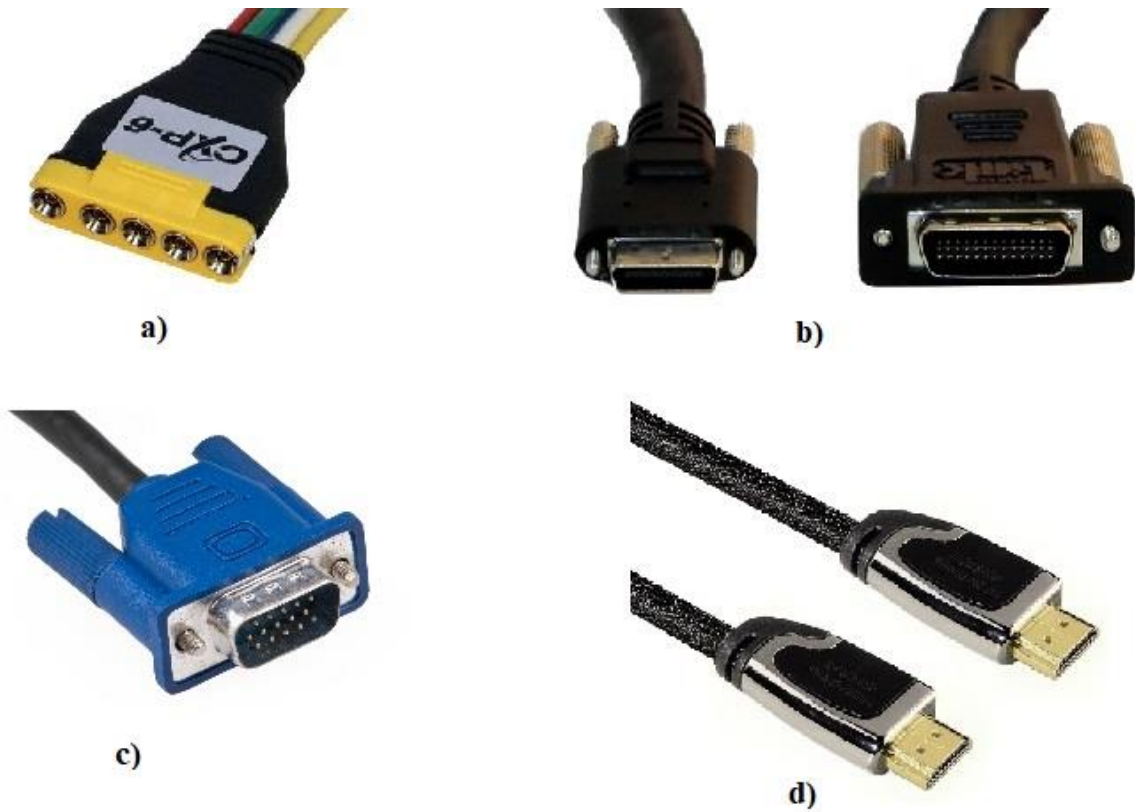
## 2.2. Pregled postojećih rješenja za snimanje video sadržaja za kasniju primjenu

Iz prethodnog potpoglavlja vidljivo je da razvoj ADAS algoritama sa sobom povlači i razvoj niza drugih tehnologija potrebnih za lakše i brže razvijanje i testiranje ADAS algoritama. Budući da je za većinu ADAS algoritama potreban skup vizualnih podataka (slike ili video sadržaj) razvila se ideja o izradi hardvera s odgovarajućom softverskom podrškom koji omogućava snimanje video sadržaja kamerama postavljenima po cijelom vozilu. Taj video sadržaj zatim bi se trebao moći spremiti na trajnu memoriju računala za kasniju primjenu u razvijanju i testiranju ADAS algoritama.

Pretraživanjem web stranica tvrtki koje se bave razvojem i proizvodnjom hardvera došlo se do saznanja da razvoj namjenskog hardvera za snimanje i spremanje video sadržaja za kasniju primjenu (engl. *Frame grabber*) nije nikakva novost. Međutim, već razvijeni uređaji se razlikuju od tvrtke do tvrtke pa su neki namijenjeni komercijalnoj primjeni, a neki industrijskoj.

Tako npr. tvrtka Epiphan [5] proizvodi mnoštvo uređaja namijenjenih prijenosu video sadržaja s uređaja za snimanje. Neki njihovi proizvodi koriste PCIe sabirnicu, a neki USB sučelje za prijenos video sadržaja na računalo. Većina njihovih proizvoda namijenjena je komercijalnim korisnicima, odnosno profesionalcima i tvrtkama koji se bave audiovizualnim poslovima (snimanje filmova, serija, Youtube videa, ...).

S druge strane, tvrtka Phase 1 Technology [6] nudi namjenski hardver koji većinom koristi PCIe sabirnicu za prijenos video sadržaja na računalo. Proizvodi te tvrtke namijenjeni su prijenosu video sadržaja s raznih vrsta kamera. Preneseni podaci namijenjeni su za industrijsku primjenu, tj. za primjenu u znanstvenim istraživanjima i obrađivanju podataka i izvlačenju korisnih informacija iz tih podataka. Mnoštvo različitih ulaznih priključaka dokazuje da im je primjena najčešće u industrijskim poslovima.



**Sl. 2.1.** Vrste priključaka: a) CoaXPress [7], b) Camera Link [8], c) VGA [9], d) HDMI [10]

Ono u čemu se proizvodi za snimanje i spremanje video sadržaja za kasniju primjenu obično razlikuju su ulazni priključci za primanje video sadržaja. Koriste se razne vrste priključaka, kao što je vidljivo na slici 2.2., od komercijalnih koji se mogu naći u svim trgovinama informatičke opreme poput DVI, HDMI, VGA, do onih koji su napravljeni za potrebe određenih tehnologija i standarda kao što su HS Link, CoaXPress, HD-SDI, Camera Link HS, Camera Link i dr.

**Tab. 2.2.** Značajke postojećih rješenja.

ZNAČAJKE	AV.io HD [11]	DVI2PCIe Duo [12]	AS-FBD- 4XCXP6- 2PE8 [13]	ASFBD2X CLM-2PE8 [14]	Xtium- CLHS PX4 [15]
Cijena	~430\$	~1290\$	Na upit	Na upit	Na upit
Programska podrška	Da	Da	Da	Da	Da
Programska podrška – cijena	Besplatno, na upit	Besplatno, na upit	Na upit	Na upit	Na upit
Međuspremnik – veličina	Nema	128 MB	512 MB	512 MB	1024 MB
Ulazni priključci – broj	1	2	4	2	1
Ulazni priključci – vrsta	DVI	DVI, SDI	CoaXPress	Camera Link	Camera Link HS
Izlazni priključci – broj	0	0	0	0	0
Izlazni priključci – vrsta	Nema	Nema	Nema	Nema	Nema
Tip sabirnice	USB 3.0	PCIe Gen2.0 x4	PCIe Gen2.0 x8	PCIe Gen2.0 x8	PCIe Gen2.0 x4

Iz tablice 2.2. vidljivo je da je cijena visoka kod rješenja kod kojih je poznata, a kod rješenja za čiju cijenu se mora poslati upit vjerojatno je još i veća. Osim plaćanja početne cijene za hardver, mora se platiti i za programsku podršku koja omogućava rad s hardverom što je još jedan nedostatak korištenja postojećeg rješenja. Također, sva prikazana postojeća rješenja uopće nemaju izlazne priključke koji su potrebni za slanje video sadržaja s računala na namjenske uređaje za testiranje ADAS algoritama. Zbog svih navedenih razloga razvijena je AMV *Grabber* hardverska ploča koja će biti opisana u sljedećem poglavlju i za koju treba biti razvijena programska podrška.

### 3. PROJEKTIRANJE I IMPLEMENTACIJA PROGRAMSKE PODRŠKE ZA SNIMANJE VIDEO SADRŽAJA

#### 3.1. Opis korištene hardverske platforme

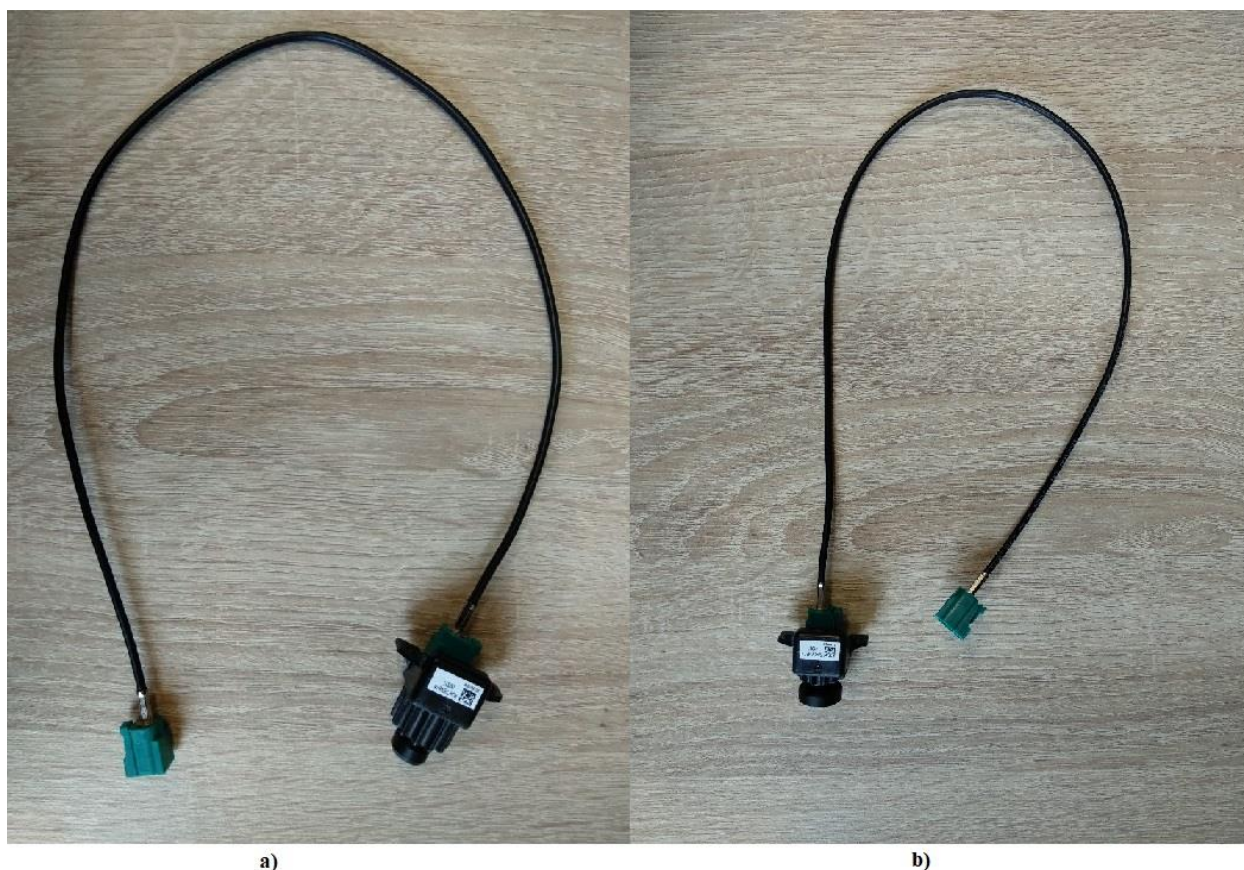
Raspoloživi hardver u obliku je ploče s namjenskim procesorom za obrađivanje podataka pristiglih s kamera, te sa zasebnom RAM (engl. *Random Access Memory*) memorijom koja služi za privremeno pohranjivanje određene količine video sadržaja prije slanja na trajnu memoriju računala. Podaci iz RAM memorije hardverske ploče trebaju se slati preko PCIe sabirnice na trajnu memoriju računala, a i u suprotnom smjeru, odnosno s računala na RAM memoriju hardverske ploče. Također, hardverska ploča ima ulaze namijenjene primanju video sadržaja s kamera, te izlaze namijenjene slanju video sadržaja s računala, preko hardverske ploče, do uređaja na kojima se testiraju ADAS algoritmi. Ulazi i izlazi s hardverske ploče koriste koaksijalni kabel za prijenos podataka s kamera na ploču, tj. na uređaje na kojima se testiraju ADAS algoritmi.



Sl. 3.1. AMV Grabber hardverska ploča.

Hardverska ploča, kao što je vidljivo na slici 3.1., je proizvedena prije početka rada na diplomskom radovima. Korištena je za razvoj i testiranje softvera kojim se omogućuje cijeli proces snimanja i prijenosa video sadržaja na računalo. Proizvela ju je tvrtka RT-RK.





**Sl. 3.2.** Kamere s različitim lećama: a) ravna leća, b) *fish-eye* leća

Također, na slici 3.2. prikazane su kamere koje se koriste za snimanje video sadržaja. Kamere na slici imaju isti senzor naziva OV10640 [16], ali različite leće pa tako prva kamera ima ravnu leću, a druga kamera ima *fish-eye* leću.

### **3.2. Zahtjevi za izradu vlastitog rješenja**

Proučavanjem postojećih rješenja uočeno je da većina koristi PCIe standard za prijenos podataka s hardverske ploče na trajnu memoriju računala. Primjena PCIe standarda korisna je jer omogućuje prijenos velike količine podataka u kratkom vremenu. U ovom slučaju potrebno je prenijeti podatke s devet kamera koje podržavaju najveću rezoluciju od  $1280 \times 1080$  (širina  $\times$  visina) elemenata slike, s time da svaki element slike sadrži 2 B podataka. Također, potrebno je prenijeti 30 okvira (engl. *frame*) po sekundi po kameri te se množenjem svih navedenih podataka dolazi do 746496000 B, što je cca. 0,7 GB koji se trebaju prenijeti svake sekunde. Stoga, odlučeno

je da će vlastito rješenje koristiti PCIe Gen2.0 s 4 trake (x4). Ta konfiguracija omogućuje teorijski prijenos podataka od 2 GB/s što je dovoljno za prijenos podataka sa svih devet kamera.

Rješenje treba biti izvedeno u C programskom jeziku i Windows radnom okruženju. Kako bi bilo moguće prenijeti podatke s AMV *Grabber* ploče na računalo potrebno je da moduli zaduženi za snimanje video sadržaja s kamera u RAM memoriju ploče i upravljački program budu funkcionalni. Rezolucije koje kamere podržavaju su  $1280 \times 720$ ,  $1280 \times 800$  i  $1280 \times 1080$ , a svaki je element slike veličine 2 B. Kamere trebaju snimati 30 okvira po sekundi. Povezane su s pločom koaksijalnim kabelom koji koristi FPD-Link III sučelje.

### 3.3. Korištene tehnologije i alati

Standard korišten za prijenos podataka s hardverske ploče na trajnu memoriju računala je PCIe. AMV *Grabber* hardverska ploča podržava PCIe druge generacije te ima priključak za PCIe sabirnicu sa četiri trake (x4) što znači da je najveća podržana brzina prijenosa podataka 2 GB/s. PCIe standard razvijen je 2004. godine u obliku sabirnice koja podržava periferne uređaje namijenjene izvršavanju zadataka poput izvođenja kalkulacija, pohrani podataka na računalo, komuniciranja s drugim računalima i ostalo. Dio je treće generacije ulazno-izlaznih sabirnica velike brzine. Može imati više konfiguracija koje ovise o broju korištenih traka pa konfiguracija tako može biti x1, x2, x4, x8 i x16. PCIe standard preuzeo je neke funkcije i karakteristike iz prijašnjih generacija poput pisanja i čitanja direktno u/iz memorije, pisanja na izlaze i čitanja s ulaza, isti model memorije, ulaza/izlaza te adresnog prostora kao prethodna generacija ulazno-izlaznih sabirnica, direktnu komunikaciju između perifernih uređaja, itd. Softverski je kompatibilan sa sustavima koji podržavaju sabirnice prethodne generacije, specifično PCI i PCI-X. Najnovija verzija, odnosno generacija, PCIe sabirnice je PCIe 5.0, iako još nije dobila potpuno izdanje. Njezina je maksimalna teorijska brzina 32 GT/s po traci, tj. 63 GB/s ako se koristi svih 16 traka (x16) u sabirnici [17, 18].

Protokol korišten za slanje i primanje zahtjeva i kontrolnih poruka te za prijenos podataka na računalo naziva se IOCP (engl. *Input/Output Completion Port*). To je API koji omogućuje mnoštvo istovremenih asinkronih ulazno/izlaznih operacija u Windows operacijskim sustavima. Prvobitno je stvoren za rad na serverima jer istovremeni model rada servera puno vremena potroši na mijenjanje konteksta niti. Zbog učestalog mijenjanja konteksta nijedna nit ne dobije dovoljno vremena da izvrši svoje operacije. Kako bi se zaobišao taj problem, na početku izvođenja procesa

stvori se „bazen“ niti (engl. *pool of threads*). „Bazen“ niti sadrži najviše dvaput više niti nego što ima procesorskih jezgri. Niti u „bazenu“ su u stanju spavanja te se, kada je potrebno, bude i izvršavaju operaciju koja im je dodijeljena. Nakon izvršene operacije vraćaju se u stanje spavanja. Ako broj operacija na redu koje se trebaju izvršiti bude veći nego broj dostupnih niti, operacije čekaju dok se niti ne oslobode, odnosno dok niti ne izvrše operacije koje su se otprije nalazile na redu. U slučaju da operacije dolaze jedna nakon druge, tj. da nijedna operacije ne dođe dok se druga izvršava, uvijek će se koristiti ista nit, a ostale niti će ostati u stanju spavanja. Takav LIFO (engl. *Last-In-First-Out*) algoritam omogućuje da memorijski resursi niti koje nisu korištene budu vraćeni na disk te da se te niti uklone iz predmemorije (engl. *cache*) procesora. IOCP API dopušta jednoj niti da zatraži ulazno/izlazni zahtjev, a drugoj niti da ga obradi. Kada je IOCP objekt stvoren, povezuje se sa *socket*-om ili datotekom. Nakon što su ulazno/izlazne usluge zatražene nad takvim objektom, završetak usluge signaliziran je porukom koja se stavlja na red *I/O completion port*-a. Proces koji potražuje ulazno/izlaznu uslugu nije obaviješten kada je ta usluga završena nego mora provjeravati red poruka *I/O completion port*-a kako bi odredio status svojih ulazno/izlaznih zahtjeva [19, 20].

Budući da je modul za pohranjivanje video sadržaja na trajnu memoriju računala potrebno realizirati u C programskom okruženju, korišten je Visual Studio 2017 Community [21]. Visual Studio je integrirano razvojno okruženje, IDE (engl. *Integrated Development Environment*), kojeg je razvio Microsoft. Koristi se za razvoj raznih programskih rješenja poput web stranica, web aplikacija, mobilnih aplikacija, računalnih programa i dr. Koristi Microsoftove razvojne platforme kao što su Windows API, Windows Forms, Windows Store, Windows Presentation Foundation i ostalo. Sastoji se od uređivača teksta, alata za pronalaženje pogrešaka u kodu, grafičkog dizajnera za lakšu vizualizaciju aplikacije i ostalih alata potrebnih za lakše pisanje, uređivanje i održavanje koda. Također, podržava velik broj dodataka koje su napravili korisnici. Sveukupno je u Visual Studio razvojnom okruženju podržano 36 različitih programskih jezika, a neki od značajnijih su C, C++, .NET, C# te JavaScript. Nadalje, moguće je dodati podršku za nepodržane programske jezike (Python, Ruby, Node.js, ...) pomoću korisničkih dodataka [22].

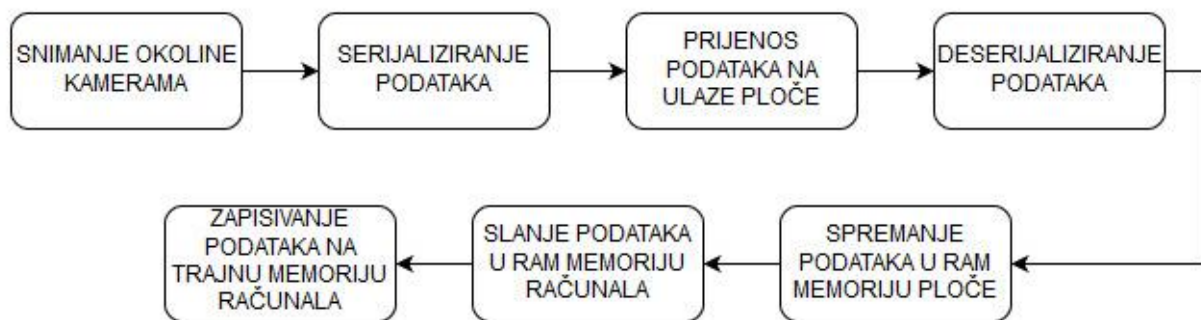
### **3.4. Koncept rješenja**

Kamere uz pomoć senzora za snimanje slike (engl. *image-capturing sensor*) snimaju okolinu. Dobiveni podaci su u *RAW* obliku što znači da nisu obrađeni nego sadrže sve što je senzor uspio snimiti, te prvo moraju biti konvertirani kako bi bili pogodni za korištenje u programima za



grafičko uređivanje slika [23]. Podaci se zatim serijaliziraju i šalju preko koaksijalnog kabela do hardverske ploče. Pri primitku podataka na ploči, podaci se prvo moraju deserijalizirati kako bi ih procesor mogao spremati u vlastitu RAM memoriju. AMV *Grabber* hardverska ploča sadrži 1 GB RAM memorije. Memorija je podijeljena tako da je prvih 21 MB rezervirano za potrebe FPGA (engl. *Field-Programmable Gate Array*) dizajna i pokretanja programa. Idućih 4 MB koristi se za spremanje adresnih deskriptora, a na sljedećih 60 MB spremaju se podatkovni deskriptori. Sveukupno je za potrebe ostalih modula zauzeto 85 MB prostora na RAM memoriji ploče.

Deskriptori su potrebni prilikom komunikacije između ploče i računala, tj. za slanje slika. Rade tako da slobodan prostor u RAM memoriji računala povezuju s prostorom u RAM memoriji ploče. Za proces povezivanja RAM memorije računala s RAM memorijom ploče koriste se adresni deskriptori. Oni spremaju adrese slobodnog prostora s RAM memorije računala u RAM memoriju ploče. Podatkovni deskriptori služe za prijenos podataka s RAM memorije ploče na RAM memoriju računala. Svaki podatkovni deskriptor prenosi 4 KB podataka. Kao što je prikazano na slici 3.3., prijenos se vrši tako da se s lokacija iz RAM memorije ploče šalju podaci na povezane lokacije u RAM memoriji računala. Nakon toga se sadržaj iz RAM memorije računala zapisuje u datoteku na trajnoj memoriji računala koju predstavlja HDD (engl. *Hard Disk Drive*) ili SSD (engl. *Solid State Drive*). Datoteka ima nastavak *.dat* i sadržava sve podatke koje je senzor kamere uspio snimiti.



**Sl. 3.3.** Blok dijagram prijenosa podataka s AMV *Grabber* ploče na računalo.

Ostatak RAM memorije ploče podijeljen je na tri međuspremnika. U svaki međuspremnik stane 248832000 B, tj. 243000 KB podataka, odnosno, u svaki međuspremnik svih devet kamera može zapisati po deset okvira najveće rezolucije  $1280 \times 1080$  elemenata slike, s 2 B po elementu slike. Sva tri međuspremnika zajedno zauzimaju 746496000 B, odnosno 729000 KB prostora na RAM memoriji ploče.

Podaci se u RAM memoriju ploče spremaju tako da procesor stvori zaglavlje okvira (engl. *frame header*) koje se zapisuje u RAM memoriju prije nego što se zapiše okvir. Zaglavlja se stvaraju za svaki okvir, a sastoje se od vremenske oznake i oznake ulaza na koji je kamera koja je poslala taj okvir spojena. Veličina zaglavlja okvira je 8 B. Zapisivanje zaglavlja okvira se izvršava na strani hardvera. Također, implementirano je i zapisivanje zaglavlja datoteke (engl. *file header*) koje se događa na strani računala. Zaglavlje datoteke veličine je 1024 B, odnosno 1 KB, i sadržava podatke o svim aktivnim ulazima s kojih se primaju podaci s kamera. Podaci sadržavaju informacije o rednom broju ulaza, širini okvira u elementima slike, visini okvira u elementima slike, broju bajtova po elementu slike i broju okvira po sekundi. Podaci zauzimaju 5 B za svaki ulaz pa bi tako, npr., četiri aktivna ulaza zauzimala 20 B prostora u zaglavlju datoteke. Prostor do 1024-og B u zaglavlju datoteke na koji ništa nije zapisano popunjen je nulama. Zapisivanje zaglavlja datoteke u datoteku odvija se prije nego je prijenos podataka iz RAM memorije ploče na računalo započet. Na slici 3.4. prikazana je struktura datoteke koja sadrži zaglavlje datoteke, zaglavlja okvira te okvire.



**Sl. 3.4.** *Struktura datoteke zapisane na trajnoj memoriji računala.*

Nakon zapisivanja zaglavlja datoteke započinje se sa zapisivanjem podataka koji pristižu u RAM memoriju računala. Zbog načina na koji je rješenje zamišljeno i zbog ograničene količine RAM memorije na računalu i ploči, podaci se šalju naizmjenično. Dok se u jedan međuspremnik na RAM memoriji ploče zapisuju podaci s kamera, iz drugog međuspremnika, u koji su otprije zapisani podaci, se ti podaci šalju na RAM memoriju računala. Zbog toga računalo odjednom može zapisati najviše deset okvira rezolucije  $1280 \times 1080$  elemenata slike, s 2 B po elementu slike što je 27648000 B, odnosno 27000 KB. Ta količina podataka predstavlja „paket“ koji je poslan s ploče u određenom trenutku.

Po završetku zapisivanja podataka u datoteku, datoteka može biti konvertirana iz *RAW* formata u format koji može biti reproduciran pomoću programa za reprodukciju video sadržaja. Budući da senzor u kameri koristi Bayerov filter, konvertiranje se vrši procesom zvanim *debayering*. Tim procesom se iz svakog elementa slike iščitava vrijednost za svaku RGB (engl. *Red-Green-Blue*) komponentu. Element slike sa zelenim filterom daje točnu vrijednost zelene komponente. Crvena

i plava komponenta za element slike sa zelenim filterom se dobivaju od susjednih elemenata slike. Vrijednost crvene boje dvaju susjeda se zbroji da se dobije crvena komponenta tog elementa slike, a da bi se dobila plava komponenta zbrajaju se vrijednosti plave boje dvaju susjeda [24].

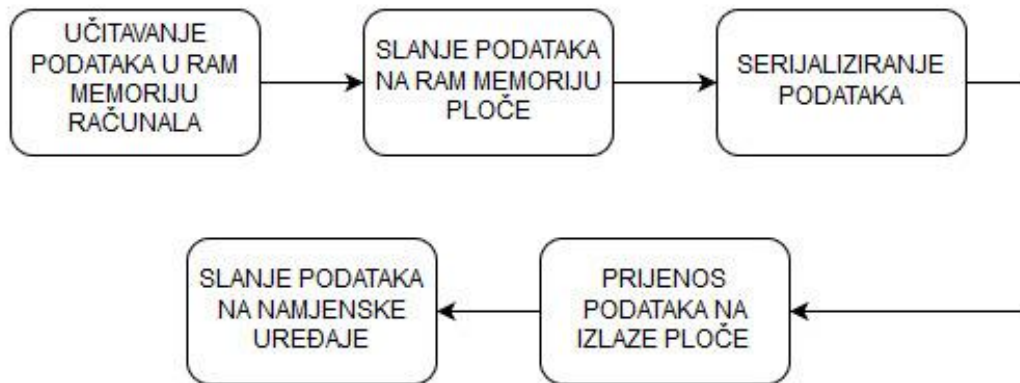
Slika 3.5. prikazuje izdvojeni okvir nakon što je nad prenesenim podacima provedeno konvertiranje kojim je dobiven gledljiv video sadržaj. Kao što se vidi sa slike, korištena je kamera s *fish-eye* lećom te je zbog toga slika „zaobljena“.



**Sl. 3.5.** Izdvojeni okvir nakon postupka konverzije.

Osim konvertiranja, datoteka, odnosno podaci iz datoteke u *RAW* formatu mogu se poslati natrag prema ploči. Kao što je vidljivo sa slike 3.6., slanje se vrši tako da se prvo iz datoteke samo okviri bez zaglavlja okvira i bez zaglavlja datoteke učitaju u RAM memoriju računala. Iz RAM memorije računala podaci se šalju preko *PCIe* sabirnice na RAM memoriju ploče uz pomoć

adresnih i podatkovnih deskriptora. Iz RAM memorije ploče podaci se zatim šalju na izlaze ploče gdje prvo moraju biti serijalizirani kako bi bili spremni za slanje. Podaci se mogu poslati na bilo koji uređaj koji ih može primiti, ali primarno su namijenjeni za slanje na uređaje na kojima se testiraju ADAS algoritmi. Za slanje podataka, kao i za primanje s kamera, koristi se FPD-Link III sučelje te koaksijalni kabel.



**Sl. 3.6.** Blok dijagram prijenosa podataka s računala na AMV Grabber ploču.

### 3.5. Implementacija rješenja

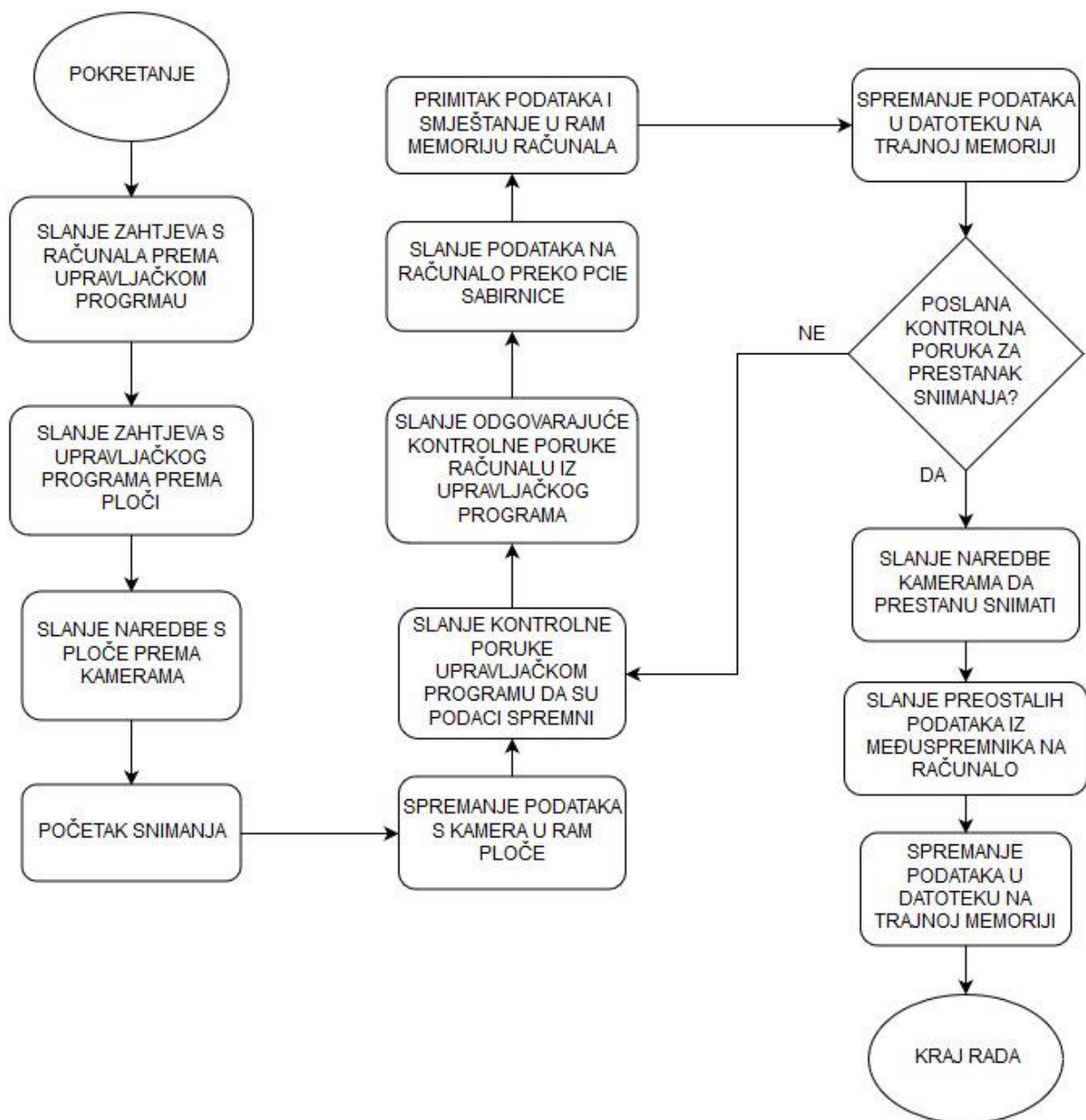
Kao što je i navedeno u prethodnom potpoglavlju, da bi se inicirao prijenos podataka iz RAM memorije ploče na RAM memoriju računala, potrebno je poslati zahtjev upravljačkom programu kojim se započinje proces snimanja kamerama. Upravljački program taj zahtjev obradi i šalje na ploču odgovarajući zahtjev kojeg AMV Grabber ploča može „razumjeti“. Ploča na temelju tog zahtjeva daje naredbu kamerama da započnu sa snimanjem. Senzor u kamerama snima sve podatke koji su mu dostupni te se ti podaci na strani kamera serijaliziraju. Podaci se šalju koaksijalnim kabelom do ploče gdje su deserijalizirani kako bi ih centralna procesorska jedinica, CPU (engl. *Central Processing Unit*), mogla smjestiti u RAM memoriju ploče. Budući da je struktura RAM memorije ploče obrađena u prošlom poglavlju, ovdje će se samo navesti da se primljeni podaci s kamera u RAM memoriju ploče spremaju od 85-og MB pa sve dok se međuspremnici ne popune.

Na slici 3.7. vidljiv je dijagram toka prijenosa podataka s AMV Grabber ploče na računalo. Kako bi bilo moguće započeti s procesom prijenosa podataka u RAM memoriju računala, potrebno je pričekati dok se jedan od međuspremnika u RAM memoriji ploče ne popuni primljenim podacima s kamera. Nakon što se jedan međuspremnik napuni, CPU počinje s popunjavanjem drugog međuspremnika i tako radi naizmjenično. Istovremeno s popunjavanjem sljedećeg

međuspremnik započinje se prijenos podataka iz već popunjenog međuspremnik na RAM memoriju računala. Upravljački program služi kao posrednik jer on inicira taj prijenos kad primi kontrolnu poruku od AMV *Grabber* ploče da je jedan od međuspremnik popunjen. Upravljački program zatim šalje kontrolnu poruku računalu, tj. aplikaciji na računalu kojom joj signalizira da se pripremi na primitak podataka s ploče. Prijenos podataka s ploče na računalo vrši se preko PCIe sabirnice.

Pri primitku podataka računalo ih smješta u vlastitu RAM memoriju za lako pristupanje pri pozivu WinAPI funkcije za spremanje na trajnu memoriju. WinAPI funkcija za spremanje je *WriteFile()* koja prima pokazivač na početnu adresu niza od 27000 KB podataka u RAM memoriji računala koji predstavljaju „paket“ od deset okvira. Također prima i *handle* na datoteku na trajnoj memoriji računala u koju će se zapisati „paket“. Funkcija se poziva svaki put kad aplikacija primi kontrolnu poruku od upravljačkog programa.

Proces slanja podataka iz RAM memorije ploče na RAM memoriju računala odvija se sve dok računalo ne pošalje upravljačkom programu kontrolnu poruku za prestanak snimanja. Upravljački ju program obradi i pošalje ploči odgovarajuću kontrolnu poruku koju ploča „razumije“. Ploča po primitku kontrolne poruke daje naredbu kamerama da zaustave snimanje. Kamere zaustavljaju snimanje tek kad snime broj okvira djeljiv s 10 jer se tako želi osigurati popunjenost međuspremnik u RAM memoriji ploče i pravilan prijenos podataka na RAM memoriju računala. Nakon snimanja dostatnog broja okvira, započinje se zadnji prijenos podataka iz međuspremnik na računalo. Računalo prihvaća podatke te ih sprema na trajnu memoriju. Nakon što se zadnji podaci trajno pohrane, program na računalu završava s izvođenjem.



Sl. 3.7. Dijagram toka prijenosa podataka.

## 4. EVALUACIJA VLASTITOG RJEŠENJA

U ovom poglavlju opisana su provedena mjerenja kojima se želi utvrditi ispravnost i učinkovitost izrade programske podrške za snimanje video sadržaja za verifikaciju ADAS algoritama. Postoje dva moguća „uska grla“ koja bi ograničavala rad vlastitog rješenja, a to su brzina prijenosa podataka preko PCIe sabirnice u implementaciji vlastitog rješenja te brzina zapisivanja pristiglih podataka na trajnu memoriju računala.

### 4.1. Brzina prijenosa preko PCIe sabirnice

Teorijska brzina prijenosa podataka preko PCIe sabirnice u jednom smjeru je 2 GB/s jer je konfiguracija korištena u implementaciji rješenja PCIe Gen2.0 x4. Međutim, zbog načina na koji su podaci koji se prenose koncipirani, odnosno načina na koji su podijeljeni u pakete, stvarna brzina je ipak niža od teorijske što je vidljivo iz mjerenja.

Mjerenja su izvršena za devet „paketa“ od deset okvira što je iznos od 243000 KB, odnosno jedan cijeli međuspremnik. Svako od pet mjerenja izvršeno je tako da je stavljen vremenski brojač koji mjeri vrijeme između primanja prvog bajta i zadnjeg bajta podataka na računalu. Zatim je broj prenesenih podataka podijeljen s izmjerenim vremenom kako bi se dobila brzina prijenosa podataka preko PCIe sabirnice.

Brzina prijenosa podataka računa se prema sljedećem izrazu:

$$v = \frac{Q}{t} \quad (4-1)$$

gdje je:

- $v$  – brzina prijenosa podataka u GB/s
- $Q$  – količina podataka u GB
- $t$  – vrijeme u s.

Prema izrazu (4-1) za svako su mjerenje izračunate brzine prijenosa podataka te su prikazane u tablici 4.1.



**Tab. 4.1.** Mjerenja brzine prijenosa preko PCIe sabirnice.

Mjerenje	Količina podataka (u KB)	Izmjereno vrijeme (u ms)	Brzina prijenosa (u GB/s)
1.	243000	236	0,982
2.	243000	242	0,958
3.	243000	231	1,003
4.	243000	239	0,969
5.	243000	229	1,012
6.	243000	231	1,003
7.	243000	234	0,990
8.	243000	237	0,978
9.	243000	240	0,966
10.	243000	235	0,986

Na temelju podataka za brzinu prijenosa iz tablice 4.1. izračunata je srednja vrijednost brzine prijenosa podataka prema izrazu:

$$\bar{v} = \frac{v_1 + \dots + v_n}{n} \quad (4-2)$$

gdje je:

- $\bar{v}$  – srednja vrijednost brzine prijenosa,
- $v_1, \dots, v_n$  – vrijednosti brzine prijenosa za svako mjerenje,
- $n$  – broj izvršenih mjerenja.

Osim srednje vrijednosti brzine prijenosa podataka izračunata je standardna devijacija prema izrazu:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (v_i - \bar{v})^2} \quad (4-3)$$

gdje je:

- $\sigma$  – standardna devijacija brzine prijenosa,
- $\bar{v}$  – srednja vrijednost brzine prijenosa,

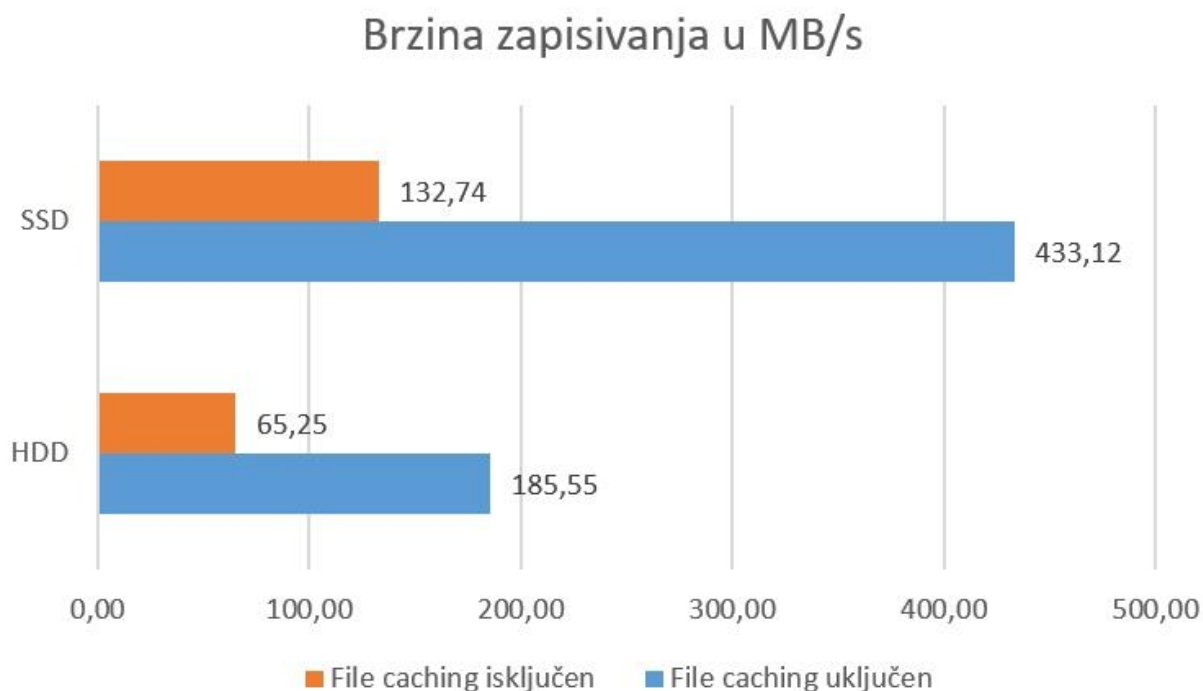


- $v_{1,\dots,n}$  – vrijednost brzine prijenosa za svako mjerenje,
- $n$  – broj izvršenih mjerenja.

Prema izrazu (4-2) srednja vrijednost brzine prijenosa podataka s RAM memorije ploče na RAM memoriju računala iznosi 0,9847 GB/s uz standardnu devijaciju od 0,017658 GB/s izračunatu prema izrazu (4-3). Zbog načina na koji je vlastito rješenje implementirano, ta je brzina prijenosa podataka dovoljna jer se želi omogućiti prebacivanje najviše 243000 KB podataka odjednom. Također, iz tablice 4.1. vidljivo je da je minimalna brzina prijenosa 0.958 GB/s, a maksimalna 1,012 GB/s iz čega je zaključeno da se brzina prijenosa nikad dovoljno ne smanji da bi utjecala na stvaranje grešaka u prijenosu podataka.

## 4.2. Brzina zapisivanja na disk

Kako bi se izmjerila općenita brzina zapisivanja podataka na disk, bilo to HDD ili SSD, korišten je softver CrystalDiskMark [25] stvoren posebno za tu namjenu. Podaci dobiveni softverskim mjerenjem su približni vrijednostima koje su proizvođači naveli na ambalažama proizvoda, odnosno stvarna je brzina najviše 20% manja od navedene. Za HDD [26] su navedene vrijednosti zapisivanja na disk od 225 MB/s dok je izmjerena brzina cca. 185 MB/s, a za SSD [27] je navedena brzina zapisivanja od 520 MB/s dok je izmjerena cca. 433 MB/s. Mjerenje je obavljeno za zapisivanje 1 GB podataka na disk. Također, testirano je zapisivanje na disk kada je opcija *File caching* uključena i kada je isključena. Mjerenje je provedeno samo jednom jer se želi pokazati koliki utjecaj ima *File caching* opcija na brzinu zapisivanja podataka. Kao što se vidi na slici 4.1., brzina zapisivanja na disk kada je opcija *File caching* isključena i za HDD i za SSD smanji se otprilike tri puta.



**Sl. 4.1.** Graf s prikazom brzine zapisivanja za HDD i SSD s uključenom i isključenom File caching postavkom.

Mjerenja brzine zapisivanja na disk provedena su za konfiguracije s tri i sa šest kamera. Mjerenje za konfiguraciju sa svih devet kamera nije bilo moguće provesti jer ni na jednom od tri prototipa AMV *Grabber* ploče nisu svi ulazi za kamere bili ispravni. Provedeno je deset mjerenja za svaku konfiguraciju. Mjerenja su provedena tako da je štopericom mjereno vrijeme od početka snimanja s kamera sve dok aplikacija ne prestane s radom. Nažalost, veličina predmemorije nije mogla biti očitana što znači da mjerenja nisu provedena u istim uvjetima što i dokazuje vrijeme snimanja za svako mjerenje. Unatoč tome, za svako mjerenje zapisana je različita količina podataka na disk pa se skaliranjem dobiju podjednake brzine zapisivanja na disk za svako mjerenje što je vidljivo iz tablica 4.2. i 4.3. Mjerenja su izvršena samo za rezoluciju od  $1280 \times 1080$  elemenata slike.

Kako bi se dobila brzina zapisivanja korišten je sljedeći izraz:

$$v = \frac{Q}{t} \tag{4-4}$$

gdje je:

-  $v$  – brzina zapisivanja na disk u MB/s,

-  $Q$  – količina podataka u MB,

-  $t$  – vrijeme u s.

Prema izrazu (4-4) za svako su mjerenje izračunate brzine zapisivanja podataka na HDD za konfiguraciju s tri kamere te su prikazane u tablici 4.2..

**Tab. 4.2.** Rezultati mjerenja brzine zapisivanja na disk (HDD) za konfiguraciju s tri kamere.

Mjerenje	Količina podataka (u MB)	Izmjereno vrijeme (u s)	Brzina zapisivanja (u MB/s)
1.	36070,31	192	188,85
2.	33222,65	176	188,76
3.	58139,65	302	192,52
4.	41765,63	219	190,71
5.	31561,52	168	187,87
6.	32748,05	175	187,13
7.	35833,01	186	192,65
8.	34646,48	185	187,28
9.	36544,92	194	188,38
10.	36307,62	192	189,10

Na temelju podataka za brzinu zapisivanja iz tablice 4.2. izračunata je srednja vrijednost brzine zapisivanja podataka na disk prema izrazu:

$$\bar{v} = \frac{v_1 + \dots + v_n}{n} \quad (4-5)$$

gdje je:

-  $\bar{v}$  – srednja vrijednost brzine zapisivanja na disk,

-  $v_1, \dots, v_n$  – vrijednosti brzine zapisivanja na disk za svako mjerenje,

-  $n$  – broj izvršenih mjerenja.

Osim srednje vrijednosti brzine zapisivanja podataka na disk izračunata je standardna devijacija prema izrazu:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (v_i - \bar{v})^2} \quad (4-6)$$

gdje je:

- $\sigma$  – standardna devijacija brzine zapisivanja na disk,
- $\bar{v}$  – srednja vrijednost brzine zapisivanja na disk,
- $v_{1,\dots,n}$  – vrijednost brzine zapisivanja na disk za svako mjerenje,
- $n$  – broj izvršenih mjerenja.

Prema izrazu (4-5) srednja vrijednost brzine zapisivanja podataka na disk iznosi 189,325 MB/s, a standardna devijacija izračunata prema izrazu (4-6) iznosi 1,994087 MB/s. Minimalna izmjerena brzina zapisivanja iznosi 187,13 GB/s, a maksimalna iznosi 192,65 MB/s što znači da tijekom mjerenja nije dolazilo do većih promjena u brzini zapisivanja podataka na disk. U jednom od mjerenja brzina zapisivanja podataka na disk smanji se već nakon 31,5 GB, a u drugom nakon 58,1 GB. Iz toga se može zaključiti da je u prvom slučaju veličina predmemorije bila manja nego u drugom zbog čega je i došlo do toga da se brzina zapisivanja smanji već nakon 31,5 GB.

Prema izrazu (4-4) za svako su mjerenje izračunate brzine zapisivanja podataka na SSD za konfiguraciju sa šest kamera te su prikazane u tablici 4.3..

**Tab. 4.3.** Rezultati mjerenja brzine zapisivanja na disk (SSD) za konfiguraciju sa šest kamera.

Mjerenje	Količina podataka (u MB)	Izmjereno vrijeme (u s)	Brzina zapisivanja (u MB/s)
1.	77361,33	182	425,06
2.	75937,50	174	436,42
3.	78785,16	184	428,18
4.	76412,11	179	426,88
5.	77361,33	180	429,78
6.	72140,63	170	424,36
7.	75462,89	176	428,77
8.	75937,50	178	426,62
9.	77835,94	183	425,33
10.	76886,72	179	429,53

Na temelju podataka za brzinu zapisivanja iz tablice 4.3. izračunata je srednja vrijednost brzine zapisivanja podataka na disk prema izrazu (4-5) te ona iznosi 428,093 MB/s. Standardna devijacija izračunata prema izrazu (4-6) iznosi 3,482104 MB/s. Također, minimalna izmjerena brzina zapisivanja iznosi 424,36 MB/s, a maksimalna iznosi 436,42 MB/s. Kao i u slučaju kod zapisivanja na HDD, i ovdje je zaključeno da nije došlo do većih promjena u brzini zapisivanja što znači da je količina zapisanih podataka na SSD bila slična za svako mjerenje. Pri mjerenjima brzine zapisivanja na disk za konfiguraciju sa šest kamera nije uočeno da se veličina predmemorije razlikovala u velikoj mjeri od mjerenja do mjerenja jer su izmjerena vremena ostajala otprilike ista.

Iz provedenih mjerenja zaključeno je da nije zadano kada će se brzina prijenosa smanjiti jer ovisi o slobodnom prostoru u predmemoriji i o broju kamera koje se koriste za snimanje. Što je broj kamera manji, brzina zapisivanja ostaje konstantna duži vremenski period. Isto vrijedi i kada je više prostora u predmemoriji slobodno, što je dokazano time da se tijekom testiranja dogodilo da se brzina zapisivanja u konfiguraciji s tri kamere u jednom slučaju smanji već nakon zapisivanja 31,5 GB podataka, a u drugom tek nakon 58,1 GB podataka. Pretpostavlja se da bi se vremenski period prije nego se brzina zapisivanja smanji mogao produljiti ako bi se povećao kapacitet alocirane predmemorije što nije bilo moguće testirati zbog nedovoljnog kapaciteta RAM memorije računala.

### **4.3. Problemi prilikom rada aplikacije**

Tijekom testiranja uspješnosti implementacije vlastitog rješenja došlo je do problema zbog većinom softverskih ograničenja. Ta ograničenja odnose se na način funkcioniranja Windows operacijskog sustava i na prijenos podataka preko PCIe sabirnice putem deskriptora.

Prvo ograničenje događa se zato što Windows operacijski sustav koristi paradigmu zvanu *File caching* [28] za ubrzavanje pisanja na disk. Prilikom zapisivanja podataka na disk, Windows alocira dio slobodnog prostora RAM memorije računala kao predmemoriju. U predmemoriju se spremaju naredbe za zapisivanje podataka na disk. Te naredbe se čuvaju u predmemoriji sve dok disk ne postane dostupan za zapisivanje podataka. Na taj se način ubrzava rad računala jer aplikacije ne moraju čekati da se podaci zapišu na disk, tj. aplikacije nisu blokirane.

Kod prijenosa podataka s ploče na računalo, poslani podaci se prvo spremaju u RAM memoriju računala. Zatim se zapisuju na trajnu memoriju računala, tj. na disk (HDD/SSD). Međutim, budući

da Windows po zadanim postavkama koristi predmemoriju za ubrzanje pisanja na disk i da je ona ograničene veličine ovisno o količini slobodne RAM memorije, predmemorija se relativno brzo popuni podacima koji čekaju da budu zapisani na disk. „Usko grlo“ postaje disk koji ne može zapisivati podatke iz predmemorije dovoljno brzo, odnosno brzina kojom podaci stižu u predmemoriju je veća od brzine zapisivanja na disk. Brzina popunjavanja predmemorije se razlikuje ovisno o broju kamera s kojih dolaze podaci. Što je manji broj kamera, predmemorija se sporije popunjava, ali nakon određenog vremena se napuni, bez obzira snima li se samo jednom kamerom ili sa svih devet kamera.

U početnim je testiranjima kao trajna memorija računala korišten HDD. Njegova teorijska, ali i stvarna brzina zapisivanja omogućavala je snimanje s najviše tri kamere uz spremanje na disk u stvarnom vremenu. Budući da je predmemorija ograničena, nakon otprilike 3 minute došlo bi do smanjenja stvarne brzine zapisivanja jer HDD ne može zapisati onoliko podataka iz predmemorije koliko ih stigne u nju. Smanjenje brzine zapisivanja naposljetku bi dovelo do greške u aplikaciji što znači da je idući korak u rješavanju problema omogućavanje veće brzine zapisivanja.

Rješenje koje omogućuje veću brzinu zapisivanja, a relativno je jeftino, ovisno o kapacitetu, je SSD. Korišteni SSD imao je stvarnu brzinu zapisivanja od otprilike 470 MB/s što je dovoljno za udvostručenje broja aktivnih kamera. Broj kamera je tako porastao s tri na šest, ali bi se brzina zapisivanja opet smanjila nakon kraćeg vremenskog perioda (cca. 2 minute) zbog popunjenosti predmemorije. Smanjena brzina zapisivanja opet bi dovela do greške u radu aplikacije i prekidanju snimanja s kamera.

Problem nije u potpunosti riješen jer se brzina zapisivanja uvijek smanji, neovisno o korištenom disku, što dovede do pogreške u radu aplikacije. Kako bi se uklonilo javljanje greške i rušenje aplikacije, korišten je mehanizam kojim se „zaobilazi“ ovaj problem. Umjesto dopuštanja aplikaciji da se sruši, aplikacija se sama zatvori i deinicijalizira sve inicijalizirane komponente sustava kad očita da podaci koji stižu s ploče ne mogu biti zapisani na disk dovoljno brzo. Tim je procesom izbjegnuto neočekivano zaustavljanje sustava zbog čega komponente sustava ostanu inicijalizirane. Zbog inicijaliziranih komponenti sustava aplikacija ne može biti pokrenuta sve dok se cijeloj AMV *Grabber* hardverskoj ploči ne prekine dotok električne energije. Prekidom i ponovnim uspostavljanjem dotoka energije ploči se omogućuje ponovno inicijaliziranje svih komponenti sustava.

Gledano sa strane diska, pretpostavka je da bi NVMe (engl. *Non-Volatile Memory Express*) SSD, koji za rad s podacima koristi PCIe standard i ima brzine zapisivanja do 2 GB/s ovisno o

modelu i kapacitetu, omogućio neprekidno snimanje sa svih devet kamera što je i cilj ovog diplomskog rada. Međutim, zbog financijskih i vremenskih troškova, rješenje s NVMe SSD-om nije testirano pa nije moguće sa potpunom sigurnošću ustanoviti da je pretpostavka valjana.

Također, testirano je zapisivanje na disk (HDD/SSD) i kad je *File caching* opcija u Windows operacijskom sustavu potpuno isključena. Zaključeno je da je bolje ostaviti *File caching* opciju uključenom jer, ako je isključena, čim je prijenos podataka započet, disk ne stiže zapisati podatke koji pristižu u predmemoriju. Brzina zapisivanja onda je konstantna, ali je uvijek ona smanjena kao kad je predmemorija popunjena.

Drugo ograničenje odnosi se na stvaranje zaglavlja okvira u RAM memoriji ploče i prijenos zaglavlja okvira i okvira na RAM memoriju računala. Budući da je količina podataka koju kamere uvijek snime, neovisno o rezoluciji, djeljiva s 4096 B, odnosno 4 KB, u početku je odlučeno da podatkovni deskriptori trebaju prenositi 4 KB podataka. Zbog takve implementacije, za okvir rezolucije  $1280 \times 1080$  potrebno je 675 deskriptora, za okvir rezolucije  $1280 \times 800$  potrebno je 500 deskriptora, a za okvir rezolucije  $1280 \times 720$  potrebno je 450 deskriptora veličine 4 KB. To znači da bi podaci bili savršeno poravnati jer ne bi bilo viška ili manjka podataka pri prijenosu.

Međutim, u kasnijoj fazi projekta odlučeno je kako bi svaki okvir trebao imati svoje zaglavlje radi lakšeg razlikovanja i sortiranja okvira na računalo. Zaglavlje treba sadržavati vremensku oznaku i oznaku ulaza na koji je kamera koja je snimila taj okvir priključena, te treba biti veličine 8 B. Zbog točno određenog broja deskriptora koji su potrebni za prijenos pojedinog okvira, nije bilo moguće prenijeti sva zaglavlja okvira i sve okvire na računalo. Uvijek bi od zadnjeg okvira u „paketu“ od deset okvira nedostajala količina podataka u iznosu od sveukupnog broja okvira, u ovom slučaju deset okvira, pomnožena s 8 B. To znači da bi zadnjem okviru nedostajalo 80 B podataka i on bi bio nepotpun. Iako 80 B podataka nije vidljivo na konvertiranom video sadržaju, zbog testiranja ADAS algoritama svi podaci moraju biti prisutni kako bi se očuvao integritet rezultata testiranja.

Problem je riješen tako što je dodan jedan deskriptor veličine 4 KB na svaki „paket“ od deset okvira. Time je dodan prostor potreban za prijenos svih snimljenih okvira i zaglavlja okvira, ali se prenose i *garbage* vrijednosti koje se nalaze u slobodnom dijelu RAM memorije ploče. Na strani računala te se *garbage* vrijednosti odbacuju i u datoteku se zapisuje samo „korisni“ dio prenesenih podataka što je 27000 KB koji predstavljaju 10 okvira i 80 B koji predstavljaju 10 zaglavlja okvira, odnosno 27648080 B.

## 5. ZAKLJUČAK

Automobilska je industrija ulaganjem u autonomna vozila omogućila i razvoj prateće tehnologije koja treba olakšati testiranje i verifikaciju ADAS algoritama potrebnih za razvoj autonomne vožnje. Zbog toga je u sklopu ovog rada izrađena programska podrška za snimanje video sadržaja za verifikaciju ADAS algoritama na strani računala. Programska podrška mora omogućiti slanje video sadržaja s *AMV Grabber* ploče na računalo preko PCIe sabirnice. Na računalu se video sadržaj treba pohraniti na trajnu memoriju kako bi kasnije mogao biti iskorišten za testiranje i verifikaciju ADAS algoritama na namjenskim uređajima.

U radu je najprije predstavljen koncept vlastitog rješenja za navedeni problem. Opisana je implementacija koncepta rješenja te problemi koji su se javili prilikom testiranja uspješnosti implementacije. Testiranjem je utvrđeno da je koncept realiziran, ali ne u potpunosti jer se pojavilo softversko ograničenje. Ono se odnosi na način kako Windows operacijski sustav zapisuje podatke na disk. Može se u potpunosti riješiti tako da se koristi disk s dostatnom brzinom zapisivanja podataka.

Nedostaci implementiranog rješenja su smanjivanje brzine zapisivanja na disk nakon određenog vremenskog perioda što nije problem u konceptu rješenja već specifičnost Windows operacijskog sustava. Također, zbog upravljačkog programa koji je funkcionalan samo na Windows operacijskom sustavu, prijenos i spremanje podataka s ploče na računalo nije moguće izvršiti u drugim operacijskim sustavima.

U mogućim budućim iteracijama vlastitog rješenja može se dodati podrška za izvršavanje na drugim operativnim sustavima. Također, dodavanjem trajne memorije koja podržava veće brzine zapisivanja može se omogućiti zapisivanje podataka sa svih devet kamera.



## LITERATURA

- [1] „How Automakers Can Survive the Self-Driving Era“. [Na internetu]. Dostupno na: <http://www.atkearney.com/automotive/article?/a/how-automakers-can-survive-the-self-driving-era>. [Pristupljeno: 12-ruj-2018].
- [2] „Automated Driving Systems: A Vision for Safety“. [Na internetu]. Dostupno na: [https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/13069a-ads2.0\\_090617\\_v9a\\_tag.pdf](https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/13069a-ads2.0_090617_v9a_tag.pdf). [Pristupljeno: 05-ruj-2018].
- [3] „Autonomous car“, *Wikipedia*. 01-ruj-2018.
- [4] „Advanced driver-assistance systems“, *Wikipedia*. 04-kol-2018.
- [5] „Devices and Hardware for Video Capture, Streaming & Recording“. [Na internetu]. Dostupno na: <https://www.epiphan.com/products/>. [Pristupljeno: 30-kol-2018].
- [6] „Industrial Frame Grabbers“. [Na internetu]. Dostupno na: <https://www.phase1vision.com/frame-grabbers>. [Pristupljeno: 30-kol-2018].
- [7] „CoaXPress“. [Na internetu]. Dostupno na: <http://www.coaxpress.com/>. [Pristupljeno: 06-ruj-2018].
- [8] „Camera Link Camera Accessories“. [Na internetu]. Dostupno na: <https://www.1stvision.com/cameras/accessories/camera-link-camera-accessories>. [Pristupljeno: 06-ruj-2018].
- [9] „VGA connector“, *Wikipedia*. 01-ruj-2018.
- [10] „HDMI“, *Wikipedia*. 28-srp-2018.
- [11] „HDMI video capture - AV.io HD Technical Specifications“, *Epiphan Video*. .
- [12] „DVI2PCIe Duo Technical Specifications“, *Epiphan Video*. .
- [13] „Active Silicon AS-FBD-4XCXP6-2PE8“, *Phase 1 Technology Corp*. [Na internetu]. Dostupno na: <https://www.phase1vision.com/frame-grabbers/active-silicon-frame-grabbers/coaxpress-frame-grabbers/as-fbd-4xcxp6-2pe8>. [Pristupljeno: 13-ruj-2018].
- [14] „Active Silicon ASFBD2XCLM-2PE8“, *Phase 1 Technology Corp*. [Na internetu]. Dostupno na: <https://www.phase1vision.com/frame-grabbers/active-silicon-frame-grabbers/camera-link-frame-grabbers/as-fbd-2xclm-2pe8>. [Pristupljeno: 13-ruj-2018].

- [15] „Dalsa Xtium-CLHS PX4“, *Phase 1 Technology Corp.* [Na internetu]. Dostupno na: <https://www.phase1vision.com/frame-grabbers/dalsa-frame-grabbers/camera-link-hs-frame-grabbers/xtium-clhs-px4>. [Pristupljeno: 13-ruj-2018].
- [16] „OmniVision | A leading developer of advanced digital imaging solutions.“ [Na internetu]. Dostupno na: <https://www.ovt.com/sensors/OV10640>. [Pristupljeno: 06-ruj-2018].
- [17] R. Budruk, Ur., *PCI express system architecture*. Boston: Addison-Wesley, 2004.
- [18] „PCI Express“, *Wikipedia*. 15-kol-2018.
- [19] „Input/output completion port“, *Wikipedia*. 11-tra-2017.
- [20] „Windows via C/C++: Synchronous and Asynchronous Device I/O | Microsoft Press Store“. [Na internetu]. Dostupno na: <https://www.microsoftpressstore.com/articles/article.aspx?p=2224047&seqNum=5>. [Pristupljeno: 12-ruj-2018].
- [21] „Visual Studio IDE“. [Na internetu]. Dostupno na: <https://visualstudio.microsoft.com/vs/>. [Pristupljeno: 31-kol-2018].
- [22] „Microsoft Visual Studio“, *Wikipedia*. 24-kol-2018.
- [23] „Raw image format“, *Wikipedia*. 30-kol-2018.
- [24] „Bayer filter“, *Wikipedia*. 25-svi-2018.
- [25] „CrystalDiskMark“. [Na internetu]. Dostupno na: <https://crystalmark.info/en/software/crystaldiskmark/>. [Pristupljeno: 04-ruj-2018].
- [26] „DT01ACA\*\*\* Series | TOSHIBA Semiconductor & Storage Products | Americas“. [Na internetu]. Dostupno na: <https://toshiba.semicon-storage.com/us/product/storage-products/client-hdd/dt01acaxxx.html>. [Pristupljeno: 10-ruj-2018].
- [27] „Ultimate SU800 SSD | ADATA Consumer“. [Na internetu]. Dostupno na: <http://www.adata.com/en/specification/410>. [Pristupljeno: 06-ruj-2018].
- [28] „File Caching | Microsoft Docs“. [Na internetu]. Dostupno na: <https://docs.microsoft.com/en-us/windows/desktop/fileio/file-caching>. [Pristupljeno: 04-ruj-2018].

## SAŽETAK

Tema ovog diplomskog rada izrada je vlastitog rješenja na problem prijenosa i pohrane video sadržaja na računalo za kasniju primjenu u verifikaciji i testiranju ADAS algoritama. Za snimanje video sadržaja može se koristiti najviše devet kamera. Rješenje je realizirano u obliku programske podrške koja video sadržaj s AMV *Grabber* hardverske ploče prenosi preko PCIe sabirnice na računalo gdje se sprema na trajnu memoriju. U radu su prikazana već postojeća rješenja na navedeni problem te su objašnjene tehnologije i standardi korišteni za realizaciju vlastitog rješenja. Također, predstavljen je način na koji se želi implementirati vlastito rješenje te je opisana stvarna implementacija rješenja. Vlastito rješenje ispravno radi, odnosno podaci se uspješno prenose s AMV *Grabber* hardverske ploče na računalo gdje se pohranjuju na trajnu memoriju. Iako je vlastito rješenje funkcionalno, prilikom testiranja javili su se problemi koji onemogućuju održavanje konstantne brzine zapisivanja podataka na disk u stvarnom vremenu, odnosno, nakon određenog vremenskog perioda brzina zapisivanja se smanji za otprilike tri puta. Stoga, omogućeno je zapisivanje podataka s najviše tri kamere ako se zapisuje na HDD, te s najviše šest kamera ako se zapisuje na SSD prije nego se brzina zapisivanja podataka smanji. Problem je djelomično riješen te je ponuđena pretpostavka za koju se smatra da bi mogla pomoći u potpunom rješavanju problema.

**Ključne riječi:** AMV *Grabber*, video sadržaj, prijenos podataka, PCIe sabirnica, zapisivanje na disk

## **ABSTRACT**

### **Software support for recording video content for the purposes of verifying ADAS algorithms on the computer side**

This master's thesis engages in the making of an own solution to the problem of transferring and storing video content on the computer for later use in testing and verifying ADAS algorithms. To record video content, up to nine cameras can be used. The solution is made in the form of software support which transfers video content from AMV *Grabber* hardware board over PCIe bus to the computer where it is stored on non-volatile memory. Already existing solutions to the problem are shown and technologies and standards used are explained in this thesis. The way the solution is to be implemented is also presented and the process of actual implementation is described. The solution works correctly, i.e. the data is successfully transferred from AMV *Grabber* board to the computer where it is stored on non-volatile memory. Even though the solution is functional, during testing some issues occurred which prevent the disk from maintaining constant write speeds, i.e., after a certain amount of time the write speeds decrease by about three times. Therefore, writing data from a maximum of three cameras is supported if writing occurs on an HDD, and if writing occurs on an SSD, writing from a maximum of six cameras is supported before the writing speed is decreased. The issue is partially solved and an assumption is offered which is considered to be helpful in solving the issue completely.

**Keywords:** AMV *Grabber*, video content, data transfer, PCIe bus, writing to disk

## ŽIVOTOPIS

Ivan Benke rođen je 6. travnja 1994. godine u Osijeku, Hrvatska. Živi na adresi Franje Račkog 15, Đakovo. Pohađa Osnovnu školu „Ivan Goran Kovačić“ u Đakovu do 2009. godine, te iste godine upisuje Gimnaziju „Antun Gustav Matoš“, opći smjer u Đakovu. Prva dva razreda srednje škole završava s vrlo dobrim, a druga dva s odličnim uspjehom. Po završetku srednje škole, 2013. godine, upisuje tadašnji Elektrotehnički fakultet Osijek, danas Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, smjer Računarstvo.

Izvršno se služi engleskim jezikom što potvrđuje sudjelovanje na međunarodnom natjecanju studenata tehničkih fakulteta „Elektrijada“ u disciplini „Engleski jezik“ 2016. te 2017. godine. Posjeduje srednju razinu znanja za rad s Microsoft Office alatima. Također, razina znanja za rad s programskim jezikom C je srednja, a razina znanja za rad sa C++ i C# programskim jezicima je osnovna.

Dobitnik je stipendije za deficitarna zanimanja Nacionalne zaklade za potporu učeničkom i studentskom standardu za 2015./2016., 2016./2017. te 2017./2018. akademsku godinu. Također, akademske godine 2016./2017. dobitnik je stipendije tvrtke RT-RK iz Osijeka, a trenutno je stipendist iste te tvrtke koja ga obučava za daljnji rad u tvrtki nakon diplomiranja.

Potpis:

---