

Razvoj programske podrške za 4K grabber uređaj

Petrinšak, Luka

Master's thesis / Diplomski rad

2016

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:200:750512>

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

Download date / Datum preuzimanja: **2024-08-12**

Repository / Repozitorij:

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



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

Diplomski sveučilišni studij

**RAZVOJ PROGRAMSKE PODRŠKE ZA 4K GRABBER
UREĐAJ**

Diplomski rad

Luka Petrinšak

Osijek, 2016.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada**

Osijek, 26.09.2016.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu diplomskog rada

Ime i prezime studenta:	Luka Petrinšak
Studij, smjer:	Diplomski sveučilišni studij Računarstvo, smjer Procesno računarstvo
Mat. br. studenta, godina upisa:	D 725 R, 14.10.2014.
OIB studenta:	91789633769
Mentor:	Doc.dr.sc. Ratko Grbić
Sumentor:	Zvonimir Kaprocki
Predsjednik Povjerenstva:	Doc.dr.sc. Mario Vranješ
Član Povjerenstva:	Doc.dr.sc. Marijan Herceg
Naslov diplomskog rada:	Razvoj programske podrške za 4K grabber uređaj
Znanstvena grana rada:	Programsko inženjerstvo (zn. polje računarstvo)
Zadatak diplomskog rada:	Jedna od metodologija koja se koristi u testiranju DTV uređaja je Black Box Testing (BBT). Kod ove metodologije testiranja, testirani uređaj se promatra kao crna kutija potpuno nepoznate interne strukture, koja za poznate ulaze treba dati očekivani odziv, koji je jedini kriterij za odlučivanje o rezultatu testa. U slučaju STB (engl. Set Top Box) uređaja, odziv uređaja su slika i zvuk, koji se najčešće dostavljaju do ekrana preko HDMI sučelja.
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 Postignuti rezultati u odnosu na složenost zadatka: 3 Jasnoća pismenog izražavanja: 3 Razina samostalnosti: 3
Datum prijedloga ocjene mentora:	26.09.2016.
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, 27.09.2016.

**Ime i
prezime
studenta:**

Luka Petrinšak

Studij:

Diplomski sveučilišni studij Računarstvo, smjer Procesno računarstvo

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

D 725 R, 14.10.2014.

**Ephorus
podudaranje
[%]:**

1

Ovom izjavom izjavljujem da je rad pod nazivom: **Razvoj programske podrške za 4K grabber uređaj**

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

i sumentora Zvonimir Kaprocki

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. UHD STANDARD.....	3
2.1 Rezolucija digitalnog okvira	3
2.2 Broj slikovnih okvira u sekundi vremena.....	5
2.3 Prostor boja	5
2.3.1 Y'UV prostor boja	6
2.4 Poduzorkovanje boja.....	7
2.5 Dubina boja	7
2.6 Omjer visine i širine okvira.....	8
3. KOMPRESIJA SLIKE	10
3.1 Kompresija slike s gubitcima	10
3.2 Kompresija slike bez gubitaka	11
3.3 Lempel-Ziv-Welch (LZW) algoritam	11
3.3.1 Način rada.....	12
4. IMPLEMENTACIJA LZW KOMPRESIJE ZA POTREBE PRIJENOSA	16
4.1 FFmpeg	16
4.2 Implementacija izrađenog programskog rješenja.....	16
4.2.1 Simulacija klijentske aplikacije i poslužitelja	17
4.2.2 Mjerenje vremena izvršavanja.....	18
5. ANALIZA REZULTATA LZW KOMPRESIJE ZA POTREBE PRIJENOSA	19
5.1 Tehničke specifikacije sklopovlja	19
5.2 Testne video sekvence.....	19
5.3 Rezultati testiranja programskog rješenja	21
5.3.1 Rezultati LZW kompresije nad prethodno komprimiranim FHD video sekvencama	22
5.3.2 Rezultati LZW kompresije na prethodno komprimiranim 4K UHD video sekvencama ...	25
5.3.3 Rezultati LZW kompresije nad neobrađenim FHD sekvencama	28
5.3.4 Rezultati LZW kompresije na neobrađenim 4K UHD sekvencama.....	31
5.4 Analiza rezultata testiranja programskog rješenja.....	35
6. ZAKLJUČAK	37
LITERATURA.....	38
SAŽETAK.....	39
ŽIVOTOPIS	41

1. UVOD

Početak 21. stoljeća započeo je razvoj video standarda koji je danas poznat pod nazivom televizija ekstremno visoke rezolucije (engl. *Ultra-High Definition Television, UHD TV*). Razlog razvoja UHD TV-a temelji se na istom cilju kao i razvoj prethodnih televizijskih standarda poput televizije visoke rezolucije (engl. *Full High-Definition Television, FHD TV*), a to je povećati razlučivost i kvalitetu slike na prikaznom području istih prostornih dimenzija. S obzirom da je rezolucija veća, slika je oštija, manje mutna te je u cjelini bliža stvarnosti. Međutim, ovo povlači za sobom problem slanja velike količine podataka u realnom vremenu. Rješenje eksponencijalnog odnosa povećanja rezolucije i količine podataka (Sl.1.1.) moguće je riješiti na dva načina: programska varijanta kompresije podataka ili sklopovskim povećavanjem propusnog opsega podatkovnog kanala korištenjem prijenosnih tehnologija novije generacije.



Sl. 1.1 Odnos između rezolucije i količine podataka pojedinog okvira.

U mnogobrojnim primjenama postoje zahtjevi na prijenos video sadržaja u realnom vremenu. U novije vrijeme ovo se sve više odnosi i na UHD video. Primjer je dohvaćanje video sadržaja putem *grabber* uređaja za potrebe analize razmatranog video sadržaja. Uređaji koji podržavaju UHD video trenutno su još uvijek u razvoju i potrebno je ispitati mogućnosti prijenosa UHD sadržaja u realnome vremenu. Kako se u radu razmatra uobičajena prijenosna tehnologija (maksimalno dopuštena pojasna širina prijenosnog kanala od 1 Gbit/s), rješenje navedenog problema pokušava se riješiti odgovarajućom programskom podrškom. Budući da prilikom dohvaćanja video sadržaja pomoću *grabber* uređaja najčešće ne smije doći do gubitaka podataka, u radu se kao moguće rješenje razmatra kompresija video sadržaja bez gubitaka. Konkretno, odabrana je *Lempel-Ziv-*

Welch (LZW) kompresijska metoda iz razloga što nudi potencijalno dovoljan stupanj kompresije uzimajući u obzir sadržaj i količinu podataka.

Pod zahtjeve rada spada i korišteni format elemenata slike (engl. *Pixel format*), YUV420P. Navedeni format pogoduje izabranom kompresijskom algoritmu zbog specifičnog pakiranja podataka na način posebnog grupiranja *luma (Y)*, a posebno *chroma (U, V)* podataka svih pojedinih elemenata slike što na kraju rezultira grupacijom sličnih podataka i potencijalno većim postotkom kompresije sadržaja.

Pored implementacije i optimizacije algoritma, glavni cilj rada svodi se na utvrđivanje je li navedeni kompresijski postupak primjenjiv na prijenos video sadržaja visoke rezolucije u realnome vremenu. Fluidni prikaz sadržaja u realnome vremenu zahtjeva prijenos od 50 okvira (engl. *Frame*) u sekundi vremena (engl. *Frames Per Second, FPS*) uključujući cjelokupnu obradu. U cjelokupnu obradu spada dekodiranje, kompresija i slanje okvira na strani poslužitelja te dekompresija i ponovno kodiranje okvira u odgovarajući spremnik na strani klijenta. Kako bi se navedeni zahtjev od 50 FPS-a zadovoljio, u najgorem slučaju, vrijeme cjelokupne obrade pojedinog digitalnog okvira ne smije prelaziti 20ms. Upravo je iz tog razloga vrijeme izvršavanja pojedine programske funkcije od pretežite važnosti. Uzimajući u obzir veličinu 4K UHD okvira od 11.86 MB pri 12 bit-a podataka po elementu slike i propusnost korištene prijenosne tehnologije od 100 MB/s (1Gbit/s) dolazi se do zaključka da pojedini okvir nikako ne smije prelaziti 2MB, odnosno postotak kompresije ne smije biti niži od 83% za YUV420 format elemenata slike.

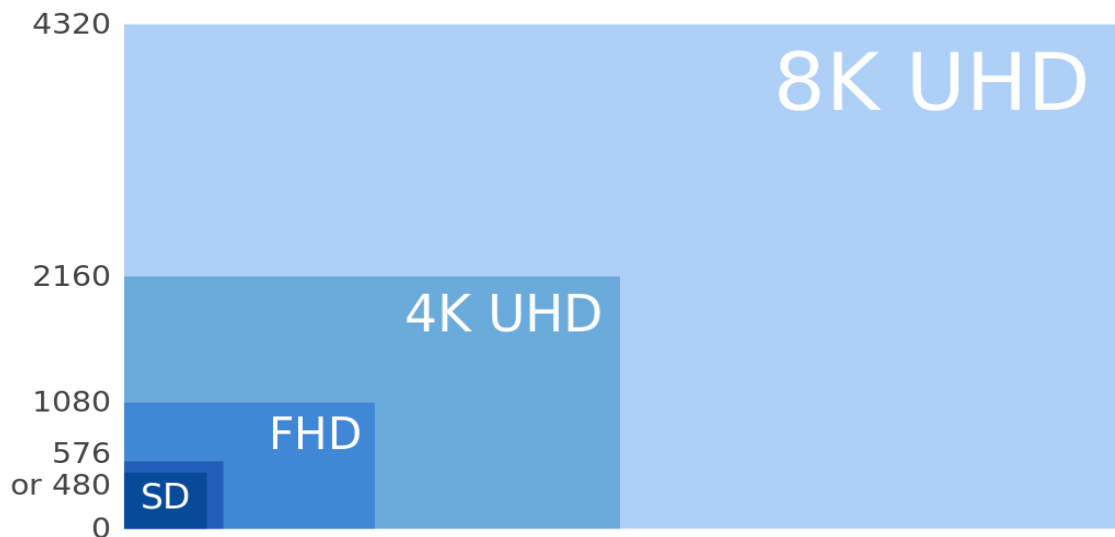
Diplomski rad strukturiran je na način da je u prvom poglavlju utvrđena teorijska podloga korištenog UHDTV-a standarda te njegovih tehničkih specifikacija. Nakon toga objašnjena je kompresija podataka te su pojašnjene dvije glavne vrste: kompresija s gubitkom podataka i kompresija bez gubitaka podataka. Pod istim poglavljem opisan je način rada korištenog kompresijskog algoritma. U idućem poglavlju navedeni su korišteni programski alati te je podrobnije opisana programska implementacija rješenja dok se u zadnjem poglavlju nalaze opisi korištenih testnih sekvenci, tehničke specifikacije sklopovlja na kojemu se predloženo rješenje izvršavalo te konačni rezultati testiranja.

2. UHD STANDARD

Video standard poznat pod skraćenicom UHDTV, donesen je 23.08.2012 godine od strane Internacionalne Telekomunikacijske Unije (engl. *International Telecommunication Union, ITU*). Sam rad na standardu započeo je u *NHK Science & Technical Research Laboratories (NHK)* u Japanu, a dovršila i odobrila ga je ITU, specijalna organizacija Ujedinjenih Naroda direktno odgovorna za sva pitanja vezana za informacijske i telekomunikacijske tehnologije poput globalnog dodjeljivanja radijskog spektra, satelitskih orbita, poboljšavanja globalne telekomunikacijske infrastrukture te upravljanje razvojem i koordinacijom tehničkih standarda. Same tehničke specifikacije UHD standarda definirane su u *Rec. 2020 (ITU-R Recommendation BT.2020)*, poput razlučivosti okvira, broja okvira po sekundi, prostora boja, poduzorkovanja boja te broja bit-ova potrebnih za prikazivanje pojedine komponente elemenata slike. Pred kraj 2012. godine, *Consumer Electronics Association (CEA)* donosi odluku kojom se termin UHD direktno odnosi na ekrane s omjerom visine i širine stranica okvira 16:9, kao i da moraju imati najmanje jedan digitalni ulaz s mogućnošću prijenosa video sadržaja visoke rezolucije.

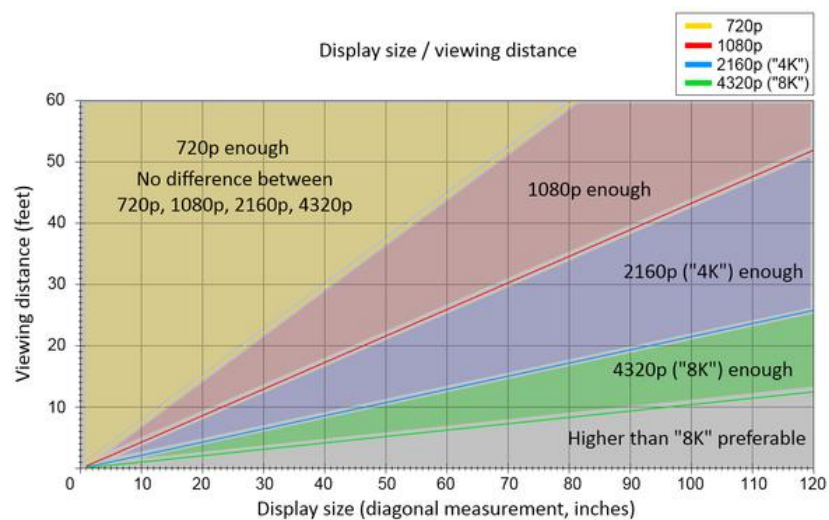
2.1 Rezolucija digitalnog okvira

Svaki video sadržaj sastoji se od niza digitalnih okvira. Rezolucija je veličina koja predstavlja broj elemenata pojedine slike. Veći broj elemenata slike direktno utječe na razlučivost prikaza, odnosno izraženiju jasnoću detalja digitalnog okvira. UHDTV standard prema specifikacijama podržava dvije rezolucije: 3840x2160 (4K UHD) i 7680x4320 (8K UHD) s progresivnim iscrtavanjem linija [1, str.1, Tab.1]. Osnovne oznake su 2160 i 4320 te se njima dodaje slovo „p“ koje označava način iscrtavanja okvira na ekranu. U progresivnom slučaju uređaj prikazuje liniju po liniju elemenata slike na ekranu. Oznake 2160 i 4320 predstavljaju visinu, dok 3840 i 7680 označavaju širinu okvira. Iz prethodno navedenog izvode se popularni nazivi 4K UHD za 2160p, odnosno 8K UHD za 4320p. 4K UHD rezolucija usmjerena je prema komercijalno dostupnim uređajima i široj javnosti te ima dvostruko više horizontalnih i vertikalnih elemenata slike od prethodne *Full High Definition (FHD, 1920x1080)* rezolucije (SI.2.1.). Shodno tome, 8K UHD ima dvostruko više elemenata slike od 4K UHD rezolucije te se u današnje vrijeme gotovo isključivo koristi u filmskoj industriji.



Sl. 2.1 Usporedba rezolucija najčešće korištenih video standarda [2].

Bitna stavka koju treba uzeti u obzir prilikom povećavanja rezolucije okvira i uočavanja detalja poznata je pod nazivom Lechnerova udaljenost. Lechnerov dijagram (Sl.2.2.) prikazuje optimalnu udaljenost gledatelja od ekrana ovisno o rezoluciji slike i veličini ekrana. Udaljavanjem gledaoca od ekrana smanjuje se razlučivost ljudskog oka te gledatelj nije u mogućnosti razaznati detalje koji su inače jasno vidljivi na bližim udaljenostima. Vrijedi i obrnuto, približavanjem gledaoca ekranu potrebna je veća rezolucija, odnosno više manjih detalja opisanih elementima slike kako bi doživljaj gledatelja i kvaliteta slike ostala ista. Iz dijagrama na slici 2.2 vidljivo je da ne postoji razlika između *High Definition (HD)*, FHD, 4K UHD i 8K UHD rezolucije pri određenim udaljenostima i veličinama ekrana (žuta zona).



Sl. 2.2 Lechnerov dijagram [3].

2.2 Broj slikovnih okvira u sekundi vremena

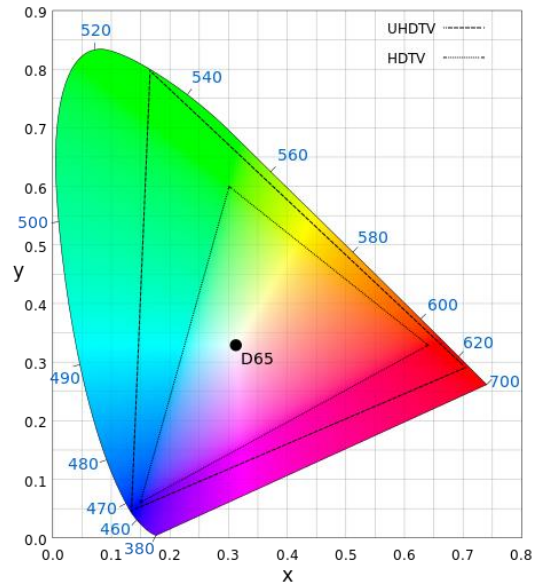
Ljudsko oko fizički je ograničeno s mogućnošću razaznavanja do maksimalno 48 različitih bljeskova svjetlosti u sekundi, odnosno 48 različitih promjena okvira [4, str.24]. Primjenjujući navedenu tvrdnju, dolazi se do zaključka da će povećanje frekvencije prikaza okvira iznad 48 FPS-a rezultirati prividnom fluidnom slikom. Pod prividnom fluidnom slikom misli se na povezivanje svih statičnih okvira u jednu cjelinu unutar ljudskog vidnog sustava, uzimajući u obzir da se pojedini okviri razlikuju jedan od drugoga. Fluidnost slike, osim o frekvenciji prikaza, ovisi i o razlici susjednih okvira. Što se okviri više razlikuju jedan od drugoga to će biti potrebna veća frekvencija prikaza da se postigne ista fluidnost, i obrnuto. Standardom je dopušteno progresivno prikazivanje pri frekvenciji od: 120P, 119.88P, 100P, 60P, 59.94P, 50P, 30P, 29.97P, 25P i 24P [1, str.3, Tab.2]. Prema slici 2.3. veća frekvencija rezultira fluidnijom slikom i prirodnijim prikazom dinamičkih kretnji dok s druge strane manja frekvencija prikaza daje dojam da je slika mutna, zaostaje i treperi.



Sl. 2.3. Razlika između različitih frekvencija prikaza okvira [5].

2.3 Prostor boja

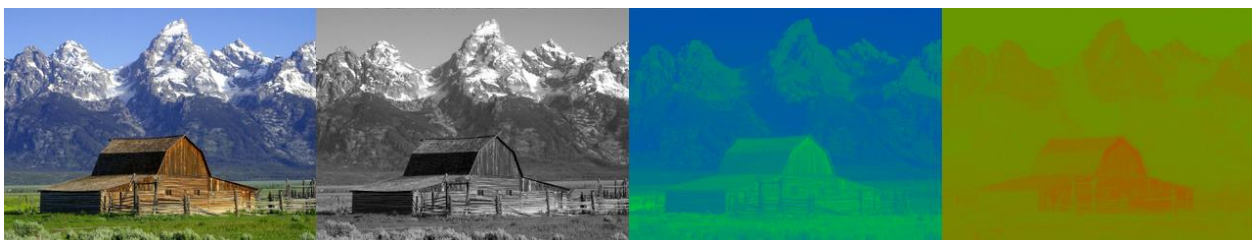
U prvoj polovici 20. stoljeća definirana je kvantitativna veza između valne duljine svjetlosti i ljudske fizičke mogućnosti percepcije boja određenog spektra [6]. Prostor boja koje ljudskog oko ima sposobnost percipirati nazvan je CIE 1931 prema organizaciji *Commision Internationale de l'eclairage* (engl. *International Commission on Illumination, CIE*) koja ga je donijela na snagu 1931. godine. Prema slici 2.4. pokrivenost CIE 1931 prostora boja dvostruko je veća kod UHDTV standarda (75.8%) nego kod HDTV-a (35.9%). Budući da s 8 bit-a podataka nije moguće reprezentirati spektar boja pokriven s CIE 1931 prostorom boja, povećava se i broj bit-a potrebnih za opisivanje pojedinog elementa slike, a samim time povećava se i ukupna količina podataka koje je potrebno obraditi, prenijeti te na posljetku pohraniti.



Sl. 2.4. CIE 1931 prostor boja i pokrivenost UHDTV standarda [7].

2.3.1 Y'UV prostor boja

Y'UV prostor boja nastao je iz potrebe implementacije televizijskog prijenosa boje bez mijenjanja već postojeće crno-bijele prijenosne infrastrukture. Luminantna (lat. *Lumen*) komponenta (Y') bila je prisutna u postojećoj prijenosnoj infrastrukturi kao razlika između crne i bijele dok su krominantne (grč. *Chroma*) komponente (U,V) naknadno dodane kao razlike između crveno-zelene i žuto-plave boje (Sl.2.5.). Ovakav pristup omogućava prikaz signala koji sadrži krominantne komponente na postojećoj crno-bijeloj televizijskoj infrastrukturi bez potrebe za implementacijom dodatnog sklopovlja.

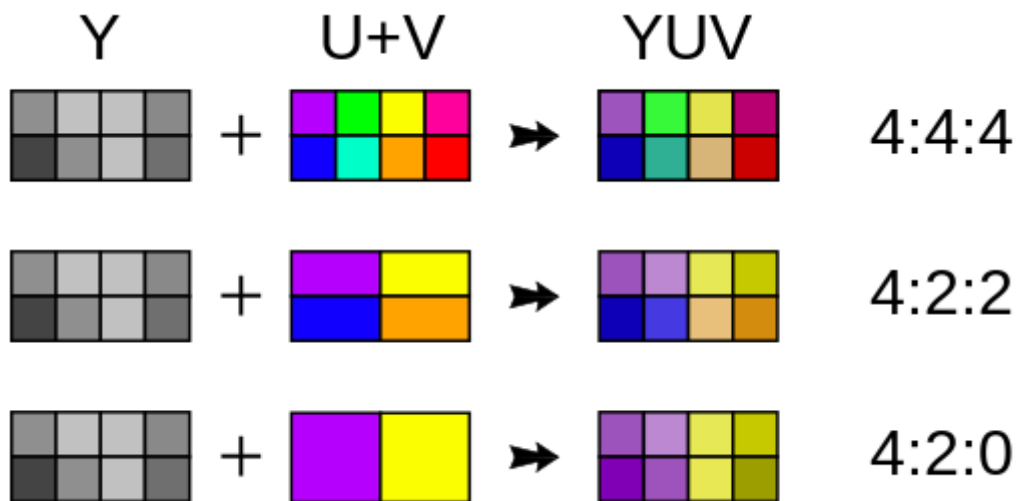


Sl. 2.5. Prikaz pojedinačnih komponenata slike [8].

Y'UV prostor boja stekao je veliku popularnost iz razloga što se prilikom kodiranja okvira, koristeći navedeni prostor boja, u obzir uzima ljudska percepcija, odnosno veća osjetljivost na promjenu intenziteta svjetlosti nego na promjenu boje.

2.4 Poduzorkovanje boja

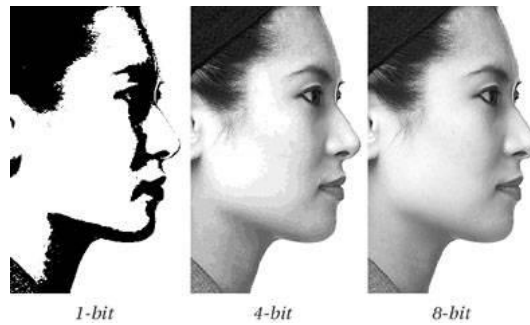
Kao što je prethodno navedeno, ljudski vidni sustav znatno je osjetljiviji na promjenu intenziteta svjetlosti nego na promjenu boje te se navedeno obilježje može iskoristiti prilikom optimiziranja pojase širine prijenosnog kanala. Navedena spoznaja omogućava uzorkovanje manjeg broja krominantnih podataka što rezultira smanjenjem ukupne količine podataka potrebnih za reprezentiranje pojedinog okvira. Smanjenjem količine podataka smanjuje se i potrebna pojasna širina prijenosnog kanala dok je razlika u kvaliteti gotovo neprimjetna. Kod korištenog YUV420P formata, signal je podijeljen na jednu luminantnu (Y) i dvije krominantne komponente (U i V). Ovisno o formatu, odnosno veličini uzoraka, mijenja se i ukupna količina podataka okvira. UHD standard dopušta korištenje RGB i YCbCr modela formata 4:4:4, 4:2:2 i 4:2:0 (Sl.2.6.). Korištenjem YUV420P formata količina podataka potrebnih za opisivanje okvira smanjuje se za 50% u odnosu na YUV444 format samim time što se uzorkuje samo četvrtina krominantnih podataka.



Sl. 2.6. UHD TV standardom dopušteni formati elemenata slike [9].

2.5 Dubina boja

Povećanjem pokrivenosti prostora boja povećava se i količina podataka kojima će pojedini slikovni okvir biti opisan. Budući da je pokrivenost boja UHD standarda veća, potrebno je i više podataka kako bi se pojedina boja prikazala (engl. *Bit Depth*). Prethodni FHD standard definirao je 8-bitu podataka po komponenti boje, dok za 4K UHD prostor boja to nije dovoljno. UHD standardom količina podataka proširena je na 10, odnosno 12 bita po komponenti boje. Povećanom dubinom boja dobiva se mogućnost prikaza širokog spektra boja pokrivenog CIE 1931 prostorom što rezultira kvalitetnijom i realističnijom slikom (Sl.2.7.).



Sl. 2.7. Kvaliteta slike u ovisnosti o dubini korištenih boja [10].

Tako je za 4K UHD okvir s dubinom boja od 10 bit-a i YUV420P formatom elemenata slike potrebno 14,832 MB dok je za okvir istih specifikacija, ali s dubinom boje od 8 bit-a, potrebno 11,865 MB što je gotovo 3 MB podataka manje. Ako se okviri ne poduzorkuju, dolazi se do zapanjujućih 29,663 MB za 10 bit-ni 4K UHD okvir, odnosno 23,730 MB za 8 bit-ni 4K UHD okvir.

2.6 Omjer visine i širine okvira

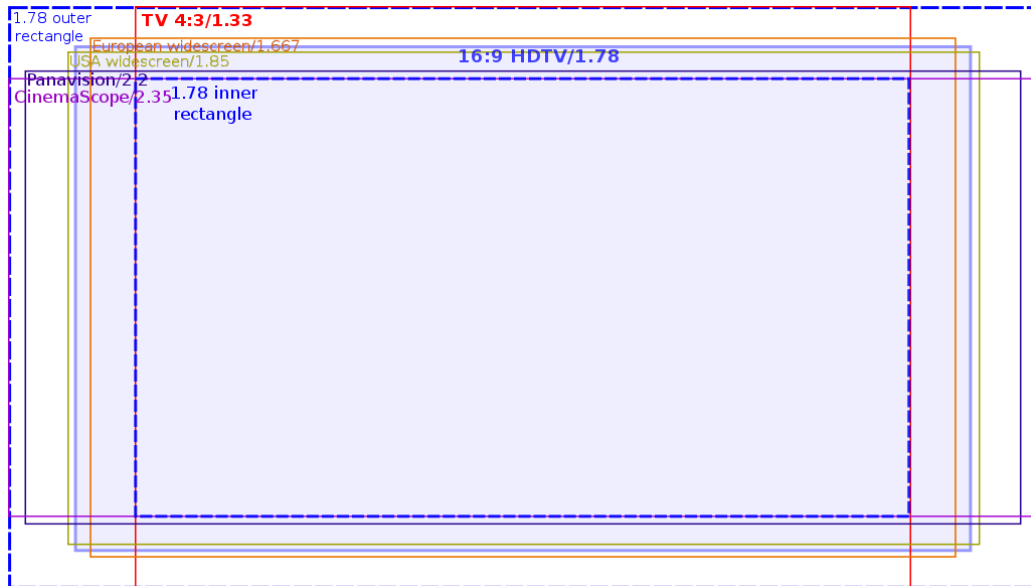
Omjer između širine i visine slike (engl. *Aspect ratio*) mijenjao se tijekom godina. Dva uobičajena omjera koja se i danas koriste su 4:3 (1.33:1) koji je prozvan univerzalnim video formatom 20. stoljeća te 16:9 (1.77:1) (Sl.2.6.) koji se primjenjuje u standardima visoke rezolucije te pokriva sve rezolucije obuhvaćene UHDTV standardom. Osim što je internacionalni standardni format slike za UHDTV, standardni je format i za HDTV kao i za neke analogne široko-ekranske standarde tipa PALplus.



Sl. 2.8. Primjer 16:9 (1.77:1) formata okvira [11].

Omjer okvira 16:9 izveden je iz rada Dr. Kerns H. Powers-a, člana *Society of Motion Picture and Television Engineers* grupe. Nastao je kao kompromis između nekolicine prethodno

korištenih omjera okvira poput 1.33:1, 1.66:1, 1.85:1, 2.2:1 i 2.35:1. Kada se prethodno navedeni omjeri poravnaju s obzirom na njihovo središte dobiva se pravokutnik koji je pokriven svim omjerima. Odnos između širine i visine dobivenog pravokutnika približno je 1.770:1 što je veoma blizu 16:9 formata (SI.2.9.).



SI. 2.9. Prikaz HDTV pravokutnika dobivenog kombinacijom prethodnih omjera [12].

3. KOMPRESIJA SLIKE

Postupci za kompresiju slike igraju važnu ulogu u raznim multimedijским primjenama koje obuhvaćaju neku vrstu prijenosnog, odnosno komunikacijskog kanala. Neophodna je iz razloga što se količina podataka koju je potrebno prenijeti povećava svakim novim standardom. Kako je pojasna širina ograničena za određene prijenosne tehnologije, da bi se zadovoljili zahtjevi potrebno je na neki način smanjiti količinu podataka koja se prenosi. Kompresija upravo to radi, smanjuje potrebnu pojasnu širinu prijenosnog kanala na način da smanjuje količinu podataka potrebnih za reprezentiranje pojedinog okvira.

U kompresiju slike spadaju sve tehnike kodiranja informacija kojima se konačna količina podataka smanjuje u odnosu na izvornu. Tehnike za kompresiju slike mogu se podijeliti u dvije glavne grupe: kompresija bez gubitaka i kompresija s gubitcima. Kompresija bez gubitaka smanjuje količinu podataka na način da uklanja statičku zalihost koju sadrži digitalna slika te se koristi u slučajevima kada slika sadrži važne informacije koje bi se mogle izgubiti u samom postupku kompresije. S druge strane, kompresija s gubitcima identificira i uklanja redundantne podatke te se koristi u slučajevima kada se može tolerirati određeni postotak izgubljenih podataka.

3.1 Kompresija slike s gubitcima

Metoda kompresije slike s gubitcima posebno je pogodna za slike prirode gdje je gubitak kvalitete slike gotovo neprimjetan ljudskom oku, a sažimanje podataka ogromno. Prilikom korištenja same kompresije postoji mogućnost određivanja krajnje kvalitete komprimirane slike. Što je kvaliteta slike veća to je kompresija manja, i obrnuto. Kod veoma visokog stupnja kompresije dolazi do pojavljivanja smetnji koje se očituju kao digitalna izobličenja poput zvonjenja slike, ocrtavanja, *posterization-a*, poduzorkovanja i pojavljivanja kvadratića.

Metode kompresije slike s gubitcima rade na principu da su neki podatci na slici manje važni ili da su određeni podatci izvan granica ljudskog vidnog sustava, odnosno da ih ljudsko oko ne može primijetiti. Ovi postupci podrazumijevaju takvu pretvorbu kod koje se namjerno bespovratno gubi dio podataka, a za uzvrat dobiva se zapis koji zauzima manje memorije. Što je odabrana kvaliteta slike manja to konačni zapis zauzima manje memorije, i obrnuto. Svakim slijedećim korakom kompresije slika će se sve više i više degradirati (Sl.3.1.).



Sl. 3.1. *Primjer kompresije slike s gubitcima [13].*

Iako ovakve metode uzrokuju manje ili više uočljivu degradaciju, one omogućuju daleko veće omjere kompresije nego metode kompresije bez gubitaka. Najpoznatija metoda za kompresiju slike s gubitcima je *JPEG* (engl. *Joint Photographic Experts Group*) koja je osmišljena tako da koristi nesavršenosti ljudskog oka s činjenicom da se okom bolje primjećuju male razlike u svjetlini nego u boji (20:1 kompresija sa svim bojama bez vidljivih gubitaka informacija).

3.2 Kompresija slike bez gubitaka

Kod ove metode ne dolazi do gubitka podataka nakon dekompresije te se zbog toga koristi u slučajevima kada je egzaktnost podataka prije kompresije i nakon dekompresije od pretežite važnosti poput arhiviranja, raznih medicinskih potreba, tehničkih crteža, stripova i ostalih primjena. Dvije stvari koje većina algoritama za kompresiju bez gubitaka ima zajedničko su kreiranje statističkog modela ulaznih podataka i korištenje generiranog statističkog modela za mapiranje ulaznih podataka na taj način da će ulazni podatci, koji se učestalo pojavljuju, biti zamijenjeni kraćom sekvencom podataka. Na ovom principu radi i *LZW* kompresijski algoritam koji je iskorišten u ovom radu.

3.3 Lempel-Ziv-Welch (LZW) algoritam

LZW algoritam nastao je iz ideje koje su Abraham Lempel i Jacob Ziv objavili 1978. godine pod nazivom *LZ77* (*LZ1*) i *LZ78* (*LZ2*). Terry Welch je 1984. godine objavio poboljšanu verziju *LZ78* algoritma koji je kasnije poprimio današnji naziv, *LZW*. Izvorni kod *LZW* algoritma otvoren je javnosti 2003. godine kada je istekao patent u Ujedinjenim Američkim Državama te 2004. godine u Ujedinjenom Kraljevstvu, Francuskoj, Njemačkoj, Italiji, Japanu i Kanadi. U svoje vrijeme korišten je u raznim programskim alatima te je postao osnovni uslužni program u *Unix*-ovom sistemu *Circa*. Nakon 1986. godine izostajao je iz novijih *Unix*-ovih distribucija iz razloga što je bio zaštićen patentima te iz razloga što su kreirani novi algoritmi s poboljšanim postotkom i

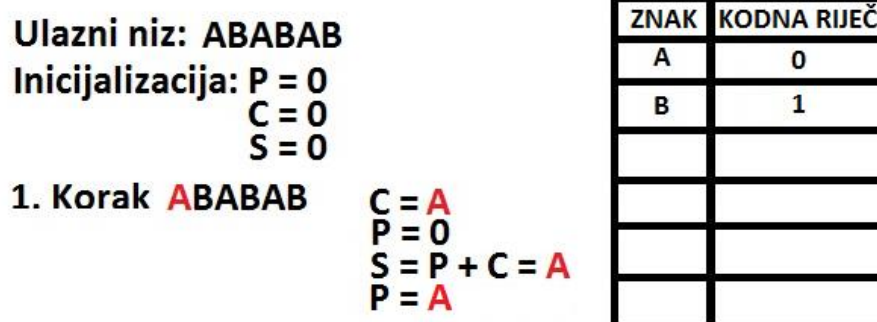
vremenom kompresije. LZW algoritam i njegove modifikacije osnova su programa za kompresiju i dekompresiju poput ZIP, GZIP i drugih. Razlog visoke popularnosti algoritma i njegovih modifikacija pripisuje se njegovoj brzini izvršavanja, zadovoljavajućem postotku kompresije, kompresiji bez gubitaka te mogućnosti primjene na bilo kakav oblik podataka zapisan kao jednodimenzionalan niz.

3.3.1 Način rada

LZW je vrsta kompresijskog algoritma bazirana na kodnoj tablici, odnosno rječniku. Prije početka same kompresije dolazi do inicijalizacije rječnika koji može imati do 4096 kodnih riječi. Rječnik se inicijalizira na način da se svim znakovima iz ASCII tablice dodjeljuje kodni broj. Prvih 256 kodnih riječi u rječniku sadrže brojeve od 0 do 255 što odgovara broju znakova ASCII tablice. Posebne kodne riječi nalaze se na 256. i 257. mjestu u tablicu, a to su kod za čišćenje i kod za označavanje kraja datoteke (engl. *End of information, EOI*). Na preostalim lokacijama u rječniku (258-4095) nalaze se kodne riječi u kojima se spremaju nizovi bitova koji se često pojavljuju u okviru. Prvih 512 kodnih riječi u tablici sastoje se od 9 bit-a, sljedećih 512 od 10 bit-a, riječi na lokacijama od 1024 do 2047 sastoje se od 11 bit-a, a preostale riječi imaju po 12 bit-a. Ovisno o rednom broju kodne riječi povećava se broj bit-ova potrebnih za pojedinu kodnu riječ.

Kreiranje, odnosno popunjavanje rječnika izvršava se tijekom kompresije slike. Kako algoritam napreduje kroz niz podataka, tako se popunjava i rječnik. Na istom principu radi i dekompresijski dio algoritma, ali obrnuto. Slikovni okviri koji dolaze kao ulazni podatci tretiraju se kao dugački jednodimenzionalni nizovi.

Na početku izvršavanja algoritma dolazi do inicijalizacije varijabli na nulu u koje će se spremati trenutačni (C) i prefiks (P) bajt podataka, te varijable u koju će se spremiti ulančani niz (S). U svakom koraku algoritma učitava se po jedan bajt podataka slikovnog okvira. Prema slici 3.2., prvi učitani bajt sprema se u trenutačnu varijablu C te se spaja s varijablom prefiksa P koja je u ovom slučaju još uvijek 0 ($S = P + C$). Ako se sadržaj varijable S već nalazi u kodnoj tablici prefiks se ažurira ($P = S = A$).



Sl. 3.2. Prvi korak rada LZW algoritma.

U sljedećem koraku učitava se sljedeći bajt ulaznih podataka te se ulančava s prethodnom vrijednosti varijable P u varijablu S (Sl.3.3). U ovom slučaju sadržaj varijable S (AB) ne nalazi se u kodnoj tablici. Tablica se dopunjava dodavanjem sadržaja varijable S na kraj kodne tablice, sadržaj P varijable šalje se u izlazni niz te se P inicijalizira na vrijednost posljednjeg pročitano bajta podataka.

2. Korak ABABAB $C = B$
 $P = A$
 $S = P + C = AB$
 $P = B$

Izlazni niz: 0

ZNAK	KODNA RIJEČ
A	0
B	1
AB	2

Sl.3.3. Drugi korak rada LZW algoritma.

Postupak se dalje ponavlja. Učitava se sljedeći bajt (A), ulančava se s prethodnim prefiksom u varijablu S te se provjerava kodna tablica i poduzimaju se određene mjere (Sl.3.4.).

3. Korak ABABAB $C = A$
 $P = B$
 $S = P + C = BA$
 $P = A$

Izlazni niz: 01

ZNAK	KODNA RIJEČ
A	0
B	1
AB	2
BA	3

Sl. 3.4. Treći korak rada LZW algoritma.

U sljedeća dva koraka dolazi se do glavnog cilja LZW kompresijskog algoritma, a to je zamijeniti veću sekvencu podataka s manjom. Učitavanjem sljedećeg bajta u varijablu C te ulančavanjem s prethodnim prefiksom P (A) u varijablu S, dobiva se sadržaj AB. Iz tablice je vidljivo da se sadržaj varijable S (AB) već nalazi u tablici pod kodnom riječi 2 te se P varijabla ažurira ($P = S = AB$).

4. Korak ABABAB $C = B$
 $P = A$
 $S = P + C = AB$
 $P = AB$

ZNAK	KODNA RIJEČ
A	0
B	1
AB	2
BA	3

Sl. 3.5. Četvrti korak rada LZW algoritma.

Budući da je u prefiks varijablu P upisan sadržaj AB koji se već nalazi u kodnoj tablici, u sljedećem koraku sadržaj AB biti će zamijenjen s kodnom riječi 2. Prema slici 3.6. dolazi do čitanja sljedećeg bajta podataka (A) te ulančavanja s prethodnim prefiksom (AB) u varijablu S. Ulančani sadržaj (ABA) ne nalazi se u kodnoj tablici što znači da će u izlazni niz biti poslana kodna riječ prefiksa (2) te će se na taj način smanjiti ukupna količina podataka.

5. Korak ABABAB $C = A$
 $P = AB$
 $S = P + C = ABA$
 $P = A$

Izlazni niz: 012

ZNAK	KODNA RIJEČ
A	0
B	1
AB	2
BA	3
ABA	4

Sl. 3.6. Završni korak rada LZW algoritma.

Postupak se ponavlja sve dok se ne dođe do kraja ulaznog niza kada se ubacuje kodna riječ za kraj datoteke, EOF. U slučaju da se kodna tablica napuni do kraja, a kompresija još nije završena, u izlazni niz upisuje se kodna riječ za čišćenje tablice te se tablica ponovno kreira i inicijalizira kao na početku izvršavanja algoritma.

Kada je potrebno izvršiti dekompresiju nad komprimiranim sadržajem, nije potrebno slati kodnu tablicu budući da se tablica rekonstruira tijekom dekompresije na isti način, ali obrnutim redoslijedom kojim se kreirala tijekom kompresijskog dijela. Ova značajka ujedno je i jedan od najbitnijih faktora popularnosti same LZW kompresije.

Vrijeme izvršavanja kompresije najviše zavisi od sličnosti sadržaja koji se komprimira te o načinu implementacije kodne tablice. Način na koji je implementirano pretraživanje tablice ima veliku ulogu u ukupnom vremenu izvršavanja algoritma te se iz tog razloga najčešće koriste *hash* tablice i *hash* funkcije pretraživanja. Popularnost *hash* tablica i funkcija proizlazi iz činjenice da je prosječno vrijeme izvršavanja funkcije za pretraživanje tablice $O(1)$. Najgori slučaj koji se u praksi može susresti je $O(N)$. Do vremena $O(N)$ dolazi kada se vrijednosti *hash*-iraju na ista mjesta u tablici. Razlog zbog kojega se uzima prosječno vrijeme $O(1)$ proizlazi iz činjenice da je vjerojatnost pojavljivanja najgoreg slučaja vrlo mala.

4. IMPLEMENTACIJA LZW KOMPRESIJE ZA POTREBE PRIJENOSA

Za potrebe analize LZW kompresije u svrhu prijenosa odabran je C programski jezik dok je za video koder (engl. *Video encoder*) na klijentskoj strani odabran C++ programski jezik. Cijela analiza zasnovana je na programskom alatu pod nazivom *Fast-Forward-mpeg* (FFmpeg) koji nudi gotova rješenja za rukovanje multimedijским sadržajem te TCP/IP protokolu namijenjenom za simulaciju komunikacije između klijenta i poslužitelja.

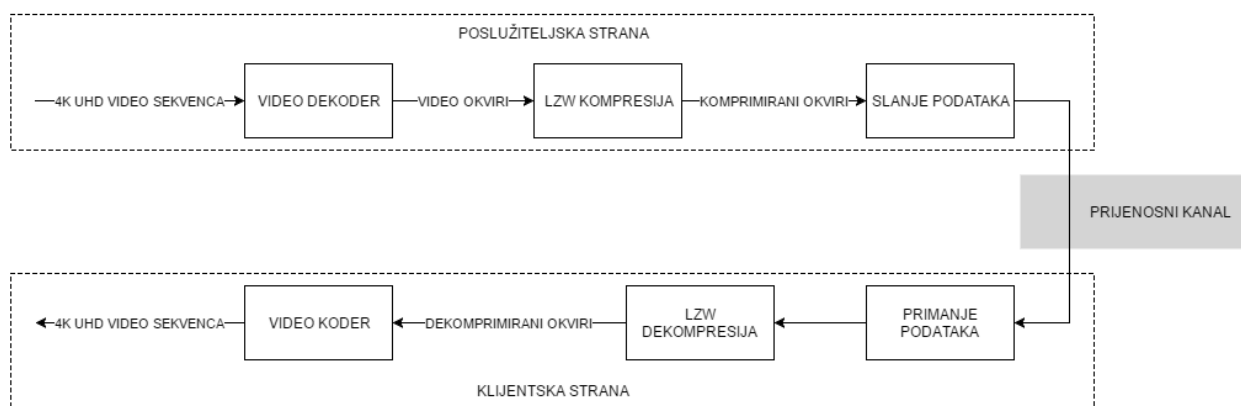
4.1 FFmpeg

FFmpeg je gotovo programsko rješenje koje nudi podršku za rukovanje multimedijским sadržajem u obliku izvršne datoteke. Sami izvorni kod pisan je u C programskom jeziku što nudi mogućnost prevođenja i izgradnje izvršne datoteke za bilo koji operacijski sustav uz minimalne izmjene. Budući da za potrebe ovoga rada nisu potrebne sve funkcionalnosti FFmpeg-a, iz izvornog koda [14] izdvojene su i prilagođene samo funkcije za rukovanje multimedijским sadržajem te funkcije vezane za LZW kompresiju i dekompresiju.

Iz razloga što je FFmpeg gotovo programsko rješenje u obliku izvršne datoteke te su svi dijelovi izvornog koda usko povezani, ne postoji mogućnost direktne implementacije u nezavisni projekt. Uzimajući u obzir gore navedeno, kreirane su i prilagođene biblioteke temeljene na izvornim funkcijama FFmpeg-a kako bi se olakšalo korištenje potrebnih funkcionalnosti. Točnije, 3 biblioteke su prilagođene od kojih dvije (*bswap-h*, *bytestream.h*) služe za rukovanje podacima ovisno o korištenoj mikroprocesorskoj arhitekturi. Treća biblioteka (*FfmpegSimplified.h*) objedinjuje prethodne dvije te nudi sučelje prema potrebnim funkcionalnostima FFmpeg-a. Najvažnije funkcije koje *FfmpegSimplified.h* implementira vezane su za inicijalizaciju i prikupljanje podataka potrebnih za pravilno funkcioniranje te slanje početnih informacija klijentskoj strani, poput veličine slikovnog okvira kojega klijentska aplikacija prima.

4.2 Implementacija izrađenog programskog rješenja

Programsko rješenje podijeljeno je u funkcionalne cjeline od kojih su najznačajnije slanje i primanje podataka korištenjem TCP/IP protokola, rukovanje multimedijским sadržajem te LZW kompresija i dekompresija podataka (Sl.4.1.).



Sl. 4.1. Blok prikaz programskog rješenja.

Budući da FFmpeg pored ostalih funkcionalnosti implementira i razne algoritme za kompresiju i dekompresiju, između kojih se nalazi i LZW, izvorni kod LZW kompresije i dekompresije prilagođen je i implementiran u postojeći projekt. Kod je optimiziran u smislu smanjivanja vremena izvršavanja pojedinih funkcija i izbacivanja redundantnih dijelova koda. Nakon svake optimizacije i izbacivanja dijela koda provjerena je funkcionalnost kako se ne bi narušio integritet same programske cjeline.

Prevođenjem (engl. *Compile*) izvornog koda generiraju se dvije izvršne datoteke, jedna za poslužiteljsku stranu i jedna za klijentsku stranu. Pokretanje datoteka izvršava se preko terminala, odnosno komandne linije (engl. *Command prompt*, *CMD*). Prilikom pokretanja izvršne datoteke poslužiteljske strane potrebno je upisati određene parametre preko kojih se predaju putanja do video sekvence te broj priključka (engl. *Port*) na kojemu se očekuju veze klijentske aplikacije.

4.2.1 Simulacija klijentske aplikacije i poslužitelja

Klijent i poslužitelj kreirani su lokalno kao posebne instance koje slušaju na određenom priključku. Pokretanjem programa kreira se server koji osluškuje spajanje klijenata. Nakon što se klijent uspješno spoji na poslužitelj, program započinje s radom. Kao ulazne parametre poslužitelj prima broj priključka na kojemu će slušati na konekciju klijenta dok je klijentu potreban broj priključka na koji će se spojiti te lokalna IP adresa poslužitelja. Slanje podataka između poslužitelja i klijenta radi na principu rukovanja (engl. *Handshaking*). Poslužitelj neće slati sljedeći paket podataka sve dok mu klijent ne javi da je prethodni paket obrađen. Funkcije potrebne za kreiranje klijentske i poslužiteljske strane te početnu inicijalizaciju, objedinjene su u posebnu biblioteku pod nazivom *TCPsimplified.h*. Budući da se kao parametar zadaje broj priključka te lokalna IP adresa poslužitelja, moguće je mijenjati na kojem priključku poslužitelj sluša odnosno na koji priključak se klijent spaja bez dodatnog mijenjanja izvornog koda. Poslužitelj je

programski ograničen na jednu konekciju koju može primiti od klijenta kako ne bi došlo do slučajnog upletanja drugih veza.

4.2.2 Mjerenje vremena izvršavanja

Vrijeme izvršavanja pojedine programske funkcije od pretežite je važnosti za zadovoljavanje vremenskih zahtjeva prijena komprimiranih video podataka u realnome vremenu. Uzimajući u obzir da je vremensko područje izvršavanja pojedinih funkcija reda mikro sekundi, korištena je 64-bitna cjelobrojna varijabla s predznakom kako bi izmjerena vremena bila što preciznija.

Mjerila se razlika između sistemskog vremena prije početka izvršavanja pojedine funkcije te sistemskog vremena nakon izvršavanja pojedine funkcije. Na taj način uvodi se minimalna nepreciznost. Pod minimalnom nepreciznosti smatra se vrijeme potrebno za izvršavanje funkcija dohvaćanja sistemskog vremena (engl. *QueryPerformanceCounter*, *QPC*). Vrijeme izvršavanja QPC funkcije nalazi se između 0.8 – 1 μ s. Uzimajući u obzir navedeno, nepreciznost samog mjerenja je + 0.8 – 1 μ s po pojedinom izvršavanju QPC funkcije. Izmjerena vremena zapisuju se i nadodaju u 64-bitne cjelobrojne varijable te se naposljetku dijele s brojem obrađenih okvira kako bi se dobilo prosječno vrijeme izvršavanja pojedinih funkcija. Isto tako, vrijeme izvršavanja pojedinih funkcija po okviru spremaju se u .csv (engl. *Comma-separated values*) datoteku kako bi se kasnije olakšao rad sa samim podatcima kao i analiza dobivenih rezultata.

5. ANALIZA REZULTATA LZW KOMPRESIJE ZA POTREBE PRIJENOSA

Kako bi se dobio uvio u efikasnost predloženog rješenja mjerena su vremena izvršavanja svih funkcija čije izvršavanje je bilo neophodno nad pojedinim okvirima. Tako su mjerena vremena potrebna za dekodiranje okvira, inicijalizaciju pojedinog konteksta okvira, promjenu veličine okvira, kopiranja okvira u međuspremnik, kompresiju međuspremnika, slanje međuspremnika te klijentsko vrijeme potrebno za obradu međuspremnika. Vrijeme koje je potrebno klijentu da obradi paket podataka po specifikaciji nije od važnosti te se iz tog razloga nije uzimalo u obzir. Prosjek navedenih vremena računao se na temelju broja obrađenih okvira. Isto tako mjeren je pojedinačan i prosječan te minimalan i maksimalan postotak kompresije okvira kao i veličina međuspremnika prije same kompresije. Na temelju dobivenih rezultata izrađeni su grafovi i tablice koje prikazuju odnose između određenih veličina kao i razliku između rezultata ovisno o rezoluciji analizirane video sekvence.

5.1 Tehničke specifikacije sklopovlja

Programsko rješenje izvršavalo se na računalu sljedećih specifikacija s uobičajenim postavkama sklopovlja:

- CPU: Intel Core i7-4790 @ 3.60GHz
- RAM: Kingston 2 x 8 GB @ 1600MHz (Dual Channel)
- HDD: Toshiba DT01ACA100 (7200 RPM, 32MB Cache, SATA 6Gb/s)
- MBO: MSI H81M-E33 (MS-7817)

Pod uobičajenim postavkama sklopovlja smatra se da postavke poput frekvencije rada pojedinog sklopa nisu mijenjane od tvorničkih standarda kojima se garantira stabilan rad.

5.2 Testne video sekvence

Testiranje učinkovitosti kompresije provodilo se na dvije različite rezolucije kako bi se dobiveni rezultati mogli usporediti, FHD i 4K UHD. Raznolikost je osigurana korištenjem 6 video sadržaja s različitim vrijednostima prostorne aktivnosti. Za svaku rezoluciju pojedinog sadržaja preuzete su dvije sekvence pri čemu je jedna prethodno komprimirana algoritmom s gubitkom podataka dok je druga preuzeta u neobrađenom formatu. Ovakvim pristupom moguće je testirati algoritam na sekvencama koje su već prethodno komprimirane te rezultate usporediti s rezultatima dobivenim nad sekvencama u neobrađenom formatu bez prethodne kompresije. Sve video sekvence preuzete su sa službene stranice istraživačke grupe *Tampere University of Technology*

[15]. Dio sekvenci koje su prethodno komprimirane direktno su preuzete u *MPEG-4 Part 14 (MP4)* formatu dok su sekvence bez prethodne kompresije preuzete u YUV formatu te su naknadno kodirane u MP4 video spremnik bez ikakve kompresije ili gubitaka podataka. Zbog potreba testiranja predloženog rješenja iskorištena je izvršna datoteka FFmpeg-a pri dubini boja od 8 bit-a po pojedinom elementu slike. Kvaliteta naknadno kodiranih video sekvenci provjerena je uspoređivanjem s preuzetim sekvencama u YUV formatu koristeći *MSU Video Quality Measurement Tool* [16] koji kao izlaz vraća postotak podudaranja okvira. Za sve naknadno kodirane MP4 sekvence podudaranje sa sekvencama u neobrađenom formatu iznosi 100%. Isto tako izračunata je prostorna aktivnost sekvenci na temelju prosjeka standardne devijacije okvira filtriranih Sobelovim filtrom. Prostorna aktivnost testnih sekvenci dana je u tablici 5.1. Može se primijetiti da korištene video sekvence imaju različiti stupanj prostorne aktivnosti. Na primjer, „*BeautyFHD*“ je veoma statičan prikaz žene s vrlo velikom sličnosti okvira dok je „*HorseRaceFHD*“ vrlo dinamičan video koji prikazuje utrku konja u krupnom planu s velikim razlikama između okvira.

Tab. 5.1. *Prostorne aktivnosti pojedinih video sekvenci.*

Ime sekvence	Prostorna aktivnost
BeautyFHD	0,075455
JockeyFHD	0,128305
HoneyBeeFHD	0,149311
BosphorusFHD	0,157913
YachtRideFHD	0,209785
HorseRaceFHD	0,237808
Beauty4K	0,061144
Jockey4K	0,077726
HoneyBeek4K	0,08564
Bosphorus4K	0,098393
YachtRide4K	0,148303
HorseRace4K	0,148761

Prema tablici 5.2. vidljiva je razlika između prethodno komprimiranih preuzetih sekvenci i sekvenci koje su naknadno kodirane bez kompresije. Podudaranje okvira, izračunato pomoću *MSU Video Quality Measurement Tool*-a, izrazito je nisko što dokazuje da su sekvence komprimirane algoritmom s gubitkom podataka.

Tab. 5.2 Prikaz podudaranja komprimiranih i nekomprimiranih sekvenci.

Redni broj	Preuzete sekvence u neobrađenom formatu (.yuv) naknadno kodirane FFmpeg-om bez kompresije (.mp4)	Preuzete sekvence prethodno komprimirane (.mp4)	Podudaranje [%]
1	Beauty4KLossless.mp4	Beauty4KLossy.mp4	27,3
2	Bosphorus4KLossless.mp4	Bosphorus4KLossy.mp4	28,04
3	HoneyBee4KLossless.mp4	HoneyBee4KLossy.mp4	38,34
4	HorseRace4KLossless.mp4	HorseRace4KLossy.mp4	19,63
5	Jockey4KLossless.mp4	Jockey4KLossy.mp4	21
6	YachtRide4KLossless.mp4	YachtRide4KLossy.mp4	21,92
7	BeautyFHDLossless.mp4	BeautyFHDLossy.mp4	46,5
8	BosphorusFHDLossless.mp4	BosphorusFHDLossy.mp4	48,51
9	HoneyBeeFHDLossless.mp4	HoneyBeeFHDLossy.mp4	49,84
10	HorseRaceFHDLossless.mp4	HorseRaceFHDLossy.mp4	43,67
11	JockeyFHDLossless.mp4	JockeyFHDLossy.mp4	48,6
12	YachtRideFHDLossless.mp4	YachtRideFHDLossy.mp4	41,74

Predloženo rješenje ispitano je na sveukupno 24 video sekvence. Kako bi se dobila valjana statistika, program se pokretao 30 puta nad svakom pojedinom sekvencom. Uzimajući u obzir sve navedeno te 600 slikovnih okvira po pojedinoj testnoj sekvenci, predloženo rješenje simulirano je na sveukupno 432 000 slikovnih okvira.

5.3 Rezultati testiranja programskog rješenja

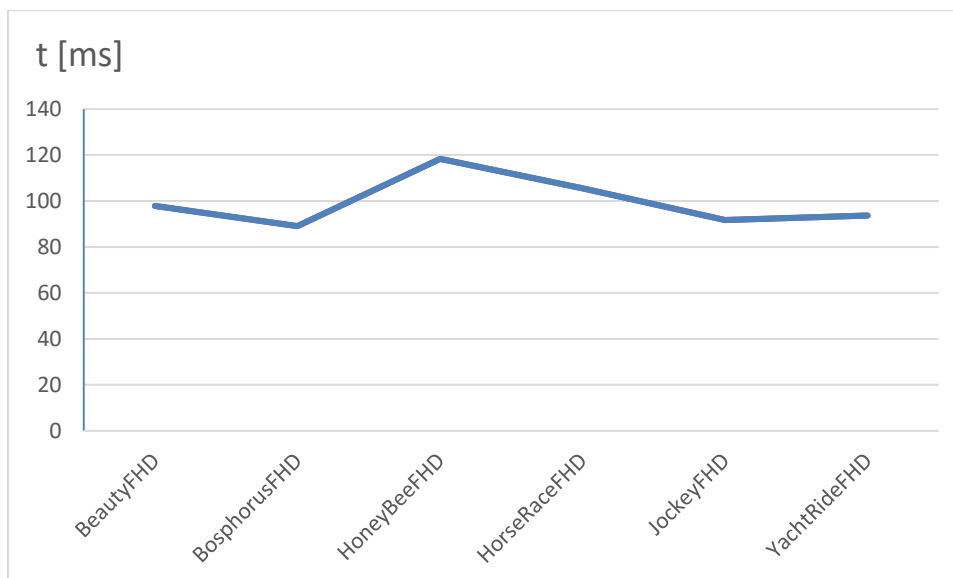
Budući da se algoritam kompresije izvršavao na 6 različitih video sadržaja, potrebno je uvesti mjeru kojom je moguće kvantitativno izraziti različitost sadržaja kako bi mogli opravdati dobiveni postotak kompresije. Uzimajući u obzir logiku rada algoritma, količinu podataka koja se obrađuje te red veličine dobivenih rezultata gotovo je nemoguće pojavljivanje slučaja da različiti video sadržaji imaju identičan broj jedinstvenih unosa u kodnu tablicu. Pod logiku rada uzima se način na koji algoritam popunjava kodnu tablicu uz to da sadržaji nad kojima se izvršava algoritam imaju jednak broj okvira zbog postojanosti rezultata. Broj jedinstvenih unosa biti će različit ako se sekvence razlikuju u samo jednom bitu podataka. Upravo iz toga razloga kao mjera različitosti sadržaja odabrana je veličina koja govori koliko postoji jedinstvenih unosa u kodnoj tablici za svaku pojedinu sekvencu. Kako bi potvrdili odnos koji postoji između broja jedinstvenih vrijednosti u kodnoj tablici i postotka kompresije zabilježen je i broj koji govori koliko puta se kodna tablica briše i inicijalizira na početne vrijednosti. Budući da se kodna tablica briše svaki puta kada se napuni s jedinstvenim vrijednostima, vrijedi isti odnos između postotka kompresije

kao što vrijedi i između broja jedinstvenih unosa. Broj okvira provjeren je prije i poslije slanja kako bi se utvrdilo da nije došlo do njihovog gubitka. Dobiveni rezultati podijeljeni su na dvije grupe ovisno o tome da li je sadržaj nad kojim se LZW kompresija izvršavala bio prethodno komprimiran ili je bio u neobrađenom formatu bez kompresije.

Bitno je naglasiti da su pojedine sekvence bile prethodno komprimirane na način da su im se vrijednosti elemenata slike svodile na određenu vrijednost što dovodi do stvaranja blokova elemenata slike s istim vrijednostima. Za očekivati je da primjena LZW kompresije na ovakve sekvence rezultira s većim postotkom kompresije nego na sekvence u neobrađenom formatu.

5.3.1 Rezultati LZW kompresije na prethodno komprimiranim FHD video sekvencama

Prema grafu na slici 5.1. prosječno vrijeme kompresije prethodno komprimiranih FHD okvira varira od 89ms do 118ms ovisno o kojem sadržaju se radi. Video sekvenca „*BosphorusFHD*“ sastoji se od vrlo sličnih krominantnih i luminantnih podataka pojedinog elementa slike dok je kod „*HoneyBeeFHD*“ sekvence situacija potpuno obrnuta, boje i intenzitet svjetline pojedinih elemenata slike veoma su različiti što utječe na brzinu izvođenja samog LZW kompresijskog algoritma. Prema dobivenim rezultatima nije moguće zadovoljiti zahtjeve realnog vremena koristeći LZW kompresijski algoritam. U najboljem slučaju moguće je komprimirati 11 okvira po sekundi vremena uzimajući u obzir dobivene rezultate vezane za vrijeme izvršavanja kompresijskog algoritma. S druge strane, uzimajući u obzir dobivene postotke kompresije pojedinih prethodno komprimiranih FHD okvira, čak i u najgorem dobivenom slučaju pri samo 37.2% kompresije video sadržaja moguće je zadovoljiti postavljeni zahtjev od 50 FPS-a koristeći prijenosnu tehnologiju ograničene pojasne širine od 1 Gbit/s.



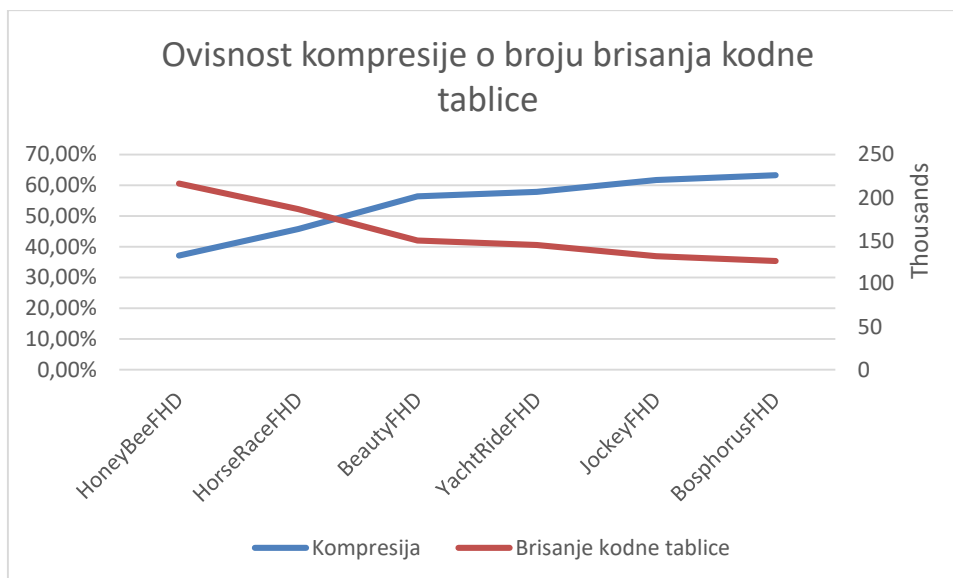
Sl. 5.1. Prosječno vrijeme potrebno za izvođenje LZW algoritma nad pojedinim FHD okvirom.

U tablici 5.3. nalaze se dobiveni rezultati vezani za prosječan postotak kompresije FHD okvira u pojedinoj sekvenci, broj jedinstvenih unosa u kodnu tablicu te broj brisanja kodne tablice.

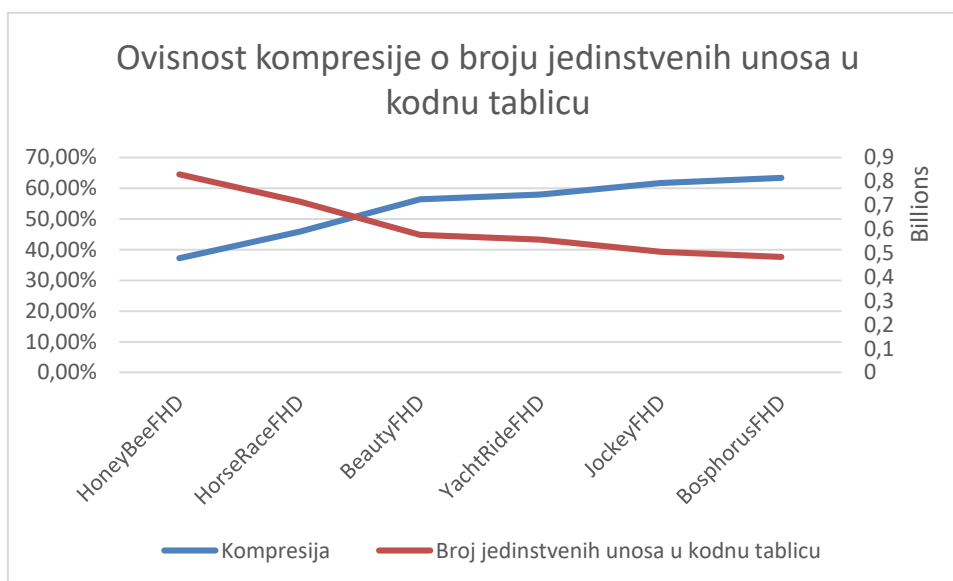
Tab. 5.3. Rezultati LZW algoritma nad prethodno komprimiranim FHD sekvencama.

Ime sekvence	Prosječna kompresija [%]	Broj brisanja kodne tablice	Broj jedinstvenih unosa u kodnu tablicu
HoneyBeeFHDLossy	37,2	216378	829186017
HorseRaceFHDLossy	45,81	186755	715529460
BeautyFHDLossy	56,4	150291	575632214
YachtRideFHDLossy	57,9	145146	555842690
JockeyFHDLossy	61,73	131973	505356346
BosphorusFHDLossy	63,32	126495	484258874

Prema tablici 5.3. te grafovima na slikama 5.2. i 5.3. dolazi se do zaključka da sekvence koje imaju više jedinstvenih unosa u kodnu tablicu imaju manji postotak kompresije sadržaja dok sekvence s manjim brojem unosa imaju veći postotak kompresije kako se i očekuje primjenom LZW algoritma.



Sl. 5.2. Prikaz obrnuto proporcionalnog odnosa između postotka kompresije prethodno komprimiranih FHD sekvenci i broja brisanja kodne tablice.

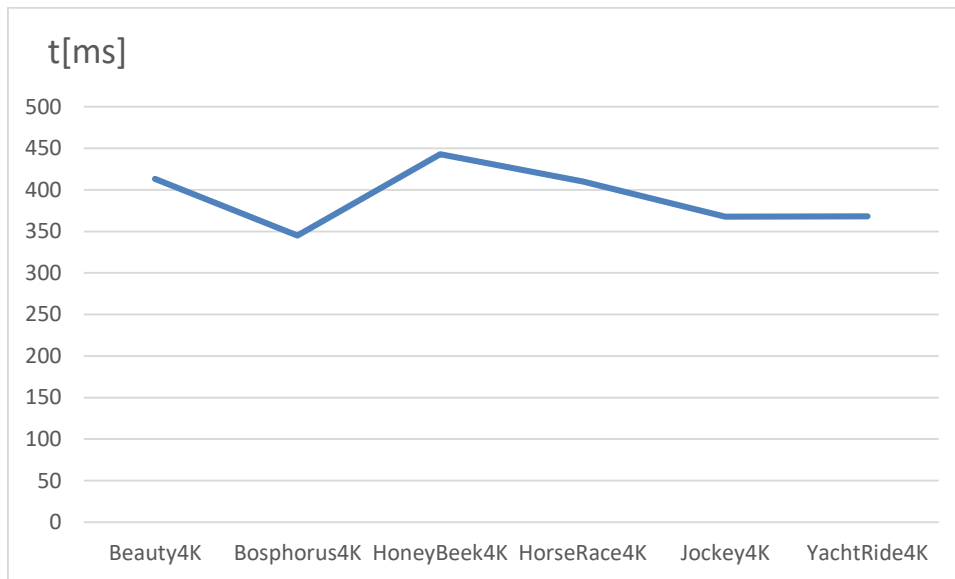


Sl. 5.3. Prikaz obrnuto proporcionalnog odnosa između postotka kompresije prethodno komprimiranih FHD sekvenci i broja jedinstvenih unosa u kodnu tablicu.

Drugim riječima, kod video sadržaja u kojemu je sličnost podataka pojedinog okvira veća, dolaziti će do većeg broja poklapanja s podacima koji se već nalaze u kodnoj tablici, što će na kraju rezultirati manjim brojem jedinstvenih unosa i većim postotkom kompresije. S druge pak strane, kompresija video sadržaja kod kojih se podaci pojedinih okvira razlikuju u velikoj mjeri rezultirati će manjim postotkom kompresije, većim brojem unosa jedinstvenih vrijednosti te većim brojem brisanja kodne tablice.

5.3.2 Rezultati LZW kompresije na prethodno komprimiranim 4K UHD video sekvencama

Budući da LZW kompresija radi na razini bit-a, uzimajući u obzir količinu podataka 4K UHD okvira te količinu podataka FHD okvira, za očekivati je da će se vrijeme kompresije u najboljem slučaju učetverostručiti. Uzmemo li u obzir još vrijeme potrebno da se svaki dodatni bit podataka zapiše u memoriju te vrijeme potrebno za zapisivanje i pretraživanje podataka u kodnoj tablici dolazi se do rezultata prikazanih grafom na slici 5.4.



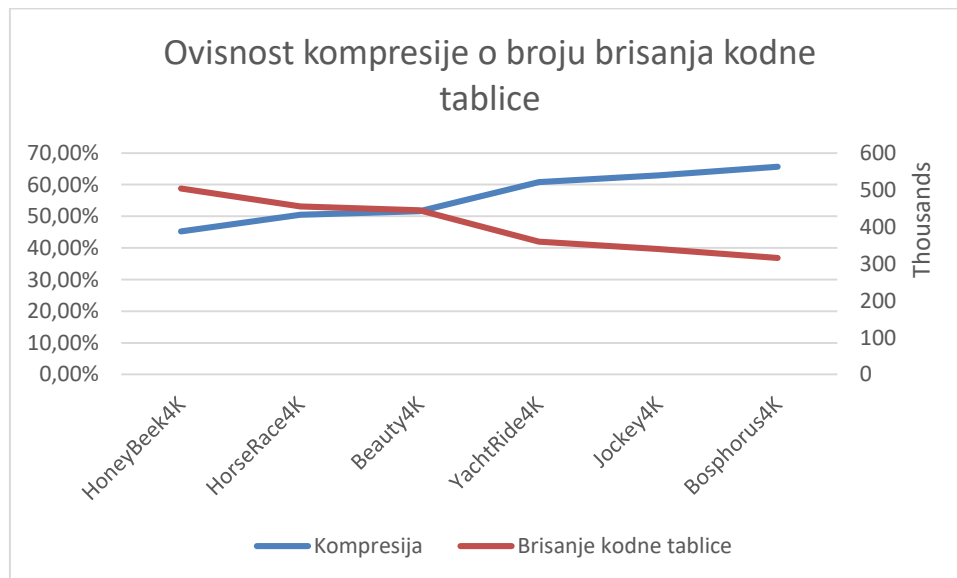
Sl. 5.4. Prosječno vrijeme potrebno za izvođenje LZW algoritma nad pojedinim 4K UHD okvirom.

Za 4K UHD okvire vrijedi isti proporcionalni odnos kao kod FHD okvira između postotka kompresije sadržaja, broja brisanja kodne tablice (Sl.5.5.) i broja jedinstvenih unosa u kodnu tablicu (Sl.5.6.). Kao što je i za očekivati prema dobivenim rezultatima za prethodno komprimirane FHD sekvence, najbrže vrijeme kompresije prethodno komprimiranih 4K UHD sekvenci ostvareno je nad sekvencom „*Bosphorus4K*“, a najsporije nad „*HoneyBee4K*“ budući da sadržaj sekvence, a samim time i sličnost krominantnih i luminantnih podataka, ostaje ista.

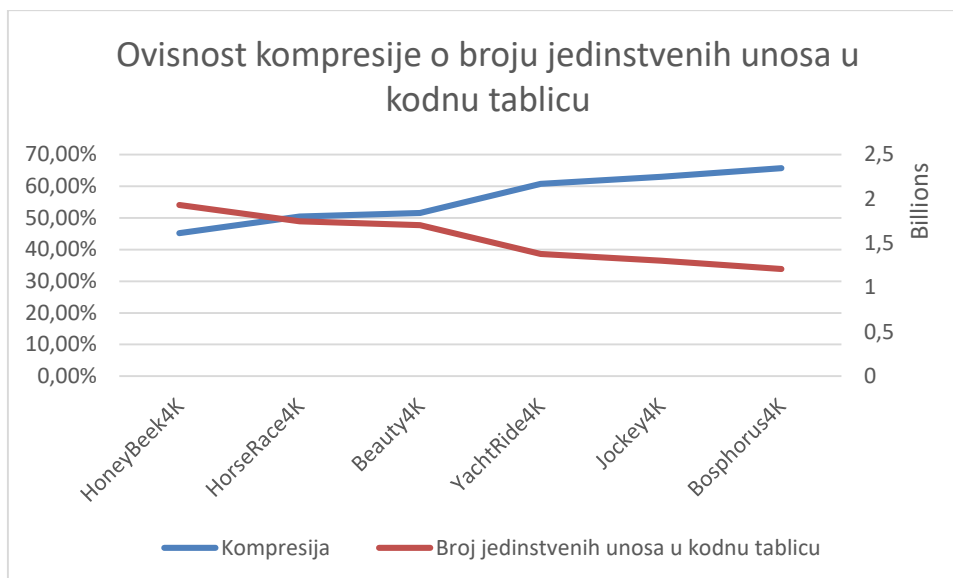
Tab. 5.4. Rezultati LZW algoritma nad prethodno komprimiranim 4K UHD sekvencama.

Ime sekvence	Prosječna kompresija [%]	Broj brisanja kodne tablice	Broj jedinstvenih unosa u kodnu tablicu
HoneyBeek4KLossy	45,21	503363	1930740899
HorseRace4KLossy	50,44	455334	1746435457
Beauty4KLossy	51,61	444546	1705038317
YachtRide4KLossy	60,82	359969	1380537597
Jockey4KLossy	62,97	340218	1304691280
Bosphorus4KLossy	65,69	315218	1208801471

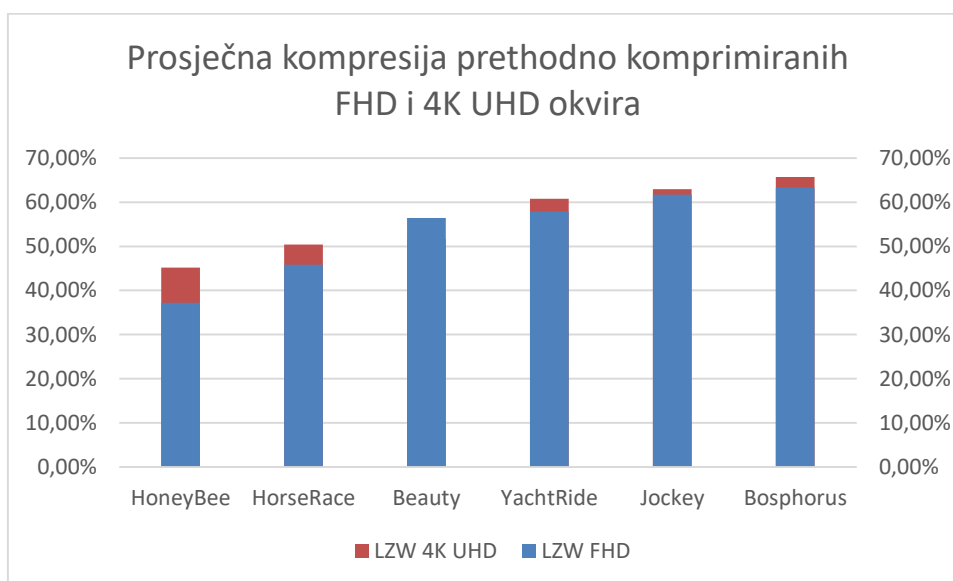
Isto tako za očekivati je da će se broj brisanja kodne tablice i broj jedinstvenih unosa u kodnu tablicu povećati ako povećamo i količinu podataka nad kojima se LZW algoritam izvršava što se može i vidjeti u tablici 5.4.



Sl. 5.5. Prikaz obrnuto proporcionalnog odnosa između postotka kompresije prethodno komprimiranih 4K UHD sekvenci i broja brisanja tablice.



Sl. 5.6. Prikaz obrnuto proporcionalnog odnosa između postotka kompresije prethodno komprimiranih 4K UHD sekvenci i broja jedinstvenih unosa u kodnu tablicu.

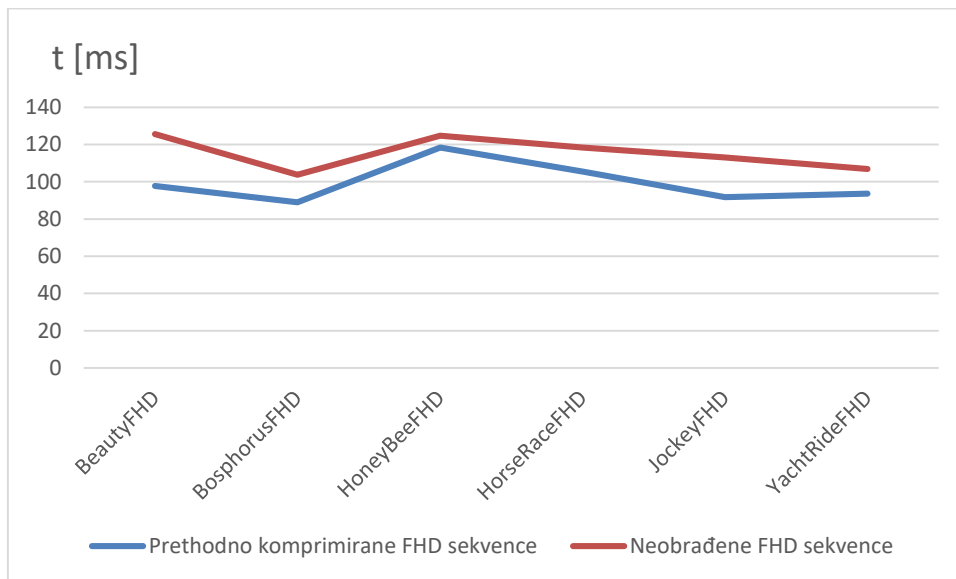


Sl. 5.7. Odnos između prosječne kompresije prethodno komprimiranih FHD i 4K UHD okvira.

Razlike u postocima kompresije prethodno komprimiranih FHD i 4K UHD veoma su male. U najlošijem slučaju iznose 8.01%. Ne postoji univerzalni algoritam za kompresiju video podataka koji bi konzistentno davao približno iste rezultate neovisno o sadržaju. Postotak kompresije najviše ovisi o sadržaju koji se komprimira te o razlici između elemenata slike. Ako je svaki element slike različit, postotak LZW kompresije biti će iznimno nizak te se iz toga razloga ne mogu sigurno garantirati konzistentni rezultati iako se radi o približno istom sadržaju (Sl.5.7.).

5.3.3 Rezultati LZW kompresije na neobrađenim FHD sekvencama

Prema dobivenim rezultatima za prosječno vrijeme kompresije neobrađenih i komprimiranih FHD okvira na slici 5.8. vidljivo je maksimalno prosječno odstupanje od 16ms.



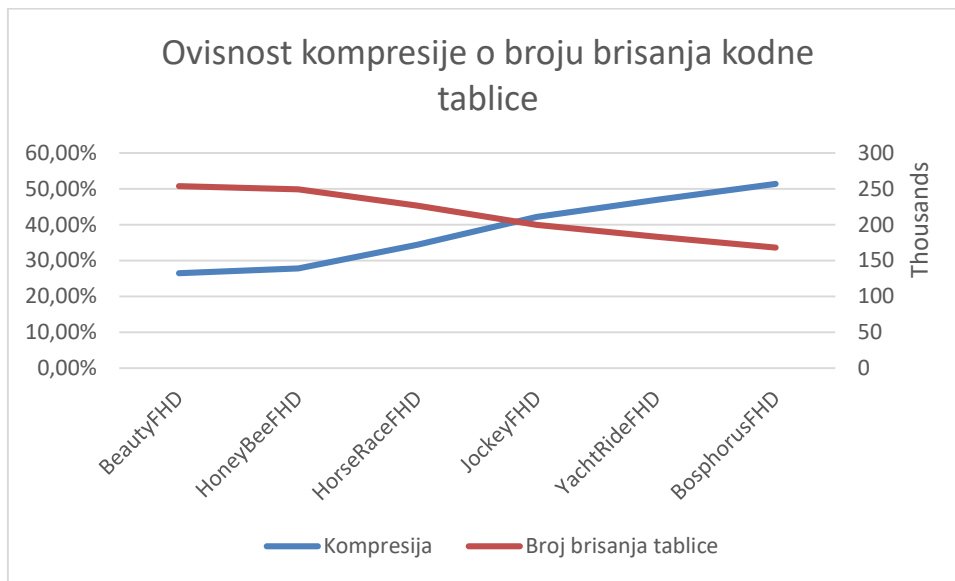
Sl. 5.8. Prosječno vrijeme potrebno za izvođenje LZW algoritma nad prethodno komprimiranim i neobrađenim FHD sekvencama.

Navedeno odstupanje nastalo je iz razloga što prethodno komprimirane sekvence imaju više sličnih podataka budući da su komprimirane koristeći algoritme s gubitkom podataka te je za očekivati da će zahtijevati manji broj unosa i manji broj brisanja kodne tablice što na kraju rezultira većim postotkom kompresije. S druge strane, neobrađene sekvence prema tablici 5.5. imaju gotovo dvostruko veći broj unosa u kodnu tablicu, a samim time potrebno je i više vremena da se izvrše pojedine operacije.

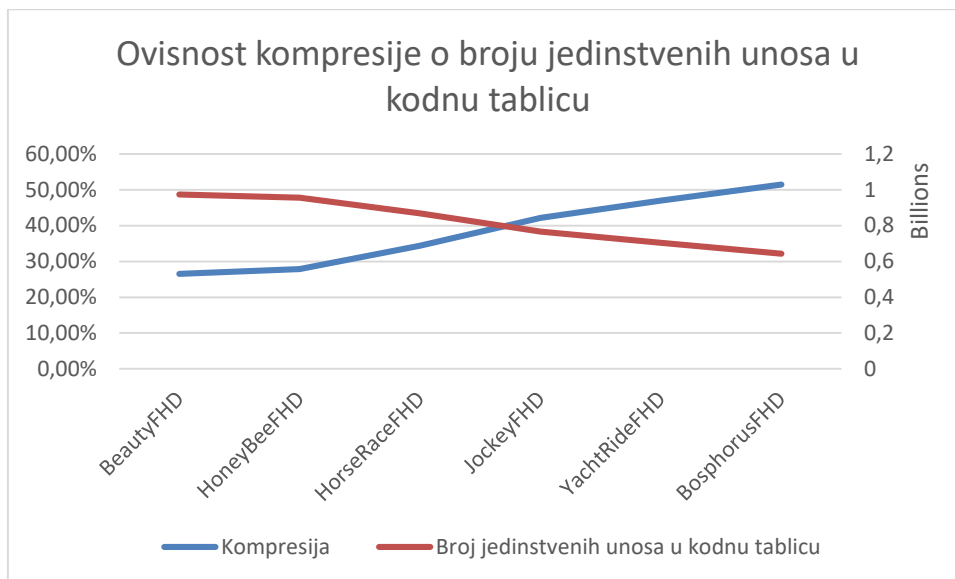
Tab. 5.5. Rezultati LZW algoritma nad neobrađenim FHD sekvencama.

Ime sekvence	Prosječna kompresija [%]	Broj brisanja kodne tablice	Broj jedinstvenih unosa u kodnu tablicu
BeautyFHDLossless	26,53	253940	973230069
HoneyBeeFHDLossless	27,86	249458	955672878
HorseRaceFHDLossless	34,41	226747	868920391
JockeyFHDLossless	42,18	199912	766002395
YachtRideFHDLossless	46,94	183480	702920797
BosphorusFHDLossless	51,42	168004	643527054

Prethodno navedeni obrnuto proporcionalni odnosi između postotka kompresije i broja brisanja tablice (Sl.5.9.) te broja jedinstvenih unosa (Sl.5.10.) vrijede i za neobrađene FHD sekvence.



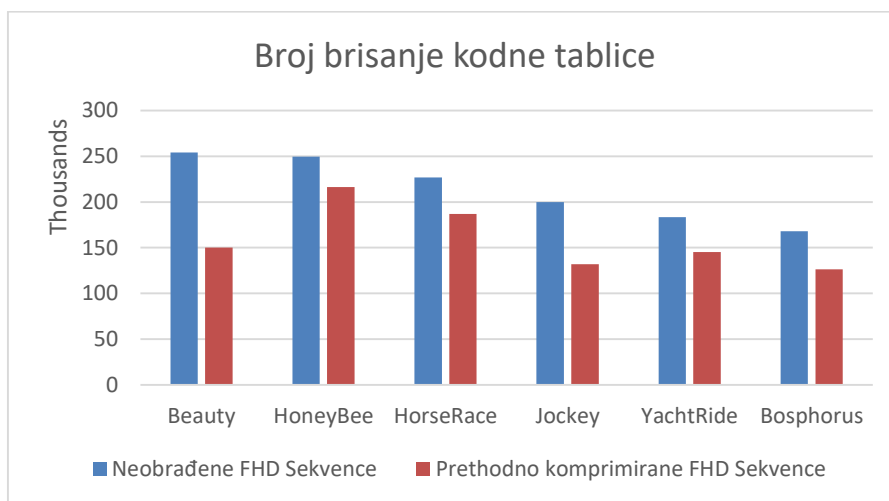
Sl. 5.9. Prikaz obrnuto proporcionalnog odnosa između postotka kompresije neobrađenih FHD sekvenci i broja brisanja tablice.



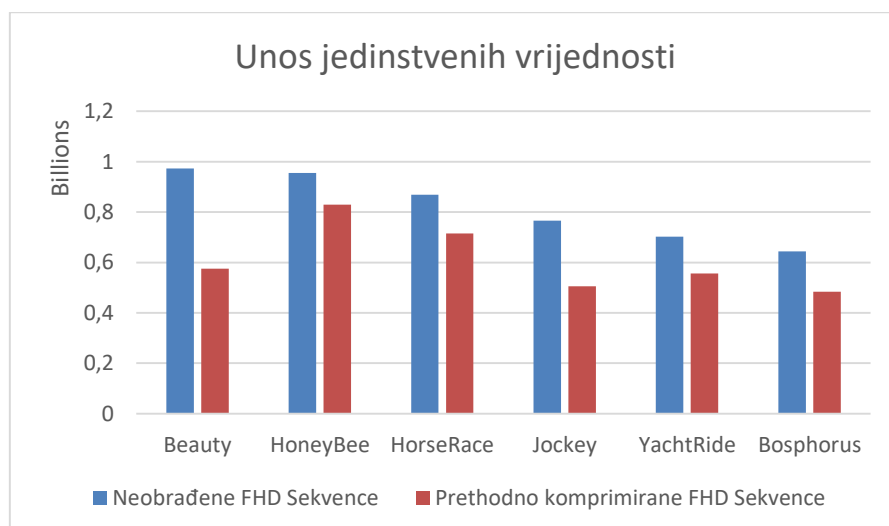
Sl. 5.10. Prikaz obrnuto proporcionalnog odnosa između postotka kompresije neobrađenih FHD sekvenci i broja jedinstvenih unosa u kodnu tablicu.

Manja sličnost podataka neobrađenih sekvenci naspram sličnosti podataka prethodno komprimiranih sekvenci očituje se u dobivenim rezultatima (Tab.5.5.). Tako se prosječan broj brisanja tablice i broj jedinstvenih unosa u kodnu tablicu neobrađenih FHD okvira povećao za

34% (Sl.5.11, Sl.5.12.) naspram rezultata vezanih za prethodno komprimirane FHD sekvence (Tab.5.3.).



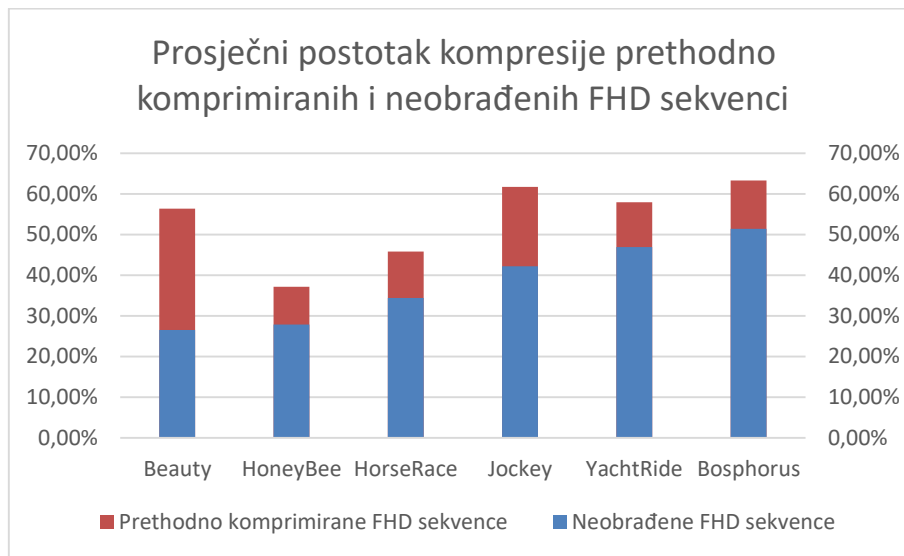
Sl. 5.11. Usporedba brisanja kodne tablice prethodno komprimiranih i neobrađenih FHD sekvenci.



Sl. 5.12. Usporedba broja unosa jedinstvenih vrijednosti u kodnu tablicu prethodno komprimiranih i neobrađenih FHD sekvenci.

Postotak kompresije kod prethodno komprimiranih sekvenci veći je od postotka kompresije neobrađenih FHD sekvenci iz razloga što je prvi dio sekvenci komprimiran algoritmom s gubitkom podataka dok su neobrađene sekvence komprimirane LZW kompresijskim algoritmom bez gubitaka podataka. Uzimajući prethodno navedeno u obzir, prosječni postotak kompresije neobrađenih FHD sekvenci u prosjeku je manji za 29% (Sl.5.13.). Najveću razliku između postotka kompresije neobrađene i prethodno komprimirane sekvence ima sekvenca „Beauty“. Kod „Beauty“ sekvence koja je prethodno komprimirana algoritmom s gubitkom podataka dolazi

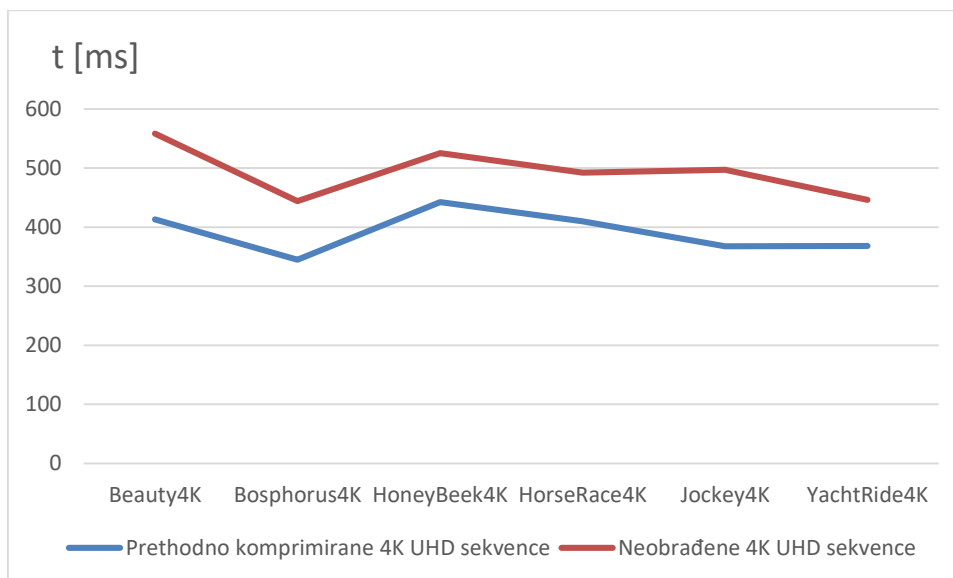
do stvaranja jednoličnih blokova iz razloga što je veliki dio površine pojedinog okvira sekvence prekriven crnom bojom, a samim stvaranjem jednoličnih blokova s istim vrijednostima crne boje dolazi i do većeg postotka kompresije. S druge strane, kod neobrađene „Beauty“ sekvence koja je komprimirana algoritmom bez gubitaka podataka ne postoje jednolični blokovi koji imaju istu vrijednost crne boje za svaki element slike. Uzmemo li u obzir količinu crne boje pojedinog okvira te razliku u crnoj boji elemenata slike koja postoji na razini bit-a, za očekivati je da će postotak kompresije biti drastično niži.



Sl. 5.13. Usporedba prosječnog postotka kompresije prethodno komprimiranih i neobrađenih FHD sekvenci.

5.3.4 Rezultati LZW kompresije na neobrađenim 4K UHD sekvencama

Kao što je bio slučaj odstupanja vremena potrebnog za kompresiju neobrađenih FHD sekvenci u odnosu na prethodno komprimirane sekvence, tako dolazi do odstupanju i u slučaju 4K UHD neobrađenih sekvenci naspram prethodno komprimiranih i to u prosjeku od 102ms. Prema slici 5.14. prosječno vrijeme potrebno za kompresiju jednog 4K UHD okvira iznosi 494ms te se zahtjevi prijenosa u realnom vremenu od 50 FPS-a nikako ne mogu zadovoljiti.



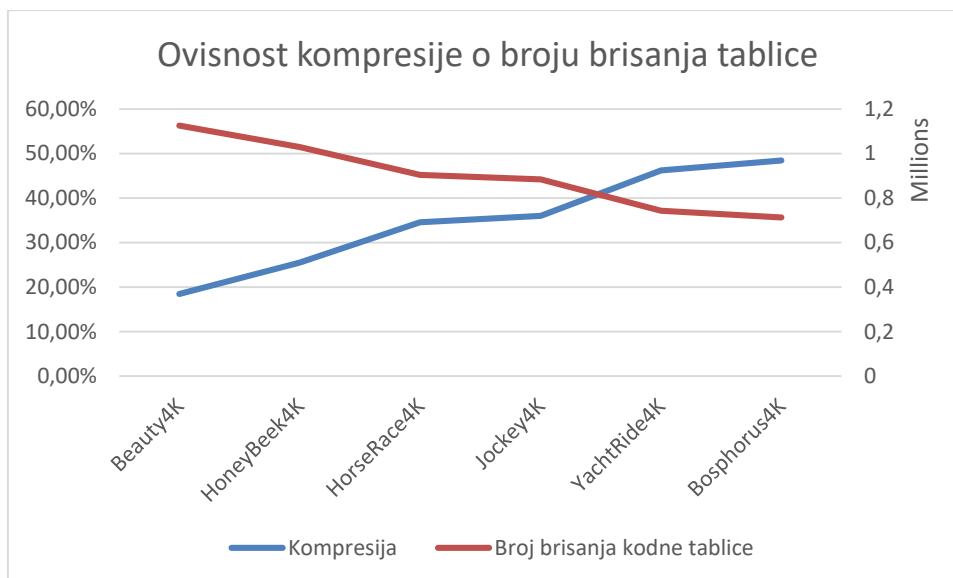
Sl. 5.14. *Prosječno vrijeme potrebno za izvođenje LZW algoritma nad prethodno komprimiranim i neobrađenim 4K UHD okvirima.*

Prema tablicama 5.6. i 5.4. broj brisanja tablice i broj jedinstvenih unosa u kodnu tablicu povisio se u prosjeka za 123% što objašnjava manji postotak kompresije. Kao što je gore navedeno, u obzir treba uzeti da je LZW kompresija bez gubitaka podataka te da su sekvence s kojima se dobiveni rezultati uspoređuju prethodno komprimirane.

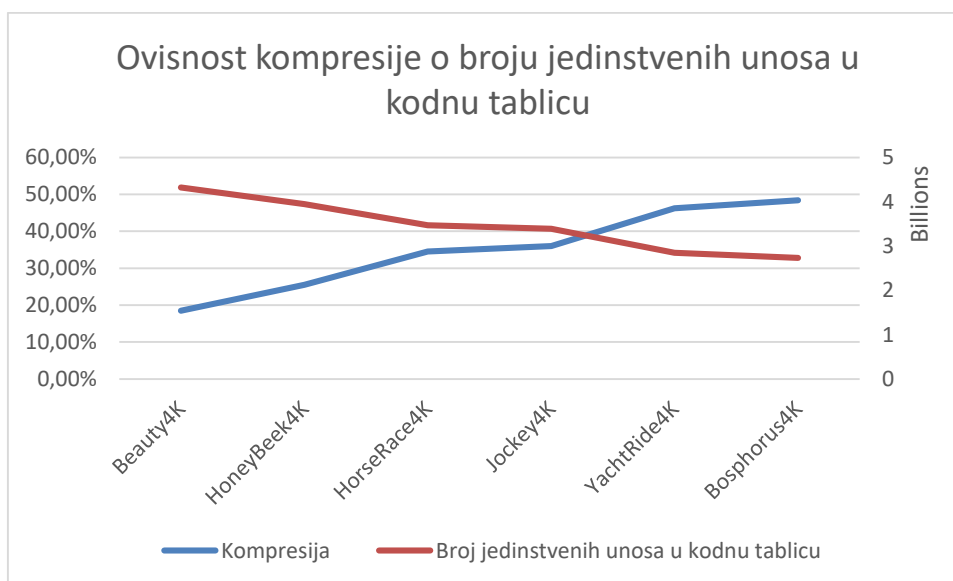
Tab. 5.6. *Rezultati LZW algoritma nad neobrađenim 4K UHD sekvencama.*

Ime sekvence	Prosječna kompresija [%]	Broj brisanja kodne tablice	Broj jedinstvenih unosa u kodnu tablicu
Beauty4KLossless	18,46	1126360	4320811442
HoneyBeek4KLossless	25,50	1028995	3947386570
HorseRace4KLossless	34,57	903842	3466965809
Jockey4KLossless	36,00	884110	3391291803
YachtRide4KLossless	46,24	742677	2848643036
Bosphorus4KLossless	48,43	712418	2732487482

Obrnuto proporcionalni odnos između postotka kompresije i broja brisanja tablice te broja jedinstvenih unosa u tablici prikazan je na slikama 5.15. i 5.16..

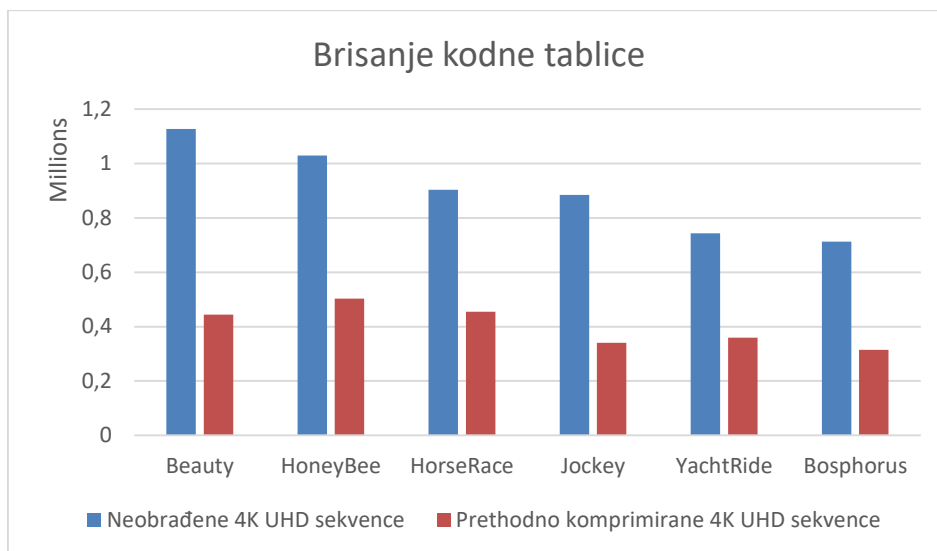


Sl. 5.15. Prikaz obrnuto proporcionalnog odnosa između postotka kompresije neobrađenih 4K UHD sekvenci i broja brisanja tablice.

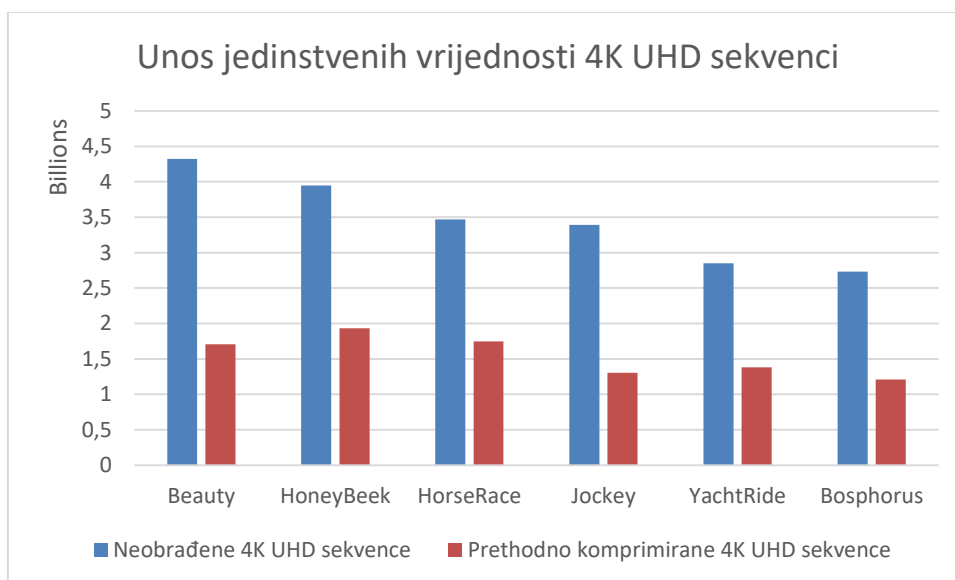


Sl. 5.16. Prikaz obrnuto proporcionalnog odnosa između postotka kompresije neobrađenih 4K UHD sekvenci i broja jedinstvenih unosa u kodnu tablicu.

Kao što je vidljivo na slikama 5.17. i 5.18., kada su u pitanju neobrađene 4K UHD sekvence razlika je puno veća između sličnosti njihovih podataka i podataka prethodno komprimiranih sekvence nego kod FHD rezolucije, pretežito zbog znatno veće količine informacija koja se obrađuju.

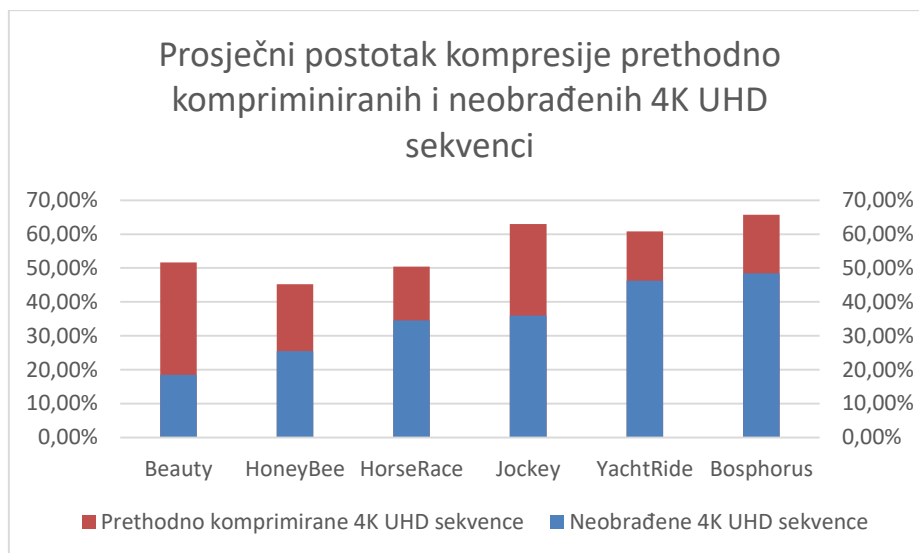


Sl. 5.17 Usporedba brisanja kodne tablice prethodno komprimiranih i neobrađenih 4K UHD sekvenci.



Sl. 5.18. Usporedba broja unosa jedinstvenih vrijednosti u kodnu tablicu prethodno komprimiranih i neobrađenih 4K UHD sekvenci.

Prosječni postotak kompresije prethodno komprimiranih 4K UHD sekvenci veći je za 60% od prosječnog postotka kompresije neobrađenih 4K UHD sekvenci. Sekvence s manjim postotkom kompresije poput „Beauty“ sadrže ogromnu količinu veoma različitih podataka koje ljudsko oko percipira kao jednu boju te ih nije u stanju primijetiti.



Sl. 5.19. Usporedba prosječnog postotka kompresije prethodno komprimiranih i neobrađenih 4K UHD sekvenci.

5.4 Analiza rezultata testiranja programskog rješenja

Kako bi se zadovoljili unaprijed definirani zahtjevi ovoga rada za prijenosom 4K UHD video sadržaja u realnome vremenu pri 50 FPS-a prijenosnim kanalom s ograničenom pojasnom širinom od 1 Gbit/s, potrebno je komprimirati početnu količinu video podataka pojedinog YUV420 4K UHD digitalnog okvira u za minimalno 83% (Tab.5.7., 3840x2160 pri 12 bit-a). S druge pak strane, vrijeme potrebno za cjelokupno procesuiranje pojedinog okvira na poslužiteljskoj strani u najgorem slučaju ne smije prelaziti 20ms. U procesuiranje pojedinog digitalnog okvira ulazi njegovo dekodiranje, komprimiranje te slanje klijentskoj strani.

Tab. 5.7. Potrebni postotci kompresije FHD i 4K UHD okvira pri različitim dubinama boja.

Visina [px]	Širina [px]	Dubina boje [bit]	Količina podataka [MB]	Potreban postotak kompresije [%]
3840	2160	12	11,87	83,14
3840	2160	16	15,82	87,36
3840	2160	24	23,73	91,57
1920	1080	12	2,97	32,58
1920	1080	16	3,96	49,43
1920	1080	24	5,93	66,29

Ako je prosjek postotka kompresije pojedinog okvira manji od 83% ili prosječno vrijeme cjelokupne obrade pojedinog okvira prelazi 20ms, smanjuje se broj digitalnih okvira koje je potencijalno moguće prenijeti klijentskoj aplikaciji, a samim time narušava se postavljeni zahtjev

prijenosa 4K UHD video sadržaja pri 50 FPS-a. U najboljem slučaju, prema dobivenim vremenskim rezultatima moguće je maksimalno prenositi 2.89 digitalna okvira po sekundi vremena. S druge pak strane, u slučaju da se zadovolji vremenski zahtjev obrade od 20ms, prijenos u realnome vremenu bio bi ograničen komprimiranom veličinom pojedinog okvira koja u najboljem slučaju iznosi 48.43%. Navedene probleme moguće je riješiti sklopovskim ili programskim ubrzanjem rada samog algoritma te korištenjem prijenosne tehnologije koja omogućuje znatno brži prijenos od 1Gbit/s.

6. ZAKLJUČAK

Glavni cilj ovog rada bio je pokazati je li moguće prenositi video sadržaj 4K UHD rezolucije u realnome vremenu pri unaprijed ograničenim propusnim opsegom prijenosnog kanala koristeći *Lempel-Ziv-Welch* kompresijsku metodu bez gubitaka podataka.

Kako je iz dobivenih rezultata vidljivo, nije moguće osigurati dostatan postotak kompresije video sadržaja visoke rezolucije koristeći LZW kompresijsku metodu. Najbolji postignuti prosječni postotak kompresije bez gubitaka pojedinog 4K UHD digitalnog okvira u neobrađenom formatu iznosi 48,43% pri 345ms potrebnih za cjelokupnu obradu te ni u kojem slučaju ne zadovoljava postavljene zahtjeve ovoga rada. S druge strane, najbolji postignuti prosječni postotak kompresije bez gubitaka pojedinog FHD digitalnog okvira u neobrađenom formatu iznosi 51,42% pri 103ms potrebnih za cjelokupnu obradu. Ako bi se vrijeme potrebno za cjelokupnu obradu FHD digitalnog okvira u neobrađenom formatu ubrzalo na 20ms svi postavljeni uvjeti bili bi zadovoljeni. Isto tako, ako se ubrza cjelokupno vrijeme obrade prethodno komprimiranog FHD digitalnog okvira s 89ms na 20ms, postavljeni uvjeti bili bi zadovoljeni čak i u najgorem dobivenom slučaju pri samo 37,2% kompresije sadržaja.

Budući da vrijeme izvršavanja LZW kompresijske metode ovisi o sklopovlju na kojemu se algoritam izvršava, ne može se sigurno tvrditi da vremenske zahtjeve nije moguće zadovoljiti. Kako bi se zadovoljili vremenski zahtjevi potrebno je korišteno sklopovlje zamijeniti sa sklopovljem veće procesorske moći kako bi se kompresijske funkcije brže izvršavale. Kao drugo rješenje problema prijenosa video sadržaja visoke rezolucije u realnome vremenu predlaže se uvođenje prijenosne tehnologije novije generacije s većom propusnosti prijenosnog kanala.

LITERATURA

- [1] International Telecommunication Union, Recommendation ITU-R BT.2020-2, 10/2015.
- [2] https://en.wikipedia.org/wiki/Ultra-high-definition_television#/media/File:8K_UHD,_4K_SHD,_FHD_and_SD.svg, 9/2016.
- [3] <http://www.listeningpost.co.nz/2K-4K-Smart-I.84011-N.81089>, 9/2016.
- [4] P. Read, M.P. Meyer, Restoration of Motion Picture Film, Butterworth-Heinemann, Oxford, 8/2000.
- [5] <http://www.gamerheadlines.com/2014/12/video-graphics-facts>, 9/2016.
- [6] W.D. Wright, A re-determination of the trichromatic coefficients of the spectral colors, Transactions of the Optical Society, No. 4, Vol. 30, od 141- do 164, 2/1929.
- [7] https://upload.wikimedia.org/wikipedia/commons/thumb/2/27/CIExy1931_Rec_2020_and_Rec_709.svg/476px-CIExy1931_Rec_2020_and_Rec_709.svg.png, 9/2016.
- [8] <https://en.wikipedia.org/wiki/YUV>, 9/2016.
- [9] <http://broadcastidea.com/index.php/2016/01/23/chroma-sub-sampling-ratio-jab>, 9/2016.
- [10] <http://www.peachpit.com/articles/article.aspx?p=433754&seqNum=2>, 9/2016.
- [11] http://tovid.wikia.com/wiki/Aspect_ratio_and_safe_area, 9/2016.
- [12] https://en.wikipedia.org/wiki/16:9#/media/File:Dr._Kerns_Powers,_SMPTE_derivation_of_16-9_aspect_ratio.svg, 9/2016.
- [13] <http://www.irinfo.org/04-01-2006-colbert>, 9/2016.
- [14] FFmpeg izvorni kod, (<https://github.com/FFmpeg/FFmpeg>), 9/2016.
- [15] Testne sekvence, Ultra Video Group, Tampere University of Technology (<http://ultravideo.cs.tut.fi/#testsequences>), 9/2016.
- [16] A. Moskvina, O. Petrov, S. Putilin, S. Grishin, A. Marat, G. Osipov, MSU Graphics & Media Lab (Video Group), MSU Video Quality Measurement Tool, (http://compression.ru/video/quality_measure/video_measurement_tool_en.html#nav), 9/2016.

SAŽETAK

Glavni problem koji se analizira u ovom diplomskom radu je mogućnost prenošenja video sadržaja visoke rezolucije bez gubitaka podataka prijenosnim kanalom s unaprijed ograničenom uskom pojasnom širinom, a da pri tome ostanu zadovoljeni specifični zahtjevi prijenosa video sadržaja u realnome vremenu. Od dva predložena moguća rješenja ovoga problema: sklopovsko povećanje propusnog opsega prijenosnog kanala te programska kompresija podataka bez gubitaka podataka, ispitana je programska varijanta kompresije podataka prije slanja koristeći *Lempel-Ziv-Welch* kompresijski algoritam. Ispitivanje je izvršeno na sveukupno 6 različitih video sadržaja u dvije rezolucije, FHD i 4K UHD. Za svaki od 6 sadržaja preuzete su video sekvence u neobrađenom i prethodno komprimiranom formatu kako bi se mogla ispitati efikasnost kompresije bez gubitaka podataka. Za rukovanje multimedijским sadržajem kao i algoritmima same kompresije iskorišten je otvoreni izvorni kod programskog alata *FFmpeg*. Rezultati dobiveni ispitivanjem pokazuju da nije moguće prenositi video sadržaj visoke rezolucije prijenosnim kanalom ograničene pojasne širine od 1 Gbit/s, a da pri tome ostanu zadovoljeni zahtjevi prijenosa video sadržaja u realnome vremenu.

Ključne riječi: video sadržaj visoke rezolucije, *Lempel-Ziv-Welch* algoritam, kompresija bez gubitaka podataka, prijenos video sadržaja u realnome vremenu

SOFTWARE MODULES FOR 4K GRABBER SUPPORT

ABSTRACT

Main problem that has been solved with this thesis was to find out if it is possible to transfer ultra high resolution video content through predefined transmission channel bandwidth without compromising real time video transmission standards. Out of two proposed solutions regarding above mentioned problem: hardware-wise transmission technology improvement and software implementation of lossless compression method, it was decided to explore possibilities of software implementation of Lempel-Ziv-Welch compression algorithm. Testing was done using 6 different video contents in two different resolutions, FHD and 4K UHD. Regarding LZW compression efficiency testing, for each video content sequences were acquired in raw and compressed format so comparison could be done. FFmpeg's open source code was used for handling multimedia as well as LZW compression routines. Acquired testing results prove that transmission of high resolution video content is not possible through restricted transmission channel bandwidth of 1Gbit/s without compromising real time video transmission standards.

Key words: ultra high resolution video content, Lempel-Ziv-Welch algorithm, lossless compression, real time video transmission

ŽIVOTOPIS

Luka Petrinšak rođen je 16. svibnja 1992. godine u Osijeku. Nakon završene osnovne škole „Jagoda Truhelka“, 2007. godine upisuje Prirodoslovno-matematičku gimnaziju u Osijeku te ju 2011. godine završava s vrlo dobrim uspjehom. Iste godine upisuje preddiplomski studij, smjer Računarstvo na Elektrotehničkom fakultetu Osijek koji polaže u roku s vrlo dobrim uspjehom. Nakon položenog preddiplomskog studija, godine 2014. na istome fakultetu upisuje diplomski sveučilišni studij Računarstva, smjer Procesno računarstvo. Godinu dana poslije postaje stipendist Instituta za informacijske tehnologije RT-RK Osijek gdje trenutno radi na problemima iz područja digitalne televizije.

Potpis:
