

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1316

PRILAGODLJIVO STRUJANJE VIDEA OD 360 STUPNJEVA

Tomislav Žulj

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1316

PRILAGODLJIVO STRUJANJE VIDEA OD 360 STUPNJEVA

Tomislav Žulj

Zagreb, lipanj 2020.

DIPLOMSKI ZADATAK br. 1316

Pristupnik: **Tomislav Žulj (0036480805)**
Studij: Informacijska i komunikacijska tehnologija
Profil: Telekomunikacije i informatika
Mentor: izv. prof. dr. sc. Lea Skorin-Kapov

Zadatak: **Prilagodljivo strujanje videa od 360 stupnjeva**

Opis zadatka:

Prikaz videa od 360 stupnjeva putem tehnologije virtualne stvarnosti (engl. Virtual Reality, VR) omogućeno je preslikavanjem sferičnog videa u 3D geometriju. Koristeći VR zaslon na glavi, korisnikova točka gledišta nalazi se u središtu sfere te se može pomicati oko triju osi (3 degrees of freedom, 3DoF). Strujanje 360-videa zahtjeva velike brzine prijenosa, što predstavlja izazov za mrežu. Zbog ograničenog mrežnog kapaciteta, potrebno je istražiti mogućnosti smanjenja i optimiranja brzine prijenosa videa. Određene tehnike baziraju se na praćenju korisnikovog područja gledanja (engl. Field of View, kratica: FoV) te prijenosa videa tako da je dio koji korisnik trenutno promatra prikazan u velikoj rezoluciji dok je dio koji korisnik ne gleda prikazan u manjoj rezoluciji. Vaš zadatak je istražiti i opisati mehanizme strujanja videa od 360 stupnjeva putem komunikacijske mreže. Potrebno je proučiti i opisati normu Omnidirectional Media Format (OMAF), predloženu od strane MPEG-a, koja standardizira format medija za razvoj aplikacija temeljenih na prikazu videa od 360 stupnjeva. Nadalje, potrebno je implementirati i evaluirati laboratorijski prototip kojim se ostvaruje prilagodljivo strujanje videa od 360 stupnjeva. Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije

Rok za predaju rada: 30. lipnja 2020.

Iskreno se zahvaljujem svojoj mentorici izv.prof.dr.sc Lei Skorin-Kapov na mentorstvu i pruženoj pomoći i potpori u izradi ovog rada. Također, zahvaljujem tvrtki Ericsson Nikola Tesla d.d. i Fakultetu Elektrotehnike i Računarstva, Sveučilišta u Zagrebu, a posebno Zavodu za telekomunikacije na pruženoj podršci i svojoj potrebnoj opremi. Za kraj, posebnu zahvalu želim izraziti svojim prijateljima, obitelji i najdražoj Dori na konstantnoj ljubavi, neiscrpnju podršci i neprestanom ohrabrenju. Hvala vam.

Sadržaj

| | |
|--|----|
| 1. Uvod..... | 1 |
| 2. Pregled povezanih radova..... | 5 |
| 3. Strujanje videa od 360 stupnjeva | 9 |
| 3.1. 360-video | 9 |
| 3.1.1. Kreiranje | 10 |
| 3.1.2. Kodiranje..... | 10 |
| 3.1.3. Reprodukcija..... | 13 |
| 3.1.4. Objavljivanje..... | 13 |
| 3.1.5. Varijacije | 14 |
| 3.2. Video kodiranje visoke učinkovitosti..... | 14 |
| 3.3. Prilagodljivo strujanje putem protokola HTTP | 19 |
| 3.4. Prilagodljivo strujanje uz prostornu podjelu na pločice..... | 22 |
| 4. OMAF standard..... | 26 |
| 4.1. Općenito o OMAF-u | 26 |
| 4.2. Formati prikazivanja svesmjernog videa | 27 |
| 4.3. Proširenja za formate za svesmjerni video..... | 29 |
| 4.4. Profili OMAF videa | 32 |
| 5. Implementacija prilagodljivog strujanja uz prostornu podjelu na pločice | 34 |
| 5.1. GPAC..... | 35 |
| 5.2. Nokiotech Omaf | 40 |
| 5.3. Laboratorijski prototip..... | 41 |
| 5.3.1. Arhitektura sustava | 41 |
| 5.3.2. Kodiranje videa | 43 |
| 5.3.3. Prikaz putem zaslona pametnog telefona | 47 |
| 6. Korištene programske tehnologije..... | 48 |

| | | |
|------|---------------------------------|----|
| 6.1. | Microsoft Visual Studio..... | 48 |
| 6.2. | Android Studio | 48 |
| 6.3. | Wireshark..... | 49 |
| 6.4. | FFmpeg..... | 50 |
| 6.5. | Kvazaar..... | 50 |
| 7. | Evaluacija i performanse | 51 |
| 7.1. | Postavljanje eksperimenta | 52 |
| 7.2. | Mjerenja | 53 |
| 7.3. | Rezultati..... | 55 |
| 8. | Zaključak i budući rad..... | 58 |
| | Literatura | 59 |
| | Sažetak | 65 |
| | Summary..... | 66 |
| | Popis slika | 67 |
| | Popis tablica..... | 69 |

1. Uvod

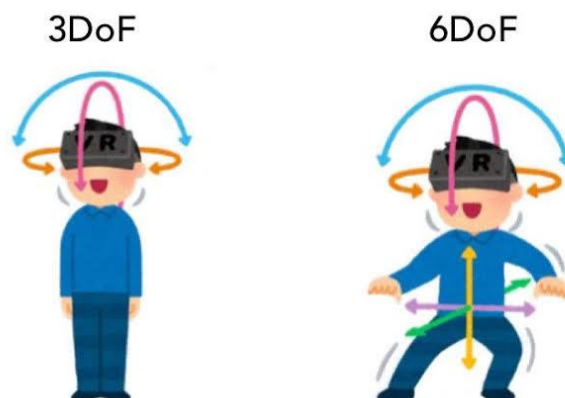
Tijekom posljednjih godina potražnja za online strujanjem video sadržaja od 360 stupnjeva putem pametnih telefona i putem tehnologije virtualne stvarnosti (engl. *Virtual Reality*, kratica: VR) znatno se povećala [1][2]. Jedan od razloga tome su nedavna dostignuća i poboljšanja u hardverskim i softverskim performansama pametnih uređaja i VR zaslona na glavi (engl. *Head-mounted Display*, HMD), koji korisnicima pružaju očaravajuća iskustva, a da im je pritom cijena pristupačnija. Također glavni pružatelji sadržaja prepoznali su vrijednost 360-videa za obogaćivanje iskustvene kvalitete (engl. *Quality of Experience*, kratica: QoE) krajnjeg korisnika Facebookova kupnja Oculus 2014. godine u iznosu od dvije milijarde dolara pokrenula je brzi razvoj VR-a, proizvodnju uređaja i usluga diljem svijeta [1]. Gotovo iznenada, VR je postao važan strateški smjer mnogih tvrtki, a na tržištu su postale dostupne sve vrste VR kamera i uređaja [1]. Prema istraživanju tvrtke *Cisco* očekuje se porast proizvodnje VR zaslona na glavi s 18 milijuna na gotovo 100 milijuna do 2022. godine, dok će se porast pripadajućeg mrežnog prometa povećati skoro 12 puta [2]. Neki od primjera takvih uređaja su HTC Vive¹, Google Cardboard² i Samsung Gear VR³. Navedeno istraživanje također ističe da će aplikacije za strujanje video sadržaja od 360 stupnjeva putem VR-a biti ključne u daljnjem razvoju virtualne stvarnosti i njenom proboju na tržište. YouTube je jedna od glavnih platformi za reprodukciju videa na tržištu, a jedna od njegovih značajki je reprodukcija videa od 360 stupnjeva koji korisniku omogućuju interakciju sa sadržajem i rotaciju pogleda u sceni [3]. YouTube se oslanja na prilagodljivo strujanje putem protokola HTTP i jedan je od najpopularnijih usluga za dostavu medija na Internetu [4][5].

¹ <https://www.vive.com/us/>

² <https://arvr.google.com/cardboard/>

³ <https://www.samsung.com/global/galaxy/gear-vr/>

Video od 360 stupnjeva, poznat i kao **svesmjeran** (engl. *Omnidirectional*) ili sferični video, je videozapis u kojem se pogled u svakom smjeru snima istovremeno pomoću 360-kamere ili kolekcije kamera [3]. Tijekom reprodukcija takvog videa na ravnom zaslonu na pametnom telefonu, korisnik ima kontrolu nad smjerom gledanja kao kod panorame te se može aktivno uključiti u videozapis i može slobodno kontrolirati svoj smjer gledanja prilikom reprodukcije. Kod prikaza putem VR-a omogućeno je mapiranje sferičnog videa u 3D geometriju. I tada, koristeći VR zaslon na glavi, korisnikova točka gledišta nalazi se u središtu sfere te se može pomicati oko triju osi (engl. *3 degrees of freedom*, kratica: 3DoF) što znači da korisnik može gledati lijevo ili desno, zakretati glavu gore ili dolje, te rotirati ju lijevo ili desno. Također postoje i 6DoF VR zasloni na glavi koji osim rotacije i pomicanja glave prate kretanje korisnika u prostoru, a primjer takvog zaslona je spomenuti HTC Vive. **Slika 1.1** prikazuje razliku tih dviju vrsti zaslona na glavi.



Slika 1.1 Razlika između 3DoF-a i 6DoF-a (slika preuzeta s [6])

Pružanje visokokvalitetnog virtualnog iskustva strujanjem 360-videa preko mreže zahtijeva znatno veću širinu prijenosnog pojasa (engl. *bandwidth*) od tradicionalnog 2D videa, što predstavlja izazov u implementaciji takvih aplikacija. Također se očekuje da će mobilni uređaji činiti 71% od ukupnog IP prometa do 2022. godine te će se prijenos sadržaja najviše odvijati putem mobilnih mreža, koje i dalje nemaju dovoljno veliku širinu prijenosnog pojasa za spomenuto strujanje [2]. Nadalje, strujanje 360-videa u VR-u zahtijeva visoke mrežne performanse kako bi zadovoljile iskustvenu kvalitetu korisnika [7]. Kako bi se stekao dojam o ovim izazovima istraživanja su pokazala da za odgovarajuću razinu kvalitete (QoE) aplikacije za strujanje 360-videa u VR-u zahtijevaju mrežno kašnjenje manje od 10 milisekundi, dok će širina prijenosnog pojasa za nadolazeće sadržaje ultra visoke kvalitete dostići 100 Mbps [8]. Implementacija ovakvih aplikacija je izazovna, ne samo za mrežne operatore koji će morati pružati bolje mrežne usluge, već i za programere koji će morati dizajnirati nove, inteligentne tehnike za optimizaciju brzine prijenosa videa kako bi zadovoljili uvjete mreže i iskustvenu kvalitetu korisnika.

Za savladavanje gore spomenutih izazova i ograničenog mrežnog kapaciteta istražuju se razni pristupi i tehnike za učinkovito strujanje video sadržaja prikazanog putem tehnologije virtualne stvarnosti. Glavni problem je taj što korisniku većina isporučenih podataka o video sadržaju nije prikazana jer VR zaslone na glavi prikazuju samo dio od potpunog 360-videa, tj. samo **korisnikovo područje gledanja** (engl. *Field of View*, kratica: FoV) koje je na Samsung Gear VR-u 101 stupanj [9], te strujanje cjelokupnog videozapisa uzima mnogo mrežnog resursa. Stoga se određene tehnike baziraju na praćenju korisnikovog područja gledanja te prijenosa videa tako da je dio koji korisnik trenutno promatra prikazan u velikoj rezoluciji dok je dio koji korisnik ne gleda prikazan u manjoj rezoluciji.

Cilj ovog rada je istražiti i opisati mehanizme strujanja videa od 360 stupnjeva putem komunikacijske mreže. Proučena je i opisana norma Omnidirectional Media Format (OMAF) [10], predložena od strane MPEG-a (engl. *Moving Picture Experts Group*), koja standardizira format medija za razvoj aplikacija temeljenih na prikazu videa od 360 stupnjeva. Nadalje, implementiran je i evaluiran laboratorijski prototip kojim se ostvaruje prilagodljivo strujanje videa od 360 stupnjeva i prikaz putem zaslona pametnog telefona.

Rad je strukturiran kako slijedi. Nakon Uvoda, u drugom poglavlju navedeni su i opisani radovi koji su svojom temom i tehnologijama povezani s ovim radom te koji su pomogli u razumijevanju spomenutih izazova implementacije prototipa za strujanje 360-videa. Potom, treće poglavlje odnosi se na strujanje videozapisa od 360 stupnjeva putem komunikacijske mreže. Dan je opis samog 360-videa, što je on i kako se dobiva te su pobliže objašnjene tehnologije potrebne za prilagodljivo strujanje. Poglavlje broj četiri opisuje standard OMAF i detaljnije razrađuje njegove značajke. Nadalje u poglavlju pet, navedena su i objašnjena rješenja prilagodljivog strujanja s prostornom podjelom na pločice te je opisan laboratorijski prototip kojim se ostvaruje prilagodljivo strujanje videa od 360 stupnjeva i prikaz putem zaslona pametnog telefona. Nadalje, šesto poglavlje nabraja i opisuje sve alate i tehnologije korištene za izradu implementacijskog dijela rada. Sedmo poglavlje opisuje evaluaciju i performanse aplikacije ispitane u laboratorijskom okruženju te su dani rezultati i analize rezultata samih testiranja. Osmo poglavlje je zaključak i budući rad, a nakon toga slijedi popis literature, sažeci rada na hrvatskom i engleskom jeziku. Za kraj je naveden popis slika i tablica koje se nalaze u ovom radu.

2. Pregled povezanih radova

U ovom poglavlju bit će navedeni i prikazani sažeci provedenih studija i povezanih radova, a koriste slične tehnologije koje su objašnjene u ovom radu.

Za 360-video, učinkovito korištenje prijenosnog pojasa pomoću tehnologije prilagodljivog strujanja, ključni je faktor za povećanje korisničke percepcijske kvalitete. To je važan aspekt koji pružatelji usluga i mrežni operatori trebaju razmotriti. Kako bi se optimizirala širina prijenosnog pojasa, istražene su tehnike za strujanje videa u virtualnoj stvarnosti koje se temelje na prostornoj podjeli 360-videa na dijelove, tj. **pločice** (engl. *tiles*) [3][11][12]. Pločice osjetljive na korisničko vidno polje zapravo su susjedni dijelovi cijelog video okvira te mogu biti kodirani i dekodirani nezavisno. Ovaj koncept dio je HEVC (engl. *High Efficiency Video Coding*) standarda i njegovog standardiziranog dekodera i smatra se kao ključna tehnologija za kodiranje videa s više razina kvalitete [13]. Navedena istraživanja proširuju dobro utvrđene pristupe strujanja 2D videa kao što je prilagodljivo strujanje putem protokola HTTP (engl. *HTTP Adaptive Streaming*, kratica: HAS) [14][15]. HAS se usredotočuje na kodiranje izvornog sadržaja na više razina kvalitete (npr. 720p, 1080p, 2K, 4K), tj. razina različitih brzina prijenosa i razlučivosti, dok se svaka razina još dodatno vremenski podijeli na male segmente koji obično traju između 2 i 10 sekundi [14][15]. Novija istraživanja išla su korak dalje i predložene su novije varijante HAS-a te se pokazalo da se širina prijenosnog pojasa također može optimizirati tako da se uz vremensku podjelu doda i prostorna koja je ranije spomenuta [16].

Radovi kao [13][17][18] uvode koncept **prilagodljivog strujanja sadržaja uz prostornu podjelu na pločice** (engl. *tile-based streaming*) u virtualnoj stvarnosti. U tim radovima se prilikom strujanja 360-videa korisniku u visokoj kvaliteti prikazuju dijelovi koji su mu u vidnom okviru (engl. *viewport*), dok su ostali koji su izvan korisničkog vidnog okvira u nižoj kvaliteti ili se opće ne prikazuju [18]. **Slika 2.1** prikazuje koncept tehnologije prilagodljivog strujanja sadržaja uz prostornu podjelu na pločice.



Slika 2.1 Tehnologija strujanja uz prostornu podjelu na pločice (slika preuzeta s [19])

Također, jedan od načina za optimizaciju strujanja videa od 360 stupnjeva je pristup temeljen na projekciji videa. Ovaj pristup, koji je predložio Kuzyakov [20], trenutno se provodi u praktičnim sustavima [21] i koristi dobit od geometrijske projekcije. Budući da se video kodiranje trenutno temelji na dvodimenzionalnim pravokutnicima, bilo koji videozapis od 360 stupnjeva treba prije kodiranja projicirati u 2D video. Uobičajene projekcije su pravokutne, kubne, i piramidalne [3][22]. Nekoliko istraživanja je istodobno proučavalo rješenja za strujanje videa od 360 stupnjeva koja su temeljena na istom principu kao i strujanje s prilagodljivim brzinama (engl. *rate-adaptive streaming*) [3][16][23][24]. Na poslužitelju je pohranjen videozapis koji je podijeljen u više segmenata čija trajanja obično variraju od 2 do 10 sekundi. Svaki se segment tada kodira u različite kvalitete prikaza gdje svaki prikaz ima različitu veličinu (u bajtovima) i distribuciju kvalitete. Pomoću algoritama prilagodbe i lokalnih informacija i predviđanja, klijent odlučuje koju video reprezentaciju (ili skup reprezentacija) preuzeti, tako da se one prilagode dostupnoj propusnosti mreže i sljedećoj budućoj poziciji korisnikovog pogleda.

Da bi bile učinkovite, ranije spomenute tehnike se oslanjaju na algoritme predviđanja korisničkog vidnog okvira. Spomenute tehnike optimizacije doprinose smanjenju mrežnih resursa te je pokazano da se prilagodljivim strujanjem 360-videa uz prostornu podjelu na pločice širina prijenosnog pojasa smanjila za 72% [25]. Prilagodljivo strujanje, predviđanje korisničkog vidnog okvira i tehnike prostorne podjele videa dovode do optimizacije brzine prijenosa, ali također mogu ozbiljno narušiti performanse strujanja VR videa te degradirati iskustvenu kvalitetu korisnika (*QoE*). Strujanje videa od 360 stupnjeva temeljeno na pločicama unosi distorziju tijekom reprodukcije, tj. **kašnjenje u prebacivanju rezolucije pločica** (engl. *Motion-to-High-Resolution Latency*). Stoga je važno dobro procijeniti iskustvenu kvalitetu (*QoE*) takvog strujanja. Takva *QoE* analiza može se provesti pomoću objektivnih metrika [26][27], poput PSNR (engl. *Peak Signal-to-Noise Ratio*), različitih inačica SSIM (engl. *Structural similarity*) ili drugih, međutim takve metrike nisu uvijek u stanju mjeriti stvarnu percepciju kvalitete korisnika. Prethodno spomenute metrike optimizirane su za kvalitetu slika, gdje se vremenski utjecaji poput zaustavljanja (privremeni prekid reprodukcije medija) ili promjene kvalitete (engl. *quality switching*) ne uzimaju u obzir. U [2] autori su procijenili performanse strujanja pri različitim propusnim kapacitetima s različitim kašnjenjima. Oni su uzeli PSNR korisničkog vidnog polja (viewport) kao mjeru performansi i pokazali da je moguće poboljšati kvalitetu videozapisa, čak i s ograničenjima propusnosti i kašnjenjima. Međutim, njihovi rezultati nisu potvrđeni subjektivnim testovima.

U [28] autori su proučavali utjecaj zastoja strujanja svesmjernog videa na *QoE* i uspoređivali ga s tradicionalnim 2D videom. Rezultati pokazuju da zaustavljanje videa (engl. *stalling*) ima jak utjecaj na *QoE*. Autori su u svojoj studiji otkrili neočekivane rezultate koji su možda pokazatelj nepouzdanog postavljanja eksperimenta zato što autori ne objašnjavaju prirodu neočekivanih rezultata. Naime rezultati su pokazali da zaustavljanje videa više utječe na nervozu korisnika kada je video statičan, nego kada je dinamičan. Autori u [14] predstavili su subjektivne rezultate ispitivanja za strujanje videozapisa u 360 stupnjeva ovisnog o korisničkom pogledu. Usporedili su dva scenarija: 1) strujanje pločica izvan vidokruga u nižoj rezoluciji i 2) strujanje pločica izvan vidokruga u punoj rezoluciji, ali pri nižem omjeru signal/šum (engl. *Signal to Noise Ratio*, kratica: SNR). U svojoj studiji, strujanje pločica izvan korisničkog vidokruga u punoj rezoluciji, ali s nižim SNR-om, pružalo

je bolju subjektivnu kvalitetu videozapisa. Spomenuta rješenja pokazuju da je moguće strujati svesmjerne videozapise u većim rezolucijama (6K i više) bez značajnog povećanja potrebne širine prijenosnog pojasa i zahtjeva za dekodiranjem. Međutim, nedostaje eksperimenata sa subjektivnom procjenom iskustvene kvalitete strujanja temeljenog na pločicama, posebno za utjecaj različitih kašnjenja promjene rezolucije pločica na percipiranu kvalitetu videozapisa, s obzirom na različite scenarije. Nadalje, nijedna od navedenih studija ne bavi se aspektom simulacijske bolesti (engl. *simulator sickness*) [29]. Stručnjaci iz MPEG-a predložili su koncept regija kvalitete, a to su pravokutne regije na sferi videa koje su karakterizirane razinom kvalitete od 1 do 100. Glavna ideja ovog rješenja je da pružatelji sadržaja na temelju nekih vanjskih informacija, analizi sadržaja i statistici pozicije prikazanog polja, mogu odrediti koje će regije korisnik najvjerojatnije gledati te zatim taj isti 360-video pripremiti i kodirati u više razina kvalitete na temelju odabranih regija.

3. Strujanje videa od 360 stupnjeva

Strujanje videa od 360 stupnjeva vrlo je izazovno jer zahtijeva veliku širinu prijenosnog pojasa i naprednije načine kodiranja videozapisa. Stoga ovo poglavlje daje pregled tehnologija kojima se rješavaju izazovi za ovakva strujanja medija. Objašnjava se što je 360° video, kako nastaje i kako se kodira. Opisuju se načini njegove reprodukcije, objavljivanja te njegove varijacije. Nadalje, opisuje se novi standard za kodiranje videozapisa te njegove prednosti nad starijim standardom. Na kraju, opisuje se kodiranje 360-videa na prilagodljiv način putem protokola HTTP i uz prostornu podjelu na pločice.

3.1. 360-video

Videozapis od 360 stupnjeva je video ultra visoke rezolucije sniman u svakom smjeru s jedinstvene točke, što znači da je on u biti sferični pogled na scenu. Obično se snima pomoću instalacije više kamera ili pomoću posebne kamere koja sadrži više objektivu ugrađenih u uređaj i istovremeno snima kutove koji se preklapaju. Metodom poznatom kao video-šivanje (engl. *video stitching*) ove zasebne snimke se spajaju u jedan sferični video, a boja i kontrast svakog kadra se kalibrira kako bi bili u skladu s ostalima. Ovaj postupak se obavlja pomoću same kamere ili pomoću specijaliziranog softvera koji može analizirati vizualne i zvučne sadržaje svake kamere i sinkronizirati ih. Ukratko, jedino područje koje se ne može vidjeti jest pogled prema nosaču kamere [30].

Osnovna podjela 360-videa su monoskopski i stereoskopski 360-video. Monoskopski video od 360 stupnjeva sastoji se od jedne slike projicirane na sferu oko korisnika. Njegove prednosti su to što je pristupačniji za gledanje i snimanje jer ga se može reproducirati na pametnim telefonima i VR zaslonima na glavi. Stereoskopski video sastoji se od dvije projekcije te je najčešće prikazan u „gornjem/donjem“ (engl. *Top/Bottom*) 3D formatu. Gornja slika je prikaz lijevog oka, dok je donja slika prikaz desnog oka. Stereoskopski 360-video zahtijeva pažljiviju i detaljniju produkciju zbog toga što on bilježi i dubinu i potrebno je dobro kalibrirati te dvije slike da se prikazuju za oba oka. Prednost takvog videa je to što je on

kvalitetniji i pruža korisniku osjećaj da je on stvarno tamo, a nedostatak je to što ga se može reproducirati samo u VR zaslonima na glavi.

3.1.1. Kreiranje

Za potrebe snimanja videa od 360 stupnjeva koriste se specijalizirane svesmjerne (omnidirectional) kamere i uređaji. To su uređaji poput GoPro Omni i Odyssey kamera (koji se sastoje od više kamera instaliranih u okviru), kamere poput HumanEyes Vuze i Nokia OZO. Također postoje i ručne kamere s dvostrukim objektivima, poput Ricoh Theta S, Samsung Gear 360, Garmin VIRB 360 (Slika 3.1) i Kogeto Dot 360 koja je panoramska leća fotoaparata razvijena kao dodatna oprema za iPhone 4, 4S i Samsung Galaxy Nexus [31][32][33][34][35][36][37].



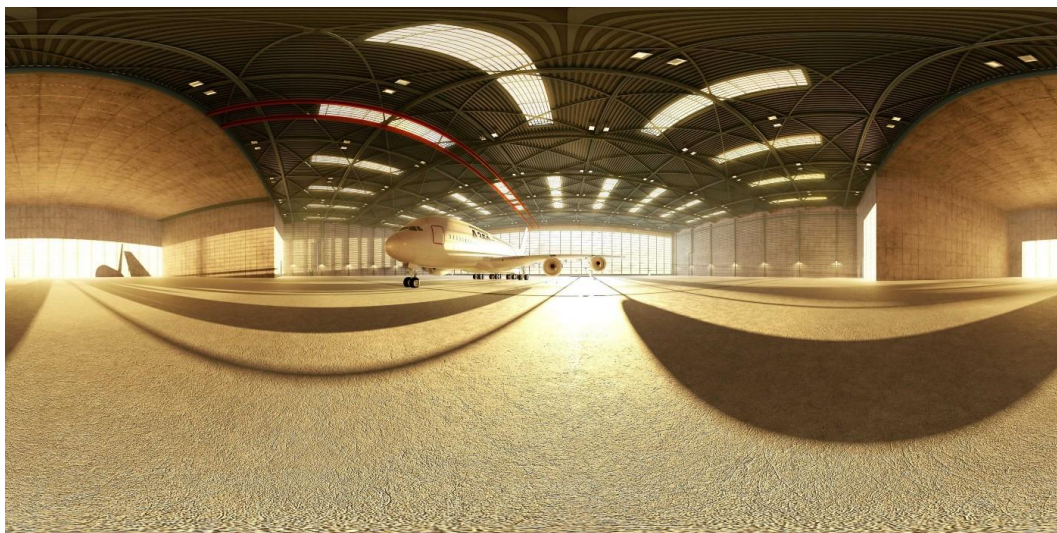
Slika 3.1 Garmin VIRB 360 kamera (slika preuzeta s [35])

3.1.2. Kodiranje

Postojeći video koderi ne mogu izravno kodirati 360-video zato što rade na principu dvodimenzionalne pravokutne slike [3]. Zbog toga se sferični video prvo treba preslikati u 2D ravninu prije samog kodiranja [25]. Postoje različiti pristupi projekcije ravnina kao što su:

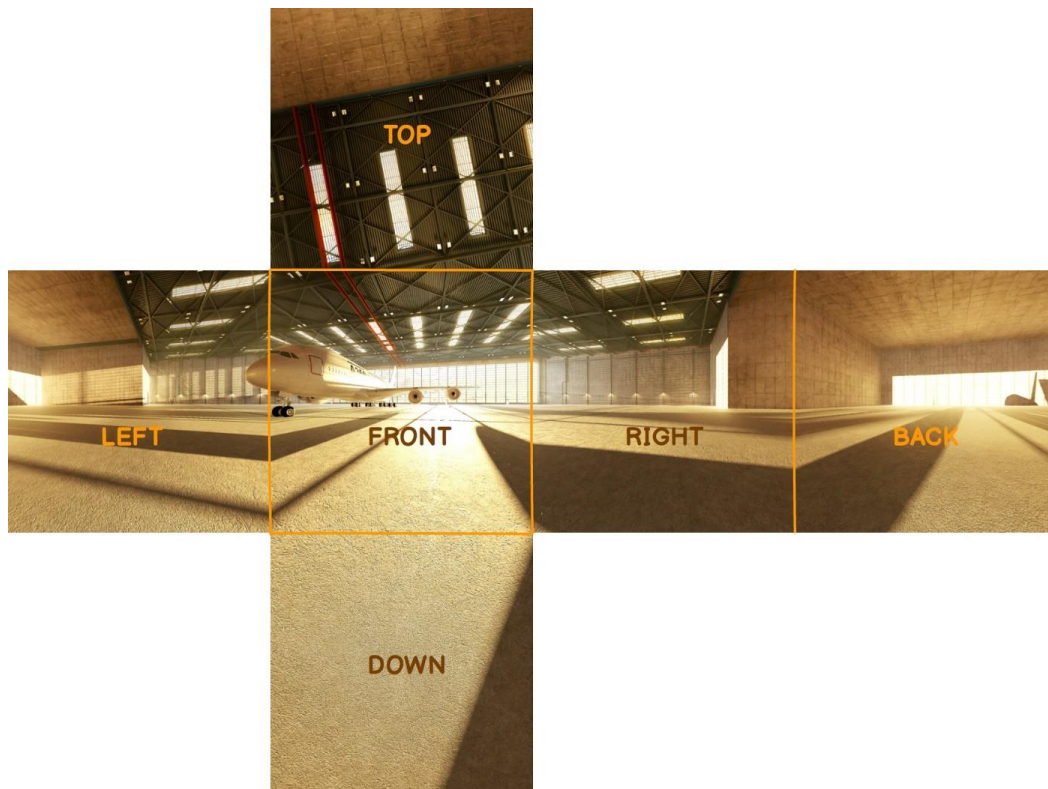
- pravokutna (engl. *equirectangular*, kratica: ERP) projekcija,
- projekcija kocke (engl. *cubemap*, kratica: CMP),
- piramidalna projekcija,
- pravokutna projekcija kocke (engl. *Equi-Angular Cubemap*, kratica: EAC).

Pravokutna projekcija (**Slika 3.2**) je najčešće korištena projekcija za preslikavanje 360 videa u 2D ravninu zbog svoje široke podrške u softverskim razvojnim okruženjima. Ova projekcija preslikava sferu u pravokutnik i rezultira velikim istezanjem slike na sjevernom i južnom polu sfere, što dovodi do redundantnih informacija na vrhu i dnu slike. Ti suvišni podaci uzrokuju smanjenu učinkovitost kodiranja i povećanu potrošnju propusnosti. YouTube je prije koristio samo pravokutnu projekciju, dok sada koristi noviju EAC projekciju [38].



Slika 3.2 Primjer pravokutne projekcije (slika preuzeta s [39])

Zbog nedostataka pravokutne projekcije, tvrtka Facebook remapira sadržaj iz pravokutnih projekcija u projekcije kocke. U računalnoj grafici, projekcija kocke je metoda mapiranja scene na šest strana kocke. Video se projicira na strane kocke i pohranjuje se kao šest kvadratnih tekstura ili se raspoređuje u šest regija jedne teksture. Projekcije kocke lakše se projiciraju, nemaju geometrijsku distorziju unutar strana kocke i pikseli su dobro raspoređeni [40]. **Slika 3.3** prikazuje primjer projekcije kocke i njenih šest kvadratnih strana.



Slika 3.3 Primjer projekcije kocke (slika preuzeta s [39])

Projekciju piramide detaljno je objavio Facebook 21. siječnja 2016., a uglavnom je bio usmjeren na VR video. Video se prikazuje u višestrukim prikazima (u Facebook slučaju 30), gdje baza piramide sadrži punu razlučivost i nalazi se točno ispred gledatelja, dok su strane prikazane s postupno smanjenom razlučivošću. Tvrtka tvrdi da je 80% smanjenje propusnosti s ovom projekcijom, a nedostatak je to što je potrebno renderirati i pohraniti mnogo više vidnog okvira [23].

Google je 14. ožujka 2017. detaljno predstavio pravokutnu projekciju kocke (EAC) [38][41]. U siječnju 2018. tvrtka je započela koristiti tu projekciju za strujanje videozapisa od 360 stupnjeva na YouTubeu. Glavni cilj EAC projekcije je distribuirati piksele što je moguće ravnomjernije po sferi tako da je gustoća informacija konzistentna, bez obzira u kojem smjeru gledatelj gleda.

3.1.3. Reprodukcija

Videozapisi od 360 stupnjeva obično se pregledavaju putem osobnih računala, mobilnih uređaja kao što su pametni telefoni ili zasebnih VR uređaja koji se montiraju na glavu i imaju svoj zaslon. Korisnici mogu pomicati videozapis klikom i povlačenjem miša, npr. na osobnom računalu, a na pametnim telefonima mogu se koristiti unutarnji senzori poput žiroskopa koji na temelju orijentacije uređaja pomiče videozapis. Zbog prednosti koje pruža žiroskop, osmišljena su kućišta za pametne telefone (poput Samsung Gear VR-a i Google Cardboard-a) koja se mogu koristiti za gledanje 360-videa u virtualnoj stvarnosti. Zaslon telefona gleda se kroz leće unutar tog kućišta, za razliku od ranije spomenutih VR uređaja s vlastitim zaslonom (HTC Vive) .

3.1.4. Objavljivanje

U ožujku 2015. YouTube je pokrenuo podršku za objavljivanje i gledanje 360-videa, s reprodukcijom na svojoj web stranici i na svojoj mobilnoj aplikaciji. Matična tvrtka Google također je najavila da će surađivati s proizvođačima kamera kako bi korisnicima olakšala prijenos 360-videa na YouTube [42]. Međutim u 2017. godini Google i YouTube su počeli promovirati alternativni stereoskopski video format pod nazivom VR180. VR180 ima ograničeno korisnikovo područje gledanja na 180 stupnjeva, ali se oglašava kao pristupačniji za proizvodnju od 360-videa i omogućuje zadržavanje veće dubine zbog toga što ne podliježe pravokutnoj projekciji [43][44]. Facebook (matična tvrtka proizvođača Oculus VR-a) slijedio je to rješenje dodavanjem podrške za videozapise od 360 stupnjeva u rujnu 2015., a potom je predstavio dizajn za vlastiti sustav 360-kamera poznat kao Facebook Surround 360. Kasnije, u ožujku 2017. Facebook je objavio da je do tog datuma na njihovu platformu objavljeno više od milijun 360-videa [45].

3.1.5. Varijacije

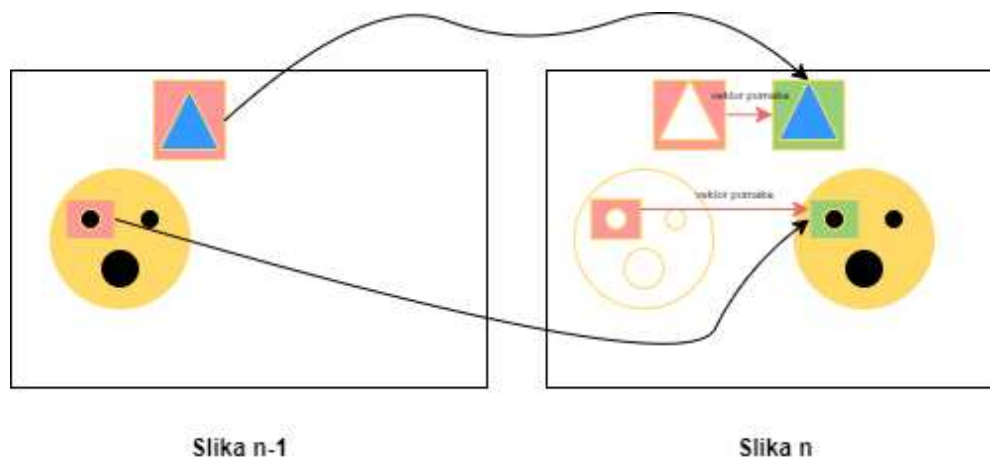
Postoji nekoliko varijacija 360-videa a najčešća podjela je na:

- Video od 180-stupnjeva – stereoskopski video koji obuhvaća samo 180 stupnjeva vidnog polja i omogućuje zadržavanje veće dubine zbog toga što ne podliježe pravokutnoj projekciji
- 6DoF video – stereoskopski video od 360 stupnjeva koji također bilježi dubinu i omogućava šest stupnjeva slobode u navigaciji u snimljenom okruženju
- Volumetrijski video – tehnika koja bilježi trodimenzionalni prostor, kao što je lokacija ili prostor i prikazuje ga putem VR-a

3.2. Video kodiranje visoke učinkovitosti

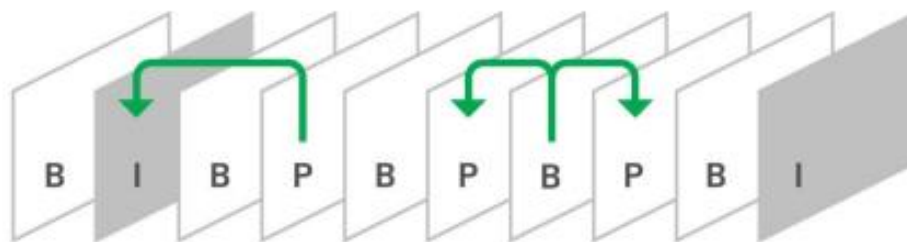
Video kodiranje visoke učinkovitosti (engl. *High Efficiency Video Coding*, kratica: HEVC), također poznato kao H.265, novi je standard za kompresiju videozapisa razvijen od tima JCT-VC (engl. *Joint Collaborative Team on Video Coding*) [46]. HEVC je razvijen s ciljem pružanja dvostruke učinkovitosti kompresije nad prethodnim, najčešće korištenim standardom **naprednim kodiranjem videozapisa** (engl. *Advanced Video Coding*, kratica: AVC) poznatog i kao H.264. Iako se rezultati učinkovitosti kompresije razlikuju ovisno o vrsti sadržaja i postavkama koda, HEVC obično može komprimirati video dvostruko efikasnije od AVC-a [46]. Krajnji korisnici mogu iskoristiti poboljšanu učinkovitost kompresije na jedan od dva načina (ili kombinaciju oba). Prvi način je da na identičnoj razini kvalitete, HEVC omogućuje komprimiranje videozapisa u datoteku koja je dvostruko manje veličine nego kod AVC-a. Drugi način je kada se video komprimira na istu veličinu datoteke kao AVC, HEVC pruža znatno bolju vizualnu kvalitetu. Također, video kodiranje visoke učinkovitosti podržava razlučivost do 8192 x 4320 piksela, uključujući 8K UHD.

Videozapis se sastoji od niza nepomičnih slika (okvira) koji su prikazani dovoljno brzo. Frekvencija promjene slike je brzina osvježavanja okvira (engl. *frame rate*) i izražena je u okvirima u sekundi (engl. *frames per second*). Kompresijom videozapisa nastoji se smanjiti ili ukloniti suvišne informacije iz toka videozapisa, tako da se mogu pohraniti ili slati preko mreže što učinkovitije. Algoritmi koji se koriste za uklanjanje suvišnih informacija čine proces kodiranja, a metoda kojom se komprimirani video reproducira i vraća nazad u što sličnije izvornom stanju, naziva se dekodiranje [47]. Svaki videozapis sadrži prostornu i vremensku redundanciju. Pronalaženje i uklanjanje prostorne redundancije vrši se unutar svake pojedinačne slike, bez referenci na prethodne ili sljedeće slike. Za uklanjanje prostorne redundancije najčešće se koristi transformacijsko kodiranje. Vremenska redundancija se pronalazi uklanja pomoću usporedbe trenutne i prethodne slike. Koristi se sličnost slika u nizu što se naziva diferencijalno kodiranje. Pomak u slici je problem za diferencijalno kodiranje. Ako su slike jako slične, zbog pomaka njihova razlika sadrži više informacija od same slike te ih se ne može direktno diferencijalno kodirati. Kako bi se riješio taj problem koristi se kompenzacija gibanja (engl. *motion compensation*). Funkcionira na način da se za svaki blok u slici, traži najbliži blok u prethodnoj slici, a razlika položaja (u pikselima) između ova dva bloka naziva se vektor pomaka (engl. *motion vector*). **Slika 3.4** prikazuje ilustraciju kompenzacije gibanja i način na koji funkcioniraju vektori pomaka. Vektori pomaka za sve blokove šalju se dekoderu i tako se konstruira slika slična prethodnoj, te se na njoj vrši diferencijalno kodiranje. Današnji najčešći princip kodiranja videa je hibridno kodiranje videa. To je diferencijalno kodiranje slika s kompenziranim gibanjem, te transformacijsko kodiranje signala razlike.



Slika 3.4 Ilustracija kompenzacije gibanja i vektora pomaka

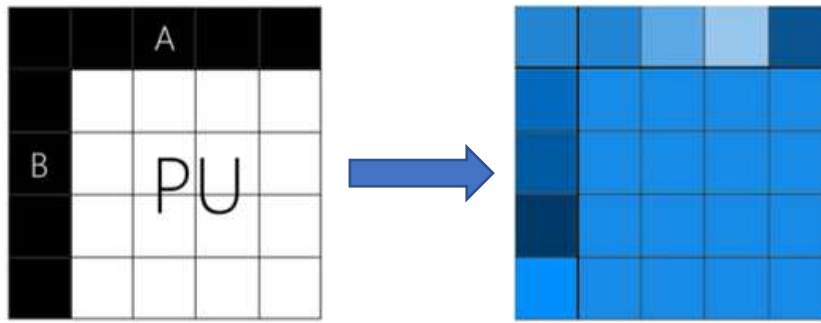
Tijekom postupka video kodiranja, koder može koristiti različite strategije predikcije okvira. Stoga postoji nekoliko vrsta video okvira, a to su I-okviri (engl. *I-frames*), P-okviri (engl. *P-frames*) i B-okviri (engl. *B-frames*). Kada se ove različite vrste okvira koriste u kombinaciji, brzina prijenosa videa se može smanjiti traženjem vremenske i prostorne redundancije. Na taj se način pikseli ili blokovi piksela koji se često ne mijenjaju iz okvira u okvir ili su identični pikselima ili blokovima piksela prijašnjeg okvira mogu obraditi na inteligentan način [47]. Diferencijalno (predikcijsko) kodiranje akumulira pogrešku, stoga je potrebno s vremena na vrijeme poslati puni okvir, tzv. I-okvir. I-okvir, ili intra-kodirani okvir je samostalni okvir koji se može nezavisno dekodirati bez reference prethodnog ili nadolazećeg okvira. Prva slika u videozapisu je uvijek I-okvir i takav okvir djeluje kao polazišna točka ako je preneseno strujanje bitova oštećeno. Također može se koristiti za premotavanje videa i detekciju promjene scene. Nedostatak I-okvira je u tome što oni troše mnogo više bitova i ne nude uštedu kompresije. P-okvir, ili predikcijski inter-kodirani okvir, koristi prethodne I-okvire ili P-okvire za kodiranje slike. P-okviri obično zahtijevaju manje bita od I-okvira, ali su osjetljivi na pogreške u prijenosu zbog njihove značajne ovisnosti o ranijim referentnim okvirima. Tradicionalni koderi koriste kodiranje I- i P- okvira, dok noviji uvode i treći B-okvir. B-okvir, ili dvosmjerno predikcijski inter-kodirani okvir, koji osim prethodnih, koristi i buduće I- i P- okvire za kodiranje slike. B-okvir se izračunava na temelju prethodnog i sljedećeg I- i P- okvira, tako da oni moraju biti poslani ranije. **Slika 3.5** prikazuje tipičan redoslijed I-, P- i B- okvira.



Slika 3.5 Redoslijed I-, P- i B- okvira

Kodiranje se započinje I-okvirom te se taj okvir dijeli na mala područja veličine 16x16 piksela – koja su se u prethodnom standardu (AVC-u) zvala „makroblokovi“. Tada se prelazi na sljedeću sličicu i uspoređuju se njeni makroblokovi s onim u I-okviru. Razlika u H.265 standardu je to što je makroblok preimenovan u stablo jedinica za kodiranje (engl. *Coding tree unit*, kratica: CTU) i njegova veličina može biti od originale 16x16 piksela pa sve do 64x64 piksela. Ovo je veliko poboljšanje, zato što 4K videozapisi zahtijevaju učinkovitije kodiranje, a upravo to daju veći makroblokovi. CTU se dalje može rekurzivno podijeliti na četiri manje kvadratna dijela koji se zovu jedinice za kodiranje (engl. *Coding Unit*). Drugo veliko poboljšanje odnosi se na kompresiju unutar samog okvira. Drugim riječima, uspoređuju se područja unutar samog okvira i traže se redundancije. A poboljšanje je povećanje smjerova predviđanja. Jedinice za kodiranje stabla, mogu se dalje podijeliti, na manje jedinice za kodiranje koje su 8x8 piksela. A one se mogu još više podijeliti i slagati na različite načine koji se nazivaju jedinice predviđanja (engl. *prediction units*, kratica: PU). Na taj način se vrijednosti piksela u bloku mogu matematički generirati, umjesto da ih se pohranjuje. I tada se primjenom različitih načina predviđanja unutar ove strukture može masivno smanjiti veličina svakog okvira. Jedan od primjera ovih metoda se zove DC predikcijski mod koja radi na osnovi prosjeka boja okolnih piksela.

Uzevši za primjer jedinicu predviđanja veličine 4x4 piksela koja je okružena s dvije grupe piksela nazvane blok A i blok B, metoda DC će u osnovi ispuniti jedinice predviđanja prosjekom boja okolnih piksela (**Slika 3.6**). Ovo se dobro primjenjuje u scenama na otvorenom, na primjer s plavim nebom, gdje nema pravog uzorka nego je u osnovi samo jedna boja. Razlika HEVC-a s AVC-om je što stariji standard AVC koristi samo 9 modova predviđanja, dok HEVC ima na raspolaganju 35 različitih načina. Prednosti ovome su brojne, uključujući i daleko veću točnost kada se koriste veliki blokovi, tj. općenito mnogo kvalitetniju kvalitetu kompresije 360- videa [46].



Slika 3.6 Primjer DC metode predikcije

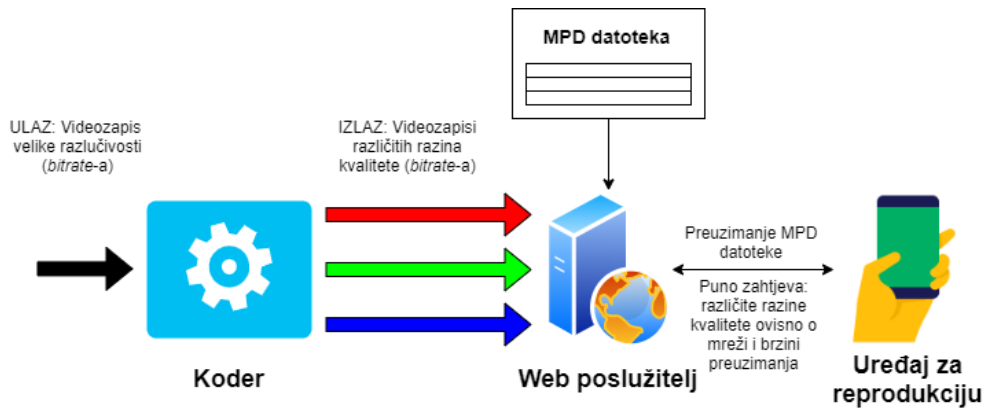
Popularna platforma za reprodukciju 360- videa, YouTube, sada dopušta učitavanje videozapisa kodiranih s H.265 standardom, dok ga još većina web preglednika izvorno ne podržava [48]. HEVC je prihvaćen kao zadani kodek na 4K Blue-ray uređajima i može se reproducirati na PC-u i na pametnim telefonima, uz softversko ili hardversko dekodiranje.

3.3. Prilagodljivo strujanje putem protokola HTTP

Kako bi se osigurala prihvatljiva iskustvena kvaliteta korisnika, često se za isporuku 360° videa preko Interneta primjenjuju načela prilagodljivog strujanja putem protokola HTTP (*HAS*). U *HAS*-u je video kodiran s različitim razinama kvalitete i vremenski je podijeljen u više segmenata s duljinom trajanja od 2 do 10 sekundi [49]. Koristeći se heurističkim prilagođavanjem brzine, klijent zahtijeva te segmente po najprikladnijoj kvaliteti na temelju dostupne širine propusnog pojasa i popunjenosti spremnika (engl. *buffer*). Klijent prvo pohrani segmente u spremnik te ih tada dekodira u linearnom redosljedju i reproducira na korisnikovom uređaju.

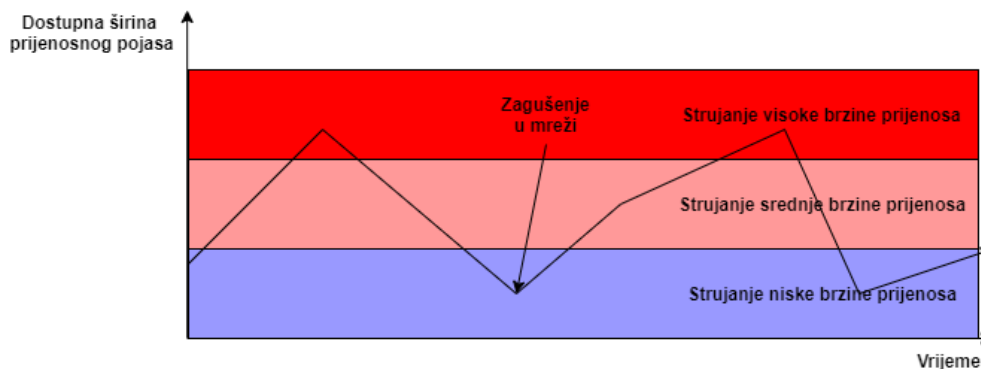
Davatelji sadržaja obično koriste *HAS* za isporuku i strujanje 360° videa na sličan način kao i za tradicionalne videozapise. Prvo se sadržaj preslikava u dvodimenzionalni prikaz koristeći ranije spomenutu pravokutnu projekciju. Dobiveni video se tada kodira u više razlučivosti, vremenski se dijeli i stavlja na raspolaganje na HTTP server.

Prilagodljivo strujanje je tehnika koja se koristi za strujanje višemedijskih podataka preko računalnih mreža. Dok su u prošlosti većina tehnologija audio i video strujanja koristila protokole poput RTP-a (engl. *Real-time Transport Protocol*) [50] i RTSP-a (engl. *Real Time Streaming Protocol*) [51], današnje prilagodljive tehnologije strujanja gotovo se isključivo temelje na HTTP-u i dizajnirane su tako da učinkovito rade na velikim distribuiranim HTTP mrežama kao što je Internet [52]. Prilagodljivo strujanje putem protokola HTTP radi tako što u stvarnom vremenu procjenjuje širinu propusnog pojasa i kapacitet *CPU*-a (engl. *Central processing unit*) i u skladu s tim prilagođava kvalitetu strujanja medija [53]. To zahtijeva upotrebu kodera koji može kodirati izvorni medij (video ili audio) s više brzina prijenosa, tj. kvalitete samog medija. *Player* na strani klijenta [54] se prebacuje između kodiranih kvaliteta ovisno o raspoloživim resursima [55]. Rezultat je efikasno i ograničeno punjenje spremnika, brzo vrijeme pokretanja i dobra krajnja iskustvena kvaliteta korisnika prilikom strujanja sadržaja u uvjetima varijabilnih mrežnih performansi [56]. Prvo, klijent preuzme *manifest* datoteku zvanu još i opis medijske prezentacije (engl. *Media presentation description*, kratica: MPD) koja opisuje dostupne segmente strujanja i njihove brzine prijenosa. **Slika 3.7** shematski prikazuje arhitekturu prilagodljivog strujanja putem protokola HTTP.



Slika 3.7 Shema prilagodljivog strujanja putem protokola HTTP [57]

Tijekom pokretanja strujanja, klijent najčešće zahtijeva segmente s najnižom kvalitetom, ali može krenuti i od srednje kvalitete. Ako klijent utvrdi da je propusnost mreže veća od brzine prijenosa preuzetog segmenta, tada će zahtijevati segment veće brzine, tj. bolje kvalitete. Kasnije, ako klijent utvrdi da se propusnost mreže smanjila, zatražit će segment s nižom brzinom prijenosa (**Slika 3.8**).



Slika 3.8 Ovisnost propusnosti mreže o preuzetim segmentima [57]

Algoritam za prilagodljivi *bitrate* (engl. *Adaptive Bitrate*, kratica: ABR) u klijentu obavlja ključnu funkciju, na temelju trenutnog stanja mreže odlučuje koji će se segment preuzeti. U komercijalnoj upotrebi je nekoliko vrsta ABR algoritama [52]:

- algoritmi temeljeni na propusnosti: za donošenje odluka koriste postignutu propusnost u prethodnim preuzimanjima,
- algoritmi temeljeni na spremnicima: koriste samo trenutnu razinu spremnika klijenta (BOLA [58] u *dash.js*-u),
- hibridni algoritmi koji kombiniraju obje vrste podataka (informacije o propusnosti i spremniku), primjer takvog algoritma je DYNAMIC [59] u *dash.js*-u.

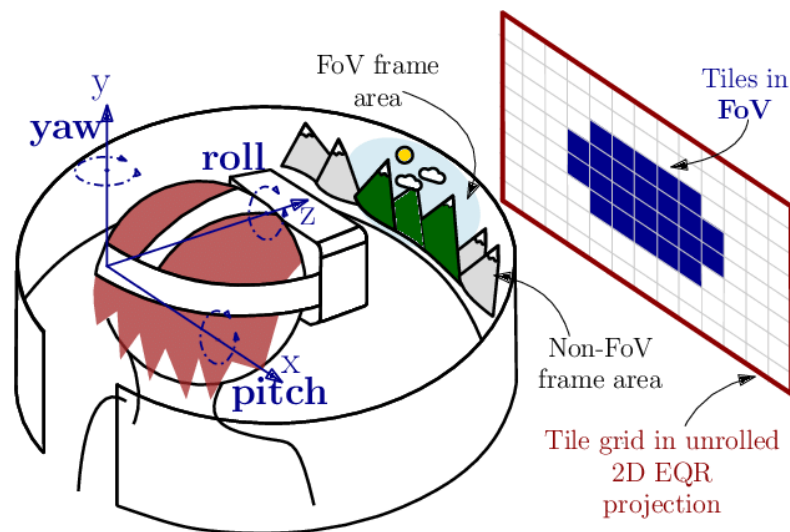
Postoji mnogo implementacija prilagodljivog strujanja putem protokola HTTP, a jedina koja je međunarodno standardizirana i koja će se u ovom radu koristiti je MPEG-DASH [60]. To je prilagodljivo strujanje temeljeno na protokolu HTTP i razvijeno je od grupe MPEG (*Moving Picture Experts Group*). Rad na DASH tehnologiji započeo je 2010., a kao ISO/IEC 23009-1 standard priznat 2012. godine, a posljednjih je godina imao veliku primjenu, npr. YouTube i Netflix. Cilj standardiziranog rješenja za prilagodljivo strujanje je pružiti povjerenje tržištu da se ono može usvojiti za univerzalnu primjenu, za razliku od sličnih ali manje interoperabilnih rješenja kao što su HLS (engl. *HTTP Live Streaming*) tvrtke Apple, *Smooth Streaming* tvrtke Microsoft ili HDS (engl. *HTTP Dynamic Streaming*) tvrtke Adobe.

Prilagodljivo strujanje putem protokola HTTP pruža korisnicima najbolje iskustvo zbog toga što se strujanje automatski prilagođava svim promjenama u korisnikovoj mreži i uvjetima reprodukcije [61]. Također ono uvodi nove prednosti naspram starijih, tradicionalnih strujanja. Prvo, budući da je tehnologija strujanja izgrađena povrh HTTP-a, suprotno strujanju temeljenom na RTP-u, paketi nemaju poteškoća zaobići vatrozid i NAT (engl. *Network address translation*) uređaje. Drugo, budući da je strujanje putem protokola HTTP usmjereno isključivo na klijenta, sva logika prilagodbe također se nalazi na klijentu. To smanjuje potrebu za postojanim vezama između poslužiteljske i klijentske aplikacije. Nadalje, poslužitelj nije obavezan održavati podatke o stanju sesije o svakom klijentu, što poboljšava skalabilnost. Konačno, može se koristiti već postojeća infrastruktura za isporuku protokola HTTP, kao što su predmemorije i poslužitelji protokola HTTP [62][63].

Postprodukcijske kuće, mreže za isporuku sadržaja i studiji koriste tehnologiju prilagodljivog strujanja kako bi potrošačima omogućili kvalitetniji video sadržaj koristeći manje resursa. Kako količina videa raste, mreže za isporuku sadržaja i pružatelji video sadržaja mogu i dalje korisnicima pružiti vrhunski doživljaj gledanja. Medijske tvrtke aktivno koriste prilagodljivo strujanje i to je u suštini postala standardna praksa za napredne pružatelje usluga.

3.4. Prilagodljivo strujanje uz prostornu podjelu na pločice

Budući da je kvaliteta tradicionalnog videozapisa fiksna za cijeli prikaz, raspoloživa širina propusnog pojasa za prijenos takvog videa je nedovoljno iskorištena. Korisnik kada nosi zaslon montiran na glavu ima ograničen prikaz, tj. vidno polje koje se naziva i vidni okvir. To znači da se značajan dio širine prijenosnog pojasa troši na sadržaj koji se nikad ne konzumira, odnosno koji korisnik nikada ne vidi. Da bi se riješio taj izazov, video sadržaj se može podijeliti na $m \times n$ broj pločica koje se dalje mogu kodirati u više razina kvalitete. Klijent tada može zatražiti svaku pločicu u različitoj kvaliteti, postavljajući prioritet pločicama unutar vidnog okvira koje su veće rezolucije. **Slika 3.9** prikazuje koncept tehnologije prilagodljivog strujanja uz prostornu podjelu na pločice u VR-u. Korisnik u $3DoF$ zaslonu montiranom na glavi može pomicati glavu u 3 osi (x,y,z os). U svakom trenutku korisnik u vidnom polju sadrži pločice visoke rezolucije, dok izvan svog vidnog polja pločice se prikazuju u nižoj rezoluciji.

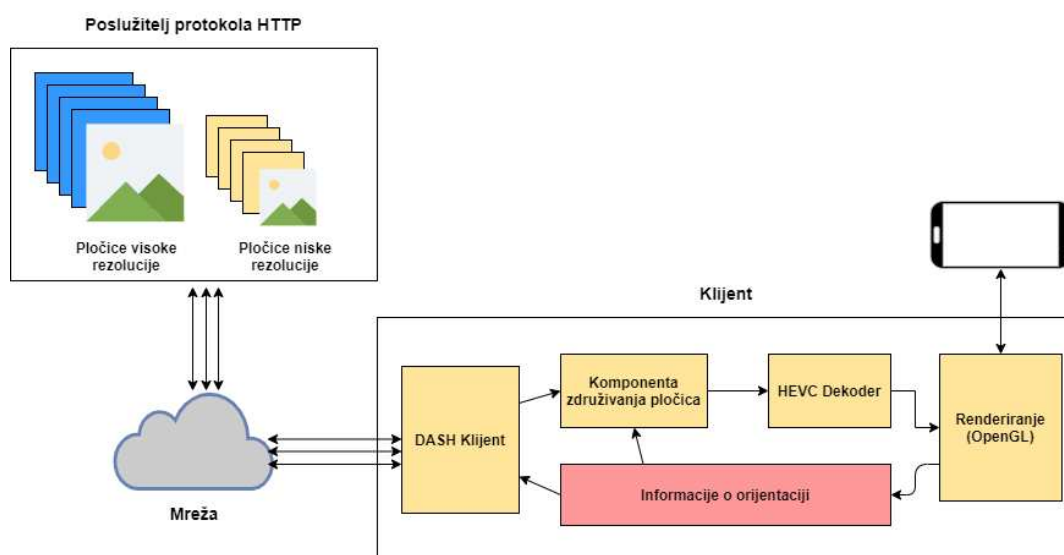


Slika 3.9 Koncept tehnologije prilagodljivog strujanja uz prostornu podjelu na pločice u VR-u (slika preuzeta s [64])

Prilagodljivo strujanje sadržaja u VR-u koje se temelji na prostornoj podjeli na pločice predstavlja niz izazova. Prvo, prilagodljivo strujanje uz prostornu podjelu na pločice sklono je pokretima korisnika. Čak i ako spremnik sadrži segmente od samo dvije sekunde, korisnik bi naglim pokretima senzora na zaslonu na glavi, unutar scene mogao privremeno uočiti slabiju kvalitetu videa koja se još nije stigla prebaciti u veću kvalitetu. Stoga su potrebna točna predviđanja budućeg korisnikovog vidnog okvira. Drugo, heuristika prilagođavanja brzine na strani klijenta, koja u svakom trenutku pokušava pružiti visoku kvalitetu videozapisa i izbjeći zatrpanost spremnika, mora uzeti u obzir novu prostornu dimenziju u donošenju odluka što povećava njenu složenost. Treće, budući da se zahtjevi putem protokola HTTP prenose jedan po jedan, vjerojatnije je da će kašnjenje mreže više utjecati na spomenuta rješenja (temeljena na prostornoj podjeli na pločice). To se posebno odnosi na mobilne mreže u kojima su vremena povrata (engl. *round trip time*, kratica: RTT) za 3G/HSPA+ mreže od 50 do 200 milisekundi, a za 4G/LTE mreže od 30 do 70 milisekundi [30]. Podjela videa na pločice smanjuje memorijske zahtjeve jer se pločice komprimiraju i rekonstruiraju neovisno [66].

Vidni okvir definiran je kutom područja gledanja zaslona na glavi, koje je obično između 100 i 120 stupnjeva. Kako bi se osigurala kvaliteta usluge, razlučivost piksela na prikazanom vidnom okviru je visoka, obično 4K (3840 x 2160). Stoga, za izdvajanje 4K korisničkog prikaza, tj. vidnog okvira od cijelih 360 stupnjeva videozapisa, potpuni videozapis bi trebao biti najmanje 12K (11520 x 6480 piksela). Većina tih informacija normalnim strujanjem je ignorirana od video *playera*, jer korisnik vidi samo dio od potpunog prikaza. Uz to, dobra kvaliteta usluge zahtijeva da broj okvira u sekundi videa bude kao i brzina osvježavanja VR zaslona na glavi koja je obično oko 100 sličica u sekundi. Za usporedbu, brzina prijenosa videozapisa od 8K razlučivosti i 60 sličica po sekundi kodiranog HEVC-om je oko 100Mbps [67].

Strujanje uz prostornu podjelu na pločice funkcioniра na sljedeći način. Informacije o trenutnoj orijentaciji korisnika prosljeđuju se u DASH klijentu i komponenti združivanja pločica. DASH klijent tada pokreće postupak odabira pločica koristeći te informacije o orijentaciji i odlučuje koje pločice treba zahtijevati pri visokoj, a koje pri niskoj rezoluciji. Drugi parametar koji utječe na postupak odabira pločica je *FoV* krajnjeg uređaja jer se mogu zatražiti različiti omjeri visokokvalitetnih i niskokvalitetnih pločica. Nakon preuzimanja segmenata, komponenta združivanja stvara jedno pojedinačno korisničko strujanje bitova ovisno o vidnom okviru. Potreban je HEVC softverski dekodер koji obrađuje strujanje bitova i prosljeđuje dekodirane slike u renderer (na primjer u *Nokiatech*⁴ rješenju taj renderer je temeljen na OpenGL-u [68]) kako bi se stvorio korisnički prikaz koji će se izvoditi na Android uređaju. **Slika 3.10** prikazuje prethodno navedeni princip rada prilagodljivog strujanja uz prostornu podjelu na pločice.



Slika 3.10 Pregled rada sustava prilagodljivog strujanja uz prostornu podjelu na pločice

Opis medijske prezentacije (MPD) na HTTP poslužitelju koristi MPEG DASH deskriptore prostornih odnosa (engl. *Spatial Relationship Descriptors*, kratica: SRD) [57] koji pružaju prostornu lokaciju pojedinačno ponuđenih strujanja unutar globalnog koordinatnog sustava, kao i odnos između pločica različitih razlučivosti. Esencijalno za prilagodljivo strujanje uz prostornu podjelu na pločice je to što SRD daje klijentu razumijevanje o tome kako su pločice postavljene i kodirane.

⁴ Nokiatech OMAF GitHub stranica: <https://github.com/nokiatech/omaf>

Veličina spremnika klijenta koja se koristi za prilagodljivo strujanje određuje stabilnost i agilnost sustava. U tradicionalnim strujanjima, koja nisu uživo, za prilagođavanje s različitim propusnostima i za rješavanje negativnih događaja, tj. prekida reprodukcije koriste se veći spremnici od nekoliko sekundi, tipično između 2 i 10 sekundi [15]. Međutim, veća veličina spremnika također utječe na prilagodljivost vidnog okvira. Potrebno je imati na umu da se sadržaj niske razlučivosti prikazuje korisniku kad god se vidni okvir promijeni preko dopuštenog stupnja tijekom reprodukcije određenog segmenta. Između ostalog, promjena vidnog okvira ovisi o omjeru visokokvalitetnih i niskokvalitetnih pločica. Međutim, važno je da se u promijenjenom vidnom okviru pločice niske rezolucije prebace, što je brže moguće na pločice visoke rezolucije. Ako se koriste veliki spremnici, bilo koji segmenti koji su već preuzeti se odbacuju i ponovo preuzimaju u odgovarajućoj verziji, što izaziva dodatan prijenos podataka i trošenje resursa mreže. Također je moguće da se prilagodba pločica s niske rezolucije na visoku odgodi dok se svi segmenti koji su već preuzeti ne reproduciraju, što kasnije dovodi do loše iskustvene kvalitete korisnika. U stvari, kod strujanja 360-videa ovisnog o promjeni vidnog okvira, prethodno spomenute činjenice nisu poželjne jer se vidni okvir brzo i često mijenja. Stoga veličina spremnika mora biti minimalna.

4. OMAF standard

Razvojem virtualne stvarnosti neizbježno je osmišljeno mnoštvo različitih rješenja VR aplikacija koja nisu interoperabilna. To zahtijeva standardizaciju, čiji je cilj omogućiti uređajima i uslugama različitih proizvođača da su međusobno kompatibilni. Nedvojbeno, prvi takav standard je *Omnidirectional Media Format* (kratica: OMAF) razvijen od strane MPEG grupe te će u ovom poglavlju on biti opisan.

4.1. Općenito o OMAF-u

Grupa MPEG započela je razvoj VR standarda u listopadu 2015. godine. Prvo izdanje OMAF standarda završeno je u listopadu 2017. godine na 117. sjednici MPEG-a u Ženevi te je takav format aplikacija nazvan „superformat“ [69]. Nazvan je tako zato što spaja odabrane tehnološke komponente iz MPEG-a i drugih standarda te osigurava veću interoperabilnost između aplikacija, što pomaže rastućim potrebama korisnika za boljim i bržim višemedijskim rješenjima [69]. OMAF definira medijski format koji omogućuje implementaciju aplikacija za svesmjerne medije usredotočujući se na video od 360 stupnjeva, slike, zvuk i tempirani tekst. Ovaj rad usredotočen je na video i neće se raspravljati o drugim vrstama medija.

OMAF podržava samo „jednostavniju“ verziju VR-a koja omogućava tri stupnja slobode (3DoF). OMAF također podržava ranije spomenuto prilagodljivo strujanje, u kojem su dijelovi u korisničkom vidokrugu veće kvalitete slike nego oni koji su izvan područja gledanja. Štoviše, sljedeća verzija standarda je u fazi izrade, a omogućavat će nove funkcionalnosti i optimizirat će isporuku svesmjernih medija. Očekivane nove funkcionalnosti uključuju mogućnost prebacivanja između više točaka gledanja, od kojih svaka točka predstavlja jednu 360° kameru, a očekuje se i podrška za šest stupnjeva slobode (6DoF).

Izrada OMAF sadržaja uključuje predprocesiranje, kodiranje i enkapsulaciju datoteka i segmenata. Najprije, predprocesiranje nužno uključuje video-šivanje i projekciju, a može uključivati i pakiranje ovisno o regiji. OMAF Player vrši dekapulaciju medijskih datoteka i segmenata, dekodiranje te prikazivanje medija.

Ključne tehnologije za enkapsulaciju datoteka i segmenata i njihovu isporuku u OMAF standardu su ISO *Base Media File Format* (ISO/BMFF, ISO/IEC 14496-12) i prilagodljivo strujanje putem protokola HTTP (DASH, ISO/IEC 23009-1). OMAF na kompatibilan način specificira format datoteke i DASH ekstenzije kako bi se omogućilo učinkovito signaliziranje i isporuka svesmjernih medija. Također ovaj standard specificira signalizaciju i isporuku svesmjernog medija preko MPEG *media transporta* (MMT, ISO/IEC 23008-1) [70].

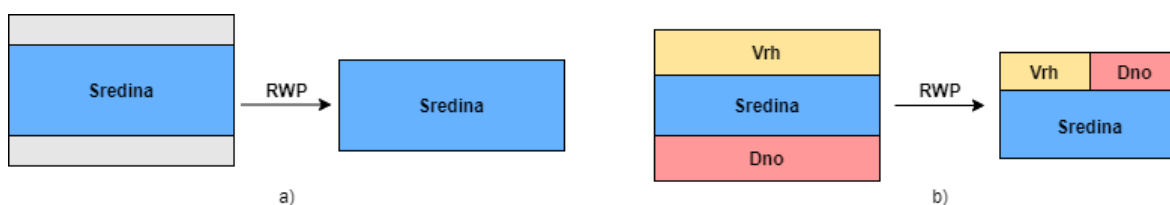
4.2. Formati prikazivanja svesmjernog videa

OMAF koristi desni koordinatni sustav, gdje korisnik gleda iz sredine sfere prema van, tj. prema unutarnjoj površini sfere na kojoj je video preslikan. OMAF specificira globalne koordinatne osi koje su podijeljene između svih vrsti medija koje se zajedno renderiraju i koriste se za određivanje početne orijentacije gledanja. Svaki videozapis može koristiti svoje lokalne koordinatne osi. Meta podaci rotacije (x, y i z podaci) određuju odnos između globalnih i lokalnih koordinatnih osi. Primjena neusklađenih globalnih i lokalnih koordinatnih osi može imati svoje prednosti. Na primjer, za ispravljanje horizonta koji u svesmjernim videozapisima mora biti točno vodoravan [78].

OMAF specificira potporu za dvije vrste projekcija, pravokutnu projekciju (*ERP*) i kubnu projekciju (*CMP*). Također su uzeti u obzir i drugi formati projekcije, ali nije dogovoreno da oni budu uključeni u OMAF standard. Proces ERP-a sličan je običnom generiranju 2D karta svijeta, ali zbog suprotne perspektive gledanja, s lijeve strane je istok, a ne zapad. U ERP-u korisnik gleda iz središta sfere van prema unutarnjoj površini sfere, dok za kartu svijeta korisnik gleda izvan sfere prema njevoj vanjskoj površini. U kubnoj projekciji signal sfere se pravocrtno projicira u šest kvadrata, koji su postavljeni tako da tvore pravokutnik s omjerom širine 3:2 u odnosu na visinu, pri čemu se neke strane okreću kako bi se maksimizirao kontinuitet preko njihovih rubova.

OMAF podržava monoskopski i stereoskopski svesmjerni video sadržaj. Stereoskopski sadržaj ostvaruje se pakiranjem okvira, a podržavaju se tri vrste pakiranja ovisnih o razmješčaju okvira: jedan kraj drugog (engl. *side-by-side*), jedan iznad drugog (engl. *top-bottom*), vremensko umetanje (engl. *temporal interleaving*) [70].

Pakiranje regija (engl. *Region-Wise Packing*, kratica: RWP) je izborni korak nakon projekcije (na strani pružatelja sadržaja). Omogućuje razne manipulacije (promjena veličine, premještanje, rotacija za 90, 180 i 270 stupnjeva, okomito i vodoravno zrcaljenje) bilo koje pravokutne regije prije samog kodiranja [70]. **Slika 4.1** prikazuje dva primjera RWP-a za pravokutnu projekciju.



Slika 4.1 Primjeri RWP-a za pravokutnu projekciju; a) određivanje sadržaja koji ne pokriva cijelu sferu; b) kompenzacija prekomjernog uzorkovanja oko polova pravokutno projicirane slike [69]

RWP se može koristiti u mnoge svrhe. Može točno odrediti sadržaj koji ne pokriva cijelu sferu i omogućuje generiranje videozapisa koji je podijeljen u regije različitih razlučivosti. Također, može kompenzirati prekomjerno uzorkovanje oko polova pravokutno projicirane slike te prilagodljivo raspoređuje strane kvadrata kod projekcije kocke.

OMAF standard također daje podršku svesmjernim medijima koji su tipa „ribljeg oka“ (engl. *fisheye*). *Fisheye* video ne upotrebljava projekciju niti RWP. Umjesto toga, za svaku sliku, kružne slike snimljene *fisheye* kamerom izravno su smještene u pravokutnik. U usporedbi s projiciranim svesmjernim video, glavna prednost *fisheye* videa je njegova podrška s jeftinim, svesmjernim sadržajima kreiranim od strane korisnika na mobilnim uređajima.

4.3. Proširenja za formate za svesmjerni video

ISOBMFF je popularan medijski format spremnika za zvuk, video i tempirani tekst [69]. ISOBMFF datoteke često se nazivaju MP4 datotekama. Osnovni blok u ISOBMFF-u naziva se kutija (engl. *box*), a ona je zapravo struktura podataka koja se sastoji od:

- četveroznakovnog koda koji označava format podataka (engl. *four-character-code*, kratica: 4CC)
- brojem bajtova kutije
- i sadržaja čiji je format određen vrstom okvira i koji može sadržavati druge kutije

ISOBMFF datoteka se sastoji od niza kutija na razini datoteke. Svaki tok tempiranih medija ili metapodataka logički se pohranjuje u zapis. Uzorak svakog zapisa opisuje format kodiranja i enkapsulacije te uključuje i tip podataka 4CC i sadržane okvire koji pružaju dodatne informacije o formatu ili sadržaju zapisa.

Video s ograničenim pristupom (engl. *restricted entry sample video*, kratica: *resv*) koristi se za videozapise koji zahtijevaju postupke naknadne obrade nakon dekodiranja kako bi se ispravno prikazale. Vrsta naknadne obrade određuje se pomoću jedne ili više vrsta shema temeljenih na *resv*-u. OMAF zahtijeva upotrebu videozapisa s ograničenim pristupom za svesmjerne videozapise. **Tablica 4.1** navodi i opisuje vrste shema za videozapise s ograničenim pristupom koje su navedene u OMAF standardu.

Tablica 4.1 Vrste shema za videozapise s ograničenim pristupom

| Vrsta | Opis |
|-------------|--|
| podv | Vrsta otvorene sheme za bilo koji projicirani svesmjerni videozapis bez ograničenja na format projekcije ili RWP. Dopusćeni su i monoskopski i stereoskopski sadržaj, a pokrivenost sadržaja može biti manja od 360. |
| erpv | Vrsta zatvorene sheme koja ograničava 'podv' samo na pravokutnu projekciju i maksimalno jedno pakirano područje po prikazu za RWP. |
| ercm | Zatvorena shema koja označava ograničavanje 'podv' na pravokutnu ili projekciju kocke. |
| fodv | Vrsta otvorene sheme za bilo koji <i>fisheye</i> svesmjerni videozapis |

Metapodaci za projicirani svesmjerni videozapis pohranjuju se kao okviri u unosu uzorka. Metapodaci mogu uključivati format projekcije, vrstu pakiranja okvira, RWP metapodatke, rotacijske metapodatke i pokrivenost sadržaja (tj. područje sfere koje je prekriveno sadržajem). Zadana orijentacija gledanja za početak prikazivanja svesmjernog videozapisa nalazi se duž osi X globalnih koordinatnih osi, a autori sadržaja mogu promijeniti zadano ponašanje.

OMAF specificira metapodatke o rangiranju prema kvaliteti regija (engl. *Region-wise quality ranking*, kratica: RWQR) kao osnovni mehanizam za omogućavanje odabira sadržaja ovisnog o korisnikovom vidnom okviru). Metapodaci o rangiranju kvalitete mogu se pružiti za regije sfere i za pravokutne regije na dekodiranim 2D slikama. Svaka regija dobiva vrijednost koja označava i opisuje relativni redoslijed kvalitete tih regija. Na primjer, kada regija A ima vrijednost različitu od nule, ali manju od vrijednosti regije B, to znači da regija A ima veću kvalitetu od regije B. Vrijednosti rangiranja kvalitete ne moraju predstavljati apsolutnu ljestvicu kvalitete, a vrijednost 0 ukazuje na nedefiniranu vrijednost rangiranja kvalitete.

Kao što je već prije rečeno, DASH specificira format datoteke opis medijske prezentacije (kratica: MPD) koji služi za opisivanje sadržaja dostupnog za strujanje i format segmenata cjelokupnog videa koji će se strujati. U formatu segmenta za ISOBMFF, svaki segment se sastoji od jednog ili više samostalnih video fragmenata. MPD je specificiran kao XML (engl. *Extensible Markup Language*) shema i sadrži jedan ili više adaptacijskih skupova, od kojih svaki sadrži jednu ili više reprezentacija. Reprezentacija odgovara ISOBMFF zapisu , a adaptacijski set sadrži prikaze istog sadržaja, ali različite kvalitete i koje onda *player* može odabrati na temelju dostupne brzine prijenosa. Svaki segment može se podijeliti na pod-segmente, koji su indeksirani unutar samog segmenta. Ovaj dizajn omogućuje klijentima da prvo dohvate segment, a zatim zaključe brzinu prijenosa i tako dohvate pod-segmente. OMAF specificira MPD proširenja koja pomažu pri odabiru svesmjernog sadržaja. **Tablica 4.2** sažima deskriptore koji se mogu koristiti za karakterizaciju svesmjernog video sadržaja.

Tablica 4.2 MPD deskriptori za karakterizaciju svesmjernog videa

| Naziv deskriptora | Opis |
|--|---|
| Element za pakiranje okvira (engl. <i>framepacking element</i>) | Stereoskopski format pakiranja okvira |
| PF | Označava projicirani svesmjerni videozapis i korišteni/e format/e projekcije |
| RWPK | Označava je li primijenjen RWP |
| CC | Pokrivenost sadržaja |
| SRQR | RWQR informacije za sferne regije |
| 2DQR | RWQR informacije za pravokutne regije na dekodiranim slikama |
| FOMV | Označava svesmjerni video ribljeg oka. Uz to se mogu naznačiti i uobičajena podešavanja s dvije leće (monoskopska od 360 stupnjeva, stereoskopska od 180 stupnjeva) |

4.4. Profili OMAF videa

OMAF podržava kreiranje i strujanje sadržaja ovisnog o korisnikovom vidnom okviru, pri čemu se mogu koristiti postojeće implementacije kodiranja i reprodukcije za 2D video uz dodatne modifikacije. Spomenute implementacije imaju sljedeće nedostatke. Brzine prijenosa potrebne za visokokvalitetni 360-video koji je neovisan o korisnikovom pogledu često su veće od propusnosti mreže u mnogim pristupnim mrežama. A kapacitet dekodiranja mnogih današnjih uređaja ograničen je na približno 4K dekodiranja u stvarnom vremenu (npr. 4096x2048 razlučivost), dok visokokvalitetni video sadržaj od 360 stupnjeva treba imati veću rezoluciju i mnogo zaslona na glavi je prikladno za prikaz vidnog okvira većeg od 4K rezolucije.

OMAF nudi alate i medijske profile za strujanje ovisno o korisnikovom vidnom okviru [71] kako bi se prevladali gore opisani nedostaci. U strujanju ovisnom o korisnikovom vidnom okviru, vidni okvir, tj. ono što korisnik vidi je prikazano u većoj kvaliteti nego preostala područja 360-videa. **Tablica 4.3** prikazuje profile OMAF videa. Potrebna razina video kodeka približno odgovara kapacitetu dekodiranja navedenom u tablici. Vrste shema navedene u tablici navode koji su formati projekcije dopušteni i koja se ograničenja, ako ih ima, primjenjuju za RWP. Vrste shema također podrazumijevaju koji metapodaci u formatu datoteka moraju i mogu biti prisutni. OMAF video profili također zahtijevaju da ekvivalentni metapodaci budu prisutni kao dodatne informacije u poboljšanju u brzinama prijenosa videa.

Tablica 4.3 Pregled profila OMAF videa

| Profil OMAF videa | Kodek | Dubina bita | Kapacitet dekodiranja | Vrsta sheme |
|---|-------|-------------|-----------------------|---------------|
| Profil temeljen na formatu HEVC neovisan o promjeni vidnog okvira | HEVC | ≤ 10 bita | 4K @ 60 Hz | erpv |
| Profil temeljen na formatu HEVC ovisan o promjeni vidnog okvira | HEVC | ≤ 10 bita | 4K @ 60 Hz | erpv, ercm |
| Profil temeljen na formatu AVC ovisan o promjeni vidnog okvira | AVC | 8 bita | 4K @ 30 Hz | erpv, ercm |

Metode [71][72] za postizanje strujanja ovisnog vidnom okviru mogu se svrstati u sljedeće pristupe:

Strujanje videa od 360 stupnjeva ovisnog o vidnom okviru

Nekoliko različitih verzija sadržaja od 360 stupnjeva je kodirano za različite orijentacije vidnog okvira i različite brzine prijenosa ili kvalitete. *Player* bira tok od 360 stupnjeva koji je u trenutnom vidnom okviru u većoj kvaliteti s time da odgovara mrežnoj propusnosti. Područja koja nisu u trenutnom vidnom okviru strujaju se u nižoj kvaliteti.

Strujanje ovisno o vidnom okviru temeljeno na podjeli na pločice

Projicirane slike su podijeljene u pločice koje su kodirane u setove pločice osjetljive na pokrete korisnika (engl. *motion-constrained tile sets*, kratica: MCTS). Nekoliko inačica sadržaja kodirano je na različitim brzinama prijenosa pomoću iste MCTS tehnologije te se svaki slijed pohranjuje u HEVC formatu i moguće ga je strujati. *Player* na osnovi MCTS-a odabire koju će brzinu prijenosa ili kvalitetu primiti.

Tablica 4.4 opisuje pristupe za strujanje ovisno o vidnom okviru temeljeno na prostornoj podjeli na pločice.

Tablica 4.4 Pristupi za strujanje ovisno o vidnom okviru temeljeno na prostornoj podjeli na pločice

| Pristup | Opis | Podrška | Primjer za OMAF |
|--|--|---------|-------------------|
| RWMQ (engl. <i>Region-wise mixed quality</i>) | Kodirano je nekoliko verzija s istim omjerom pločica i različitom brzinom prijenosa, tj. kvalitetom slike. <i>Player</i> bira visokokvalitetni MCTS za vidni okvir | Da | OMAF Annex D [71] |
| Vidni okvir + 360-video | Dohvaćaju se kompletne svesmjerne slike niske razlučivosti/kvalitete i MCTS koji pokriva prikazni okvir | Da | OMAF Annex D |
| RWMR (engl. <i>Region-wise mixed resolutions</i>) | Pločice su kodirane u više razlučivosti. <i>Player</i> odabire kombinaciju pločica visoke razlučivosti koje pokrivaju vidni okvir i pločice niske razlučivosti za preostala područja | Da | OMAF Annex D [72] |

5. Implementacija prilagodljivog strujanja uz prostornu podjelu na pločice

U ovom poglavlju bit će ponuđeno nekoliko rješenja za ostvarivanje prilagodljivog strujanja temeljenog na prostornoj podjeli na pločice. Korištena su dva besplatno dostupna alata, tj. pomagala za implementaciju tih rješenja. Svaki će biti opisan i pokazat će se koje su razlike među njima. Cilj ovog rada je implementirati i evaluirati laboratorijski prototip kojim se ostvaruje spomenuto strujanje i koje ga prikazuje putem zaslona pametnog telefona. U potpoglavlju „Laboratorijski prototip“ bit će opisano rješenje koje je koristilo Nokiatech alate. To rješenje uvodi OMAF tehnologiju i kodira video sadržaj tim standardom te ga je moguće pokrenuti na Android operativnom sustavu te prikazati strujanje na zaslonu pametnog telefona. Kodirani su videozapisi u OMAF standardu i napravljena je aplikacija za prilagodljivo strujanje uz prostornu podjelu na pločice te će sljedeća poglavlja reći nešto više o arhitektura sustava i objasniti će se način kodiranja videa u OMAF standardu te njegova podjela na pločice i prikaz putem pametnog telefona.

5.1. GPAC

GPAC je višemedijski okvir otvorenog koda, koji je usmjeren na modularnost i usklađenost sa standardima [73]. GPAC nudi alate za obradu, pregled, pakiranje, reprodukciju strujanja i interakciju s medijskim sadržajem. Takav sadržaj može biti bilo koja kombinacija zvuka, videozapisa, titlova, metapodataka, itd. GPAC je najpoznatiji po širokim mogućnostima s radom s MP4 formatom. Obilježava ga korištenje MPEG-4 standarda [73] te pruža alate za reprodukciju raznih medija, vektorske grafike i 3D renderiranja, stvaranje i daljnju distribuciju MPEG-4 standarda. Namijenjen je širokoj publici u rasponu od krajnjih korisnika koji s programerskim vještinama žele eksperimentirati s novim standardima za interaktivne tehnologije, do programera koji trebaju *playere*/poslužitelje za izradu aplikacija za strujanje višemedijskog sadržaja.

GPAC je rješenje za više platformi, napisano u cijelosti u programskom jeziku C. Trenutno je podržan u skoro svim najpoznatijim operacijskim sustavima kao što su Windows, Linux, Solaris, iOS i Android. Zanimljiv je što pruža strujanje HEVC videozapisa od 360 stupnjeva podijeljenog na pločice u DASH formatu. U ovom radu koristi se GPAC instalacija za platformu Windows [74].

GPAC nudi tri skupa alata temeljenih na biblioteci libgpac:

- Višemedijski alat za pakiranje sadržaja (engl. *packager*): **MP4Box**,
- Višemedijski *player*: **MP4Client** baziran na naredbenom retku (engl. *command line*) koji radi na višestrukim platformama ili grafičko korisničko sučelje (engl. *Graphical User Interface*, kratica: GUI) **Osmo4**,
- Alati poslužitelja za multipleksiranje i strujanje medijskog sadržaja.

Višemedijski alat za pakiranje podataka dostupan u GPAC-u naziva se MP4Box. Može se koristiti za obavljanje različitih manipulacija nad višemedijskim datotekama poput AVI, MPG, MPEG-2 TS, ali uglavnom na ISO medijskim datotekama (npr. datotekama formata MP4 i 3GP). MP4Box pruža mnogo funkcionalnosti, a samo će se navesti najvažnije funkcionalnosti, korištene u ovom radu. Alat nudi manipulaciju nad ISO datotekama kao što je MP4 te omogućuje dodavanje, uklanjanje i multipleksiranje zvuka, videa i prezentacijskih podataka (uključujući titlove) iz različitih izvora i u različitim formatima. Također omogućava podjelu 360-videa na pločice i pripremu takvog sadržaja za prilagodljivo strujanje putem protokola HTTP. *Player* podržava videozapis od 360 stupnjeva zasad samo u pravokutnoj projekciji.

GPAC također nudi *player* za višemedijski sadržaj zvan MP4Client koji je baziran na naredbenom retku ili Osmo4 ako je baziran kao grafičko sučelje. Ovaj *player* je mnogo više od tradicionalnog audiovizualnog *playera*, zato što se osim mogućnosti reprodukcije većine video ili audio formata i podrške za većinu postojećih protokola, usredotočuje na grafiku, animacije i interaktivnost.

Postupak kodiranja videa u pločice i prikaz putem MP4Client-a objašnjen je u sljedećem dijelu i nabrojan u redosljednim koracima.

1. Na početku je s [75] preuzet video od 360 stupnjeva, rezolucije 4096x2048 i s 30 sličica po sekundi, koji se zove *London on Tower Bridge*. Prvi korak se sastoji od ekstrakcije sirovih sličica u datoteku YUV formata koju će Kvazaar dalje koristiti. To se postiže korištenjem besplatno dostupnog kodera FFmpeg upisivanjem naredbe:

```
ffmpeg -i input.mp4 output.yuv
```

Gdje je *input* izvorni 360-video, a *output* kodirana datoteka u YUV formatu. Zanimljivo je što video koji je u .mp4 formatu imao 60 MB, sada ima 10.5 GB.

2. Međutim, dobiveni VR 360 videozapis nije podijeljen na pločice, pa ga je potrebno ponovno kodirati. Stoga, pomoću kodera Kvazaar dobiven video u .yuv format kodiramo u HEVC format te ga podijelimo na pločice:

```
kvazaar -i output.yuv --input-res 4096x2048 -o output.hvc --  
tiles 8x3 --slices tiles --mv-constraint frametilemargin --  
bitrate 1000000 --period 30  
--input-fps 30
```

U samom primjeru korišteno je više zastavica, a njihovo značenje je:

- *-i* označava ulazni video
- *--input-res* označava razlučivost izvornog videa (u ovom slučaju 4096x2048)
- *-o* označava izlazni video
- *--tiles m×n* govori na koliko pločica će se video podijeliti (m stupci, n su retci)
- *--slices tiles* kontrolira što će se kapsulirati u zaseban dio (*tiles* – svaka pločica će se kapsulirati u zaseban dio)
- *--mv-constraint frametilemargin* ograničuje vektore kretanja (*frametilemargin* – ograničuje ih unutar pločice pa čak i više)
- *--bitrate* određuje brzinu prijenosa (u ovom slučaju 1 Mbps)
- *--period* određuje period I-okvira (u ovom primjeru I-okvir će se pojaviti svaku 30. sličicu videozapisa)
- *--input-fps* označava brzinu osvježavanja okvira ulaznog videa

Ciljana brzina prijenosa se može postaviti preko *bitrate* zastavice i preko *qp* (engl. *quantization parameter*) zastavice, koja služi kao regulator brzine prijenosa. Prilikom kodiranja pločica u više razina kvaliteta, što je u ovom radu i napravljeno, potrebno je zadržati sve postavke kodiranja kao što je gore navedeno, jedino je potrebno mijenjati *bitrate* vrijednost, jer bi promjena *qp*-a dovela do nekompatibilnih strujanja.

3. Izlazni HEVC video je sirovi tok podataka i potrebno ga je zapakirati u spremnik. Kako bi se to postiglo, potrebno je koristiti alat MP4Box na sljedeći način:

```
MP4Box -add output.hvc:split_tiles -fps 30 -new
video_tiled.mp4
```

Važno je da brzina osvježavanja okvira bude ista kao što je bila prilikom kodiranja videa.

4. Tada smo dobili MP4 datoteku te možemo generirati DASH segmente i datoteku za opis medijske prezentacije za zapakirani video na sljedeći način:

```
MP4Box -dash 1000 -rap -frag-rap -profile live -out  
dash_tiled.mpd video_tiled.mp4
```

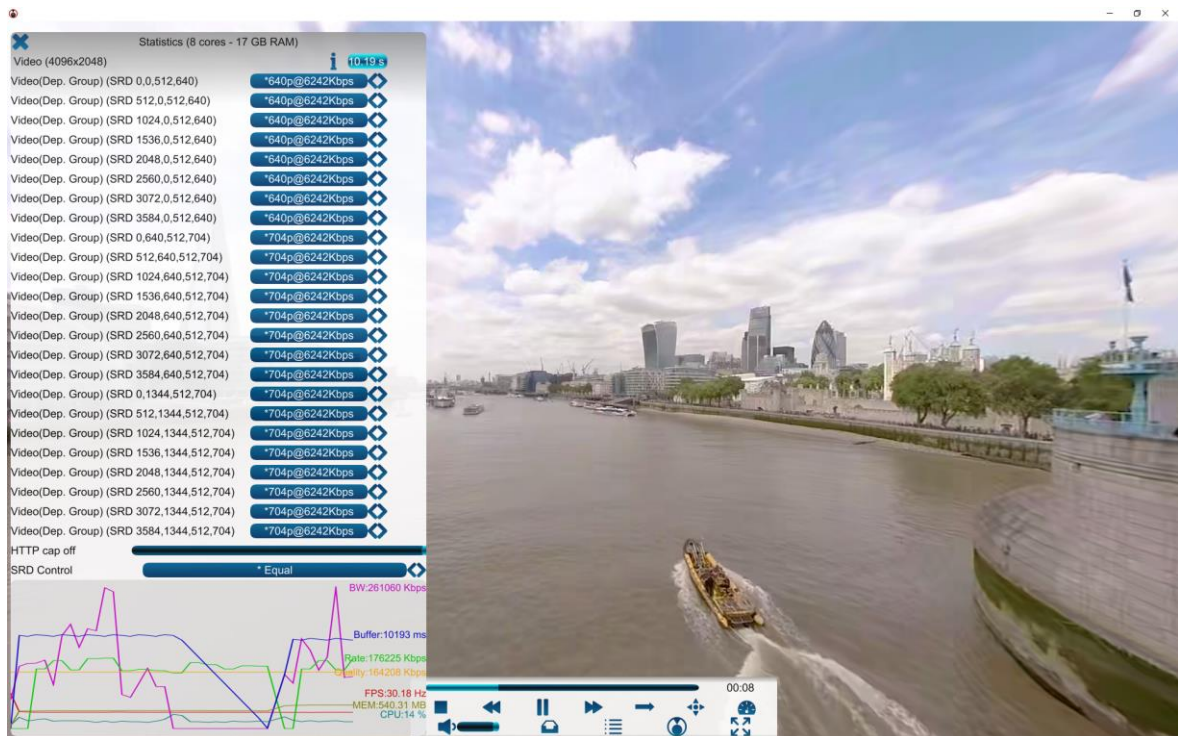
MP4Box također može generirati MPD s više razina kvaliteta dodavanjem više ulaznih datoteka (*rep1* i *rep2*) koje imaju različiti brzine prijenosa.

```
MP4Box -dash 1000 -rap -frag-rap -profile live -out  
dash_tiled.mpd rep1.mp4 rep2.mp4
```

- Zastavica *-dash* stvara DASH segmente iz ulaznih datoteka i određena je trajanjem u milisekundama,
 - *-rap* osigurava da segmenti počnu s nasumičnim pristupnim točkama,
 - *-frag-rap* osigurava da svi fragmenti počnu s nasumičnim pristupnim točkama,
 - *-profile* određuje ciljni profil DASH-a i postavlja zadane opcije kako bi se osigurala sukladnost željenom profilu (*live* – dinamičan mod),
 - *-out* određuje naziv izlazne MPD datoteke.
5. Jednom kada se sadržaj generira i pohrani na strani poslužitelja, može se strujati pomoću MP4Client-a na klijentskoj strani. Jedini potrebni argument je URL MPD datoteke . U slučaju videozapisa od 360 stupnjeva, moguće je prikazati samo korisnički vidni okvir pomoću prefiksa *#360LIVE* odmah nakon URL-a. Kao primjer su dane dvije naredbe, prva koja prikazuje video u normalnom načinu za ravan zaslon i druga, koja reproducira video na ranije spomenuti način za 360-video:

```
MP4Client -gui http://server:port/test.mpd  
MP4Client -gui http://server:port/test.mpd#360LIVE
```

Upisivanjem zastavice *-gui* (*graphical user interface*) *Player* MP4Client se pokreće uz grafički prikaz koji nudi statistiku na kojoj se nalaze kodirane pločice i njihove razlučivosti, kontrola brzine prijenosa i načina SRD kontrole, a na dnu, kao što prikazuje **Slika 5.1**, dan je graf raznih uvjeta u mreži. Na grafu se nalaze: širina prijenosnog pojasa, veličina spremnika, brzina prijenosa videa, kvaliteta videa, brzina osvježavanja okvira te iskorištenost radne memorije i procesora računala.



Slika 5.1 Reprodukcijska 360-vidica *London on Tower Bridge* u MP4Client-u

5.2. Nokiotech Omaf

OMAF je standard koji je razvila grupa MPEG za omogućavanje standardiziranih aplikacija za svesmjerne medija s fokusom na 360-video. *Creator/Player Engine* je implementacija OMAF standarda koju je razvila tvrtka Nokia [76] kako bi se demonstrirale njegove moćne značajke i kako bi se pomoglo u postizanju interoperabilnosti između implementacija OMAF-a.

Nokia OMAF *Creator/Player* aplikacija implementira OMAF standard verziju 1 i uključuje sljedeće značajke:

1. Pretvaranje pravokutno projiciranog HEVC videa od 360 stupnjeva pohranjenog u konvencionalnu MP4 datoteku u OMAF HEVC datoteke ili DASH strujanje neovisno o promjeni vidnog okvira.
2. Kreiranje OMAF HEVC DASH strujanja od 360 stupnjeva koje je ovisno o promjeni vidnog okvira od HEVC datoteka koje su podijeljene na pločice.
3. Kreiranje OMAF slika od 360 stupnjeva, od HEVC datoteka.
4. Reprodukcijska OMAF HEVC datoteka ili DASH strujanja ovisnih i neovisnih o promjeni vidnog okvira na Windowsima i na Androidu.
5. Reprodukcijska OMAF slika na Windowsima i na Androidu.

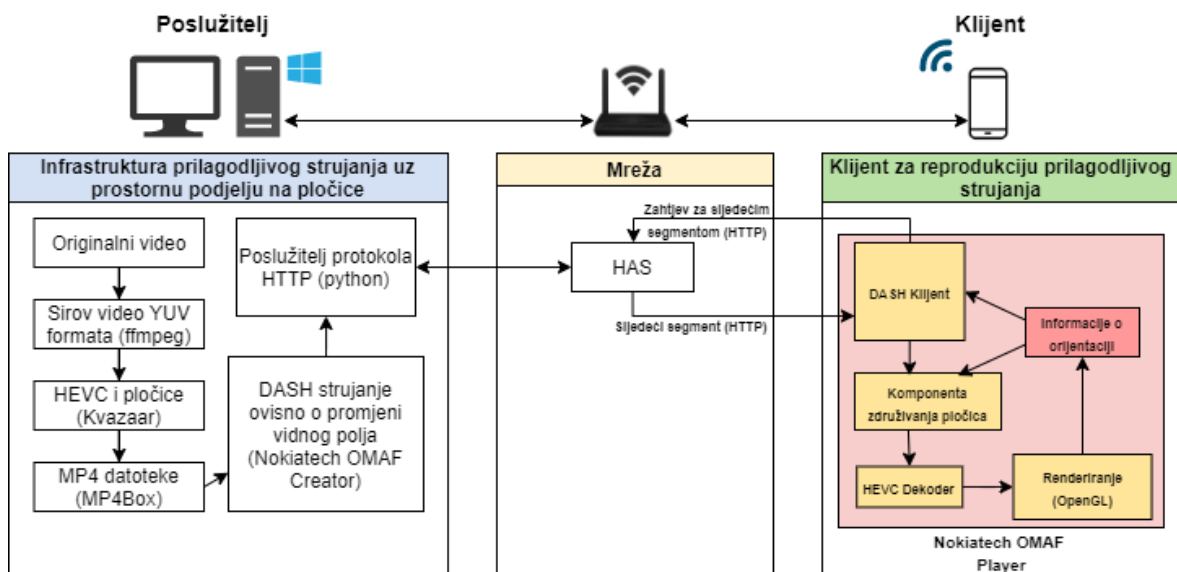
Sve se komponente kompajliraju i pokreću na Windows 10 operacijskom sustavu. OMAF *Player* u praksi, na Windowsima zahtijeva hardverski HEVC dekodirer, koji ubrzava dekodiranje, a dostupan je na grafičkim karticama Nvidia 10xx. OMAF *Player* je također moguće pokrenuti na Androidu, dok je *Creator* podržan na Windows 10 i Linux operacijskim sustavima.

5.3. Laboratorijski prototip

U ovom poglavlju bit će opisano rješenje za prilagodljivo strujanje 360-videa uz prostornu podjelu na pločice. U poglavljima ispod dan je pregled arhitekture sustava, redosljed kodiranja videa, njegova podjela na pločice i prilagodljivo strujanje. Na kraju će biti naveden primjer te aplikacije na VR platformi.

5.3.1. Arhitektura sustava

Kao dokaz koncepta (engl. *Proof of Concept*, kratica: POC), implementiran je laboratorijski prototip. **Slika 5.2** prikazuje arhitekturu laboratorijskog prototipa, tj. prototipa za prilagodljivo strujanje uz prostornu podjelu na pločice i njegov prikaz na Android pametnom uređaju. Arhitektura se sastoji od **poslužitelja, komunikacijske mreže i klijenta**.



Slika 5.2 Arhitektura sustava prilagodljivog strujanja uz prostornu podjelu na pločice

VR poslužitelj je zapravo računalo na kojem se kodiraju videozapisi i pripremaju za prilagodljivo strujanje kroz komunikacijsku mrežu. Prilikom izrade laboratorijskog prototipa koristilo se osobno računalo HP Omen 17 (Intel Core i7 7700HQ 2.8GHz osmo-jezgreni procesor, 16GB DDR3 RAM memorije, NVIDIA GeForce GTX 1060 grafička kartica od 6GB VRAM-a).

Na poslužitelju se nalazi originalni videozapis preuzet s [75] te se on pomoću ffmpeg-a kodira u sirov video YUV formata. Tada se YUV datoteka kodira HEVC standardom i dijeli na pločice te se pomoću alata MP4Box pretvara u MP4 format koji će kasnije aplikacije Nokiatech OMAF *Creator* koristiti za izradu DASH segmenata. Na kraju se na računalu pokreće Python poslužitelj protokola HTTP [76] koji kodirane DASH segmente priprema za strujanje preko mreže.

Sljedeća u arhitekturi je komunikacijske mreža, koja koristi prilagodljivo strujanje putem protokola HTTP za prijenos podataka između poslužitelja i klijenta. Zadnji je prilagodljivi klijent za prikaz videozapisa preko zaslona pametnog telefona. U ovom radu koristio se Huawei P30 pametni telefon. Za reprodukciju prilagodljivog strujanja uz prostornu podjelu na pločice zaslužan je Nokiatech-ov OMAF *Player* koji je Android aplikacija te funkcionira na način da na klijentskoj strani prati korisnikov vidni okvir. Informacije o trenutnoj orijentaciji korisnika prosljeđuju se DASH klijentu i komponenti združivanja pločica. DASH klijent tada pokreće postupak odabira pločica koristeći te informacije o orijentaciji i odlučuje koje pločice treba zahtijevati pri visokoj, a koje pri niskoj rezoluciji. Nakon preuzimanja segmenata, komponenta združivanja stvara jedno pojedinačno korisničko strujanje bitova ovisno o vidnom okviru. Potreban je HEVC softverski dekodirer koji obrađuje strujanje bitova i prosljeđuje dekodirane slike u OpenGL renderer kako bi se stvorio korisnički prikaz koji će se izvoditi na Android uređaju.

Opis rada je sljedeći, kodirani segmenti nalaze se na poslužitelju i dostupni su putem python poslužitelja protokola HTTP. Klijentska aplikacija preko URL-a dohvati MPD datoteku te kreće sa strujanjem. Klijent cijelo vrijeme prati vidni okvir, te u korisnikovom vidnom okviru (ako su zadovoljeni uvjeti mreže) prikazuje pločice videozapisa više rezolucije, a izvan vidnog okvira pločice niže kvalitete i razlučivosti.

5.3.2. Kodiranje videa

U ovom poglavlju bit će navedeni koraci potrebni za stvaranje potrebnih datoteka za prilagodljivo strujanje videozapisa od 360 stupnjeva ovisno o promjeni vidnog okvira, uz korištenje pravokutne projekcije (ERP). Ukratko, postupak je sljedeći: Priprema YUV-a, kodiranje te pisanje JSON datoteke.

- Najprije je preuzet video s web stranice [75] koji se naziva *London on Tower Bridge*. Preuzet je u ultra visokoj kvaliteti (engl. *ultra high definition*, kratica: UHD). **Tablica 5.1** prikazuje detaljne podatke preuzetog videa.

Tablica 5.1 Detalji preuzetog videa

| Naziv videa | <i>London on Tower Bridge</i> |
|----------------------------|-------------------------------|
| Format | MP4 |
| Trajanje | 29 sekundi |
| Razlučivost | 4096x2048 piksela |
| Veličina | 70 MB |
| Brzina osvježavanja okvira | 30 FPS-a |
| Brzina prijenosa videa | 19.6 Mbps |

Za izradu OMAF DASH strujanja ovisnog o promjeni vidnog okvira (engl. *Viewport Dependent DASH stream*) potreban je kodirani HEVC videozapis podijeljen na pločice. Za HEVC (H.265) kodiranje korišten je Kvazaar zbog toga što on pruža opciju podjele 360-videa na pločice. Kvazaar zahtijeva zakrpe uz njihovo najnovije izdanje, pa se preporučuje preuzimanje najnovije verzije izvornog koda i njegova samostalna instalacija. Također, Kvazaar zahtijeva YUV format kao izvorni video koji će se kodirati stoga je potrebno ranije spomenuti *London on Tower Bridge* koji je u MP4 formatu pretvoriti u YUV pomoću alata FFmpeg. To je napravljeno istom naredbom kao i u GPAC rješenju.

- Za kodiranje YUV datoteke HEVC standardom korišten je koder Kvazaar. Kao i u GPAC rješenju, korištena je slična naredba uz male preinake:

```
kvazaar -i london.yuv --input-res 4096x2048 -o output.hvc
--tiles 8x3 -slices tiles --mv-constraint frametilemargin
--input-fps 30 --gop 8 --no-open-gop --bipred -p 32
--set-qp-in-cu -bitrate 10000000
```

Značenje dodatnih zastavica je:

- `--gop 8 --no-open-gop --bipred` omogućuje uporabu piramide B-okvira s IDR okvirima
- `--bipred` omogućuje uporabu B okvira i dvosmjerno predviđanje
- `-p 32` označava period I-okvira (svaku 32. sličicu)

U kodiranju videozapisa, GOP (engl. *Group of pictures*) struktura određuje redoslijed u kojem su raspoređeni I-okviri i P ili B okviri. GOP je zbirka uspješno kodiranih slika u strujanju videa. Svako kodirano strujanje sastoji se od uzastopnih GOP-ova iz kojih se generiraju vidljivi okviri. Ulazak u novi GOP znači da dekoderu ne trebaju prethodni okviri za dekodiranje sljedećih, što omogućuje brzo pretraživanje videozapisa.

IDR (engl. *Instantaneous Decoder Refresh*) okvir je posebna vrsta I-okvira te određuje da se niti jedan okvir nakon IDR okvira ne može odnositi na okvire prije njega. Ovo olakšava traženje videozapisa i bržu reakciju *playera*. IDR okviri uvode se kako bi se izbjegla izobličenja u videozapisu kada se želi preskočiti na neko mjesto u videozapisu ili započeti gledanje u sredini videozapisa.

Slika 5.3 prikazuje tijek kodiranja Kvazaara nakon upisivanja prethodne naredbe i ispis podataka u Windows PowerShellu naredbenom retku

```

Video size: 4096x2048 (input=4096x2048)
POC 0 QP 28 AVG QP 28.0 (I-frame) 7615432 bits PSNR Y 37.8603 U 42.8981 V 47.0602
POC 8 QP 28 AVG QP 28.0 (P-frame) 3588464 bits PSNR Y 37.6520 U 42.5956 V 46.8037
POC 4 QP 36 AVG QP 30.7 (B-frame) 807480 bits PSNR Y 33.5986 U 41.3519 V 46.1273
POC 2 QP 42 AVG QP 33.5 (B-frame) 250696 bits PSNR Y 32.0596 U 41.6441 V 46.4037
POC 1 QP 49 AVG QP 36.6 (B-frame) 32056 bits PSNR Y 29.9581 U 41.9993 V 46.5511
POC 3 QP 50 AVG QP 38.8 (B-frame) 23192 bits PSNR Y 29.6051 U 41.4089 V 46.1977
POC 6 QP 43 AVG QP 39.4 (B-frame) 191344 bits PSNR Y 32.2569 U 41.8291 V 46.4727
POC 5 QP 45 AVG QP 40.1 (B-frame) 77312 bits PSNR Y 31.2143 U 41.5844 V 46.3219
POC 7 QP 47 AVG QP 40.9 (B-frame) 47368 bits PSNR Y 30.9690 U 42.1223 V 46.5469
POC 16 QP 39 AVG QP 40.7 (P-frame) 643496 bits PSNR Y 31.0619 U 39.4297 V 45.0196
POC 12 QP 44 AVG QP 41.0 (B-frame) 157224 bits PSNR Y 28.9885 U 39.6611 V 45.2453
POC 10 QP 47 AVG QP 41.5 (B-frame) 60560 bits PSNR Y 29.8638 U 41.2569 V 46.1657
POC 9 QP 51 AVG QP 42.2 (B-frame) 14672 bits PSNR Y 29.4047 U 41.8169 V 46.4011
POC 11 QP 51 AVG QP 42.9 (B-frame) 14280 bits PSNR Y 28.4665 U 40.5873 V 45.8430
POC 14 QP 48 AVG QP 43.2 (B-frame) 44352 bits PSNR Y 28.7316 U 39.7201 V 45.2319
POC 13 QP 51 AVG QP 43.7 (B-frame) 11616 bits PSNR Y 28.1455 U 39.8505 V 45.3084
POC 15 QP 51 AVG QP 44.1 (B-frame) 11704 bits PSNR Y 28.0774 U 39.6166 V 45.2095
POC 24 QP 34 AVG QP 43.6 (P-frame) 1249392 bits PSNR Y 32.5181 U 39.9924 V 44.9092
POC 20 QP 47 AVG QP 43.7 (B-frame) 33816 bits PSNR Y 27.5767 U 39.1069 V 44.8107
POC 18 QP 48 AVG QP 44.0 (B-frame) 17384 bits PSNR Y 27.7892 U 39.4773 V 45.0994
POC 17 QP 51 AVG QP 44.3 (B-frame) 15448 bits PSNR Y 27.9382 U 39.5101 V 45.1135
POC 19 QP 51 AVG QP 44.6 (B-frame) 17520 bits PSNR Y 27.9823 U 39.4089 V 45.0017
POC 22 QP 43 AVG QP 44.5 (B-frame) 73240 bits PSNR Y 29.2945 U 39.8549 V 44.9299
POC 21 QP 51 AVG QP 44.8 (B-frame) 11008 bits PSNR Y 27.9242 U 39.6813 V 44.9816
POC 23 QP 50 AVG QP 45.0 (B-frame) 14536 bits PSNR Y 28.7642 U 39.9231 V 44.8414
POC 32 QP 31 AVG QP 44.5 (P-frame) 1890248 bits PSNR Y 33.9490 U 40.5879 V 45.2443
POC 28 QP 42 AVG QP 44.4 (B-frame) 122472 bits PSNR Y 28.9623 U 39.2387 V 44.6515
POC 26 QP 41 AVG QP 44.3 (B-frame) 141840 bits PSNR Y 30.1197 U 39.9245 V 44.9632
POC 25 QP 51 AVG QP 44.5 (B-frame) 14912 bits PSNR Y 29.0899 U 40.1078 V 44.9936
POC 27 QP 51 AVG QP 44.7 (B-frame) 20744 bits PSNR Y 28.2926 U 39.5615 V 44.8513
POC 30 QP 43 AVG QP 44.6 (B-frame) 222008 bits PSNR Y 30.6709 U 40.0802 V 45.0670
POC 29 QP 49 AVG QP 44.8 (B-frame) 12616 bits PSNR Y 28.3770 U 39.7579 V 44.9022
POC 31 QP 50 AVG QP 45.0 (B-frame) 14368 bits PSNR Y 28.7978 U 40.2348 V 45.0589
POC 0 QP 34 AVG QP 44.7 (I-frame) 3553208 bits PSNR Y 31.4737 U 37.7581 V 42.9289

```

Slika 5.3 Tijek Kvazaar kodiranja i ispis u Windows PowerShellu

Na slici je redosljedno prikazan POC (engl. *Picture order count*) koji određuje redosljed dekodiranih okvira, počevši od IDR okvira (POC = 0). Tada je ispisan parametar kvantiziranja za svaki okvir i dosadašnji prosječni QP. Nakon toga označen je dekodirani okvir (I-okvir, P-okvir, B-okvir) te njegova brzina prijenosa, te je na kraju prikazan PSNR pojedinog okvira i njegove 3 vrijednosti za YUV format.

- Izlani tok podataka u Kvazaaru je HEVC/H.265 formata. Najlakši način da se H.265 pretvori u MP4 je korištenjem MP4Box-a:

```
MP4Box -add output.hvc -new output.mp4 -fps 30
```

Jedina razlika s GPAC rješenjem je ta što se ne koristi `:split_tiles`, zbog toga što će se u Nokiotech rješenju pločice programski podijeliti korištenjem konfiguracijske JSON datoteke.

Sljedeće je potrebno ispravno urediti konfiguracijsku JSON datoteku dostupnu s [77]. Datoteka sadrži podatke o putanji do kodiranih tokova. Podjela na pločice unaprijed je konfigurirana pomoću Nokia *omafvd* alata i automatski se primjenjuje na videozapise. **Slika 5.4** prikazuje primjer konfiguracijske JSON datoteke korištene prilikom implementacije laboratorijskog prototipa za strujanje jednake rezolucije na više kvaliteta (OMAF ANNEX D.4.2).

```
{
  // "video" is must
  "video": {
    "common": {
      "projection": "equiangular", // or cubemap; currently only the default OMAF cubemap format is supported
      "output_mode": "MultiQ" // Mode of VD: MultiQ == single resolution, multiple bitrates
    },
    "low": {
      "filename": "london_low_res.mp4",
      "quality": 100 // 1...255 where 1 is the best; if not given, defaults to 1
    },
    "med": {
      "filename": "london_med_res.mp4",
      "quality": 50 // 1...255 where 1 is the best; if not given, defaults to 1
    },
    "high": {
      "filename": "london_high_res.mp4",
      "quality": 1 // 1...255 where 1 is the best; if not given, defaults to 1
    }
    // add as many as needed
  },
  "dash": {
    "output_name_base": "test", // Basename for the DASH output files
    "mpd": {
      "filename": "$Name$.mpd" // optional, default is $Name$.mpd. $Name$ expands to the value of "output_name_base" or if it is not given, to "output_mode"
    },
    "media": {
      "subsegments_per_segment": 1, // optional, defaults to 1. If more than 1, 1 GOP is mapped to a subsegment, otherwise 1 GOP is mapped to a segment
      "segment_name": {
        // $Name$ expands to the value of "output_name_base" or if it is not given, to "output_mode"
        // $Segment$ expands to MPD's "$Number$" or "init" depending on the segment type. Using MPD's $Number$ directly here is not allowed.
        "video": "$Name$.video.$Segment$.mp4",
        "audio": "$Name$.audio.$Segment$.mp4",
        "extractor": "$Name$.extractor.$Segment$.mp4"
      }
    }
  }
}
```

Slika 5.4 Primjer konfiguracijske JSON datoteke

Na početku datoteke zadaje se način projekcije videozapisa (u ovom slučaju korištena je pravokutna projekcija), a nakon toga mod izlaznih datoteka (*MultiQ* označava da se koristi jedna rezolucija, a više brzina prijenosa). Nakon početnih postavki slijedi unos dva ili više reprezentacija različitih brzina prijenosa, od kojih se svaka sastoji od imena dotične datoteke i pozitivnog cijelog broja kvalitete. Taj broj može biti između 1 i 255, gdje broj 1 označava najbolju kvalitetu, a broj 255 najgoru.

- Kada se ispravno napiše JSON datoteka, kreira se DASH strujanje ovisno o promjeni vidnog okvira naredbom:

```
omafvd.exe config.json
```

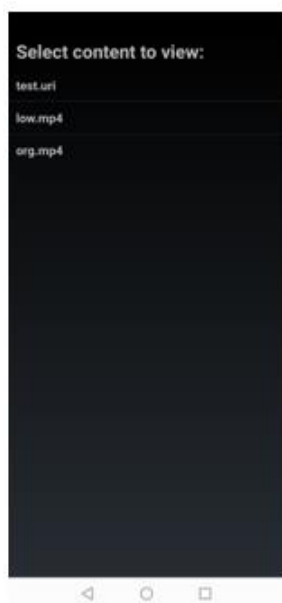
5.3.3. Prikaz putem zaslona pametnog telefona

U ovom rješenju dostupna je Android aplikacija koja može reproducirati OMAF prilagodljivo strujanje uz prostornu podjelu na pločice. Aplikacija pakira URI i MP4 datoteke i prikazuje njihov popis prilikom pokretanja. URI datoteke su tekstualne datoteke s .uri ekstenzijom korištene za specificiranje URL-a DASH strujanja, tj. lokaciju MPD datoteke.

Nakon generiranja datoteka spremnih za strujanje, potrebno je osposobiti poslužitelj. To je učinjeno korištenjem python poslužitelja protokola HTTP pokretanjem naredbe na računalu:

```
python -m http.server 8000
```

IP adresa računala je npr. 192.168.0.29, stoga je u URI datoteku potrebno upisati URL `http://192.168.0.29:8000/test.mpd` kojeg će pametni telefon otvoriti prilikom pokretanja. **Slika 5.5** prikazuju scene pokrenute aplikacije. U prvoj sceni korisnik može odabrati URI datoteku za reprodukciju strujanja, dok druga scena prikazuje strujanje odabranog videozapisa.



Slika 5.5 Scena za odabir datoteke za strujanje (lijevo); Scena prilagodljivog strujanja uz prostornu podjelu na pločice (desno)

6. Korištene programske tehnologije

Ovo poglavlje navodi i opisuje dodatno korištene programske alate i tehnologije bez kojih ne bi bilo moguće implementirati rješenja ovog rada.

6.1. Microsoft Visual Studio

Microsoft Visual Studio je integrirano razvojno okruženje (engl. *integrated development environment*, kratica: IDE) tvrtke Microsoft koje se koristi za izradu računalnih programa, web stranica i servisa te mobilnih i web aplikacija [78]. Visual Studio podržava više različitih programskih jezika pri čemu su neki od njih ugrađeni (npr. C, C++, C#, HTML, CSS), a za preostale (npr. Python, Ruby) je potrebno koristiti priključke (engl. *plug-in*). Visual Studio sadrži uređivač koda (engl. *code editor*) koji podržava isticanje sintakse (engl. *syntax highlighting*) i kompletiranje koda koristeći IntelliSense komponentu za kompletiranje koda. Visual Studio također sadrži program za pronalazak pogrešaka (engl. *debugger*) te vizualne dizajnere kao što su Windows Forms Designer, Class Designer, Data Designer i slično. Za izgrađivanje rješenja prilagodljivog strujanja i izmjenu skripti korišten je Microsoft Visual Studio Community 2017.

6.2. Android Studio

Android Studio službeno je integrirano razvojno okruženje (IDE) za Googleov operativni sustav Android, izgrađeno na JetBrainsovom IntelliJ IDEA softveru i dizajnirano posebno za razvoj Androida [79]. Dostupan je za preuzimanje na operacijskim sustavima Windows, macOS i Linux. To je zamjena za Eclipse Android Development Tools (ADT) kao primarni IDE za razvoj izvornih Android aplikacija. Android Studio najavljen je 16. svibnja 2013. na Google I / O konferenciji. 7. svibnja 2019. Kotlin je zamijenio Javu kao Googleov preferirani jezik za razvoj aplikacija za Android. Java je i dalje podržana, kao i C ++. Za implementaciju laboratorijskog prototipa za Android operativni sustav korišten je Android Studio 4.0.

6.3. Wireshark

Wireshark je vodeći svjetski analizator mrežnog protokola [80]. To je besplatan program otvorenog koda (engl. *open source*) i služi za analizu mrežnog prometa i rješavanje raznih problema mreže te za razvoj protokola. Također koristi se za edukativne svrhe. Omogućuje korisnicima da na „mikroskopskoj“ razini vide što se događa u mreži i standard je u mnogim komercijalnim i neprofitnim poduzećima, vladinim agencijama i obrazovnim institucijama. Razvoj Wiresharka uspijeva zahvaljujući dobrovoljnim doprinosima stručnjaka za umrežavanje širom svijeta i nastavak je projekta koji je Gerald Combs pokrenuo 1998. godine. Dostupan je za različite operacijske sustave kao što su GNU/Linux, OS X, BDS, Solaris i Microsoft Windows. Također, u programskim jezicima C i C++ razvijena je i verzija Wiresharka za terminal zvana TShark. Wireshark hvata mrežne pakete te nastoji prikazati i objasniti njihov sadržaj na najbolji, odnosno najefikasniji način. Wireshark ima bogat set značajki, a neke od njegovih mogućnosti su:

- Detaljan uvid u stotine protokola, uz stalno dodavanje novih
- Uživo hvatanje paketa i izvan mrežna analiza
- Hvatanje mrežnih paketa na odabranom mrežnom sučelju
- Podrška za više platformi
- Pretraživanje i filtriranje dohvaćenih mrežnih paketa po različitim kriterijima
- Kreiranje statistike mreže, koja se može spremirati u datoteke formata XML, PostScript, CSV ili u običnu tekstualnu datoteku

6.4. FFmpeg

FFmpeg [81] je besplatni softver koji se sastoji od brojnih programskih biblioteka i programa za rukovanje videom, zvukom i ostalim višemedijskim datotekama i sadržajima. FFmpeg je u svojoj srži program, dizajniran za obradu video i audio datoteka koji se temelji i koristi kao naredbeni redak. Široko je korišten za transkodiranje formata, osnovno uređivanje (obrezivanje i spajanje) videozapisa, promjene veličine videozapisa, umetanje efekata u postprodukciji videozapisa i usklađivanje standarda. FFmpeg uključuje libavcodec, biblioteku audio/video kodeka koju koriste mnogi komercijalni i besplatni softverski proizvodi, libavformat (Lavf), te jezgri ffmpeg program komandne linije za kodiranje višemedijskih datoteka. Naziv projekta inspiriran je MPEG video standardom, zajedno s "FF" za "fast forward". Logotip koristi cik-cak uzorak koji prikazuje kako MPEG video kodeci rukuju entropijskim kodiranjem FFmpeg je dio stotina drugih softvera i projekata, a njegove su biblioteke temeljni dio softverskih medijskih playera kao što je VLC, a uključen je u procesiranje YouTube i iTunes datoteka. FFmpeg sadrži kodeke za kodiranje i dekodiranje većine poznatih formata audio i video datoteka, što ga čini vrlo korisnim za transkodiranje uobičajenih i neuobičajenih medijskih datoteka u jedan zajednički format.

6.5. Kvazaar

Kvazaar [82][83] je akademski softver za kodiranje videa u novom HEVC/H.265 (engl. *High Efficiency Video Codnig*) standardu. Studentima i stručnjacima nudi besplatni, više-platformski HEVC-inkoder. Kvazaar se razvija u C-u i optimizira u Assembly-u pod licencom LGPLv2.1. Razvijen je od *Ultra Video Group* na tehnološkom sveučilištu Tampere (TUT), a rad na implementaciji provodi aktivna zajednica na GitHub-u. Trenutno Kvazaar uključuje sve bitne alate za kodiranje HEVC-a i može postići brzinu HEVC kodiranja u stvarnom vremenu do 4K videa. Također ga podržavaju ranije spomenuti *FFmpeg* te *Libav*.

7. Evaluacija i performanse

Ovo poglavlje opisuje provedenu evaluaciju i testiranje dokaza koncepta, tj. rješenja za prilagodljivo strujanje uz prostornu podjelu na pločice. Testiranje je provedeno u laboratoriju na Fakultetu elektrotehnike i računarstva u Zagrebu.

Odabrana su tri videozapisa koja pokazuju različite značajke. Sva tri videozapisa besplatno su preuzeta s [75]. Prvi je „*Crystal Shower Falls*“ (pogled na vodopad gdje se scena previše ne mijenja; originalno duljine 2 minute, ali zbog svoje ogromne veličine kada se kodira u YUV format, video je smanjen na 1 minutu), drugi je „*London on Tower Bridge*“ (pogled na London s mosta gdje se scena umjereno mijenja; duljine je 30 sekundi), a treći je „*Ayutthaya – Easy Tripod Paint*“ (leteći dron snima pogled na stari grad na Tajlandu gdje se scena često mijenja, duljine je 30 sekundi). Međutim, ovi originalni VR videozapisi od 360 stupnjeva nisu podijeljeni na pločice, stoga ih je potrebno pravilno kodirati. Također, preuzeti videozapisi prikazuju različite karakteristike u smislu prostorne i vremenske složenosti, što čini njihovu analizu komplementarnom.

Odjeljak sadrži detalje o sekvencama testnih videozapisa i ispitanim uzorcima. Također, objasnit će se kako je postavljen eksperiment i navest metodologija ispitivanja korištena za subjektivnu analizu. Nadalje, provest će se statistička analiza različitih videozapisa kodiranih na više načina i subjektivno će procijeniti njihova razlika. Na kraju će biti navedeni rezultati ovog ispitivanja uz isječak uhvaćenog prometa iz Wireshark programa.

7.1. Postavljanje eksperimenta

Za testiranje rada aplikacije, tj. prilagodljivog strujanja u raznim mrežnim uvjetima postavljen je laboratorijski eksperiment. Osobno računalo s pohranjenim videozapisima spremnim za prilagodljivo strujanje uz prostornu podjelu na pločice korišteno je kao poslužitelj na kojem je pokrenuta python skripta za instanciranje poslužitelja protokola HTTP. Poslužitelj i WIFI usmjeritelj su zbog tehničkih razloga spojeni LAN kabelom s uređajem Net.Storm od tvrtke Albedo. Net.Storm je FPGA (engl. *field-programmable gate array*) emulator koji nudi inženjerima da modeliraju i mijenjaju proizvoljnu dinamiku performansi, uključujući kašnjenje paketa, kolebanje kašnjenja u mreži (engl. *jitter*), ograničenja prijenosne brzine, zagušenja, gubitak paketa, pogreške i dupliciranje na živim paketima protokola IP [84]. Sav promet koji prolazi između računala na kojem se pokreće poslužitelj i WIFI usmjeritelja prolazi kroz taj uređaj i u ovom testiranju regulira se samo prijenosna brzina (bandwidth) između poslužitelja i usmjeritelja. Pametni uređaj Huawei P30 također je spojen na WIFI usmjeritelj i na njemu se pokreće aplikacija za prilagodljivo strujanje. Sav mrežni promet koji se odvija snima se pomoću programa Wireshark na osobnom računalu koje služi kao poslužitelj. **Slika 7.1** prikazuje postavljeni laboratorijski eksperiment.



Slika 7.1 Postavljanje laboratorijskog eksperimenta

Za ovaj eksperiment, zbog svojih moćnih značajki korišten je Huawei P30 pametni telefon koji radi na Android 9.0 (Pie) operativnom sustavu i pokreće ga osmo-jezgreni procesor (2x2.6 GHz Cortex-A76, 2x1.92 GHz Cortex-A76 i 4x1.8 GHz Cortex-A55). Huawei P30 sadrži 6 GB radne memorije i ima OLED zaslon razlučivosti 2340x1080 piksela s gustoćom piksela od 422 piksela po inču. Također jedan od razloga zašto je izabran ovaj pametan telefon je to što ima podršku za video kodek H.265.

7.2. Mjerenja

Svi 360-videi su projicirani pravokutnom projekcijom, imaju razlučivosti 4K (4096x2048 piksela) i sadrže 30 sličica po sekundi. Duljina grupa slika (GOP) postavljena je na 32, što rezultira u trajanju segmenta od oko 1 sekunde. Svaki od razmatranih videozapisa nije prethodno podijeljen na pločice, stoga ih je potrebno kodirati pomoću Kvazaar koda [82][83]. Kako bi se to postiglo, prvi korak se sastojao od kodiranja svakog videozapisa iz MP4 u YUV datoteku. U rezultirajućem kodiranju koristile su se tri sheme popločavanja: 3x3, 8x4 i 12x4. Dodatno svaka shema popločavanja bila je kodirana u više razina kvalitete. Za različite razine korišteni su različiti kvantizacijski parametri (QP): 25, 30, 35, 40. Ti kvantizacijski parametri rezultirali su brzinom prijenosa (**Mbps**) prikazanim u navedenoj tablici (**Tablica 7.1**). Nadalje, u laboratorijskom testiranju se korištenjem Net.Sorm širina prijenosnog pojasa redoslijedno postavljala na 3 razine: 5 Mbps, 25Mbps, 50Mbps.

Također provedeno je još jedno ilustrativno testiranje u kojem su videozapisi od 360 stupnjeva, prije samog HEVC kodiranja, kodirani filterom boje. Razlog ovog testiranja je bolja preglednost samog rada tehnologija opisanih u ovom radu i lakše praćenje promjena pločica iz niske razlučivosti u visoku i obrnuto. Videozapisi su kodirani filterom boje prije samog HEVC kodiranja, iz razloga da ne bi došlo do slučajnih pogrešaka prilikom samog HEVC kodiranja i podjele na segmente, tj. poželjno je da sve reprezentacije budu jednako kodirane. Dakle, kodiranje bojom napravljeno je na početku, nad početno preuzetom MP4 datotekom koja se kasnije prebacuje u datoteku YUV formata. U obzir su uzete tri boje: crvena, plava i zelena. Segmenti najbolje rezolucije kodirani su plavom bojom, srednji zelenom, a segmenti najlošije kvalitete crvenom bojom. Za stavljanje filtera korišten je FFmpeg i neprovidnost (engl. *opacity*) od 0,7, a naredba koja se koristila je:

```
FFmpeg -i ime_ulazne_datoteke.mp4 -f lavfi -i  
„color=blue:s=4096x2048“ -filter-complex  
„blend=shortest=1:all_mode=overlay:all_opacity=0.7“  
ime_izlazne_datoteke.mp4
```

Umjesto plave boje (*blue*) navedene u ovom primjeru, koristile su se još *green* i *red*, dok su ostali parametri bili isti.

Tablica 7.1 Dobivene brzine prijenosa u Mbps 3 korištena videozapisa za različite kvantizacijske parametre i za različite sheme popločavanja

| Video | QP | 3×3 | 8×4 | 12×4 |
|--------------------------------------|-----------|------------|------------|-------------|
| Crystal Shower Falls | 25 | 43,480 | 43,689 | 43,792 |
| | 30 | 23,282 | 23,429 | 23,500 |
| | 35 | 10,526 | 10,628 | 10,696 |
| | 40 | 4,372 | 4,470 | 4,545 |
| London on Tower Bridge | 25 | 10,200 | 10,534 | 10,689 |
| | 30 | 4,840 | 5,086 | 5,210 |
| | 35 | 2,376 | 2,562 | 2,668 |
| | 40 | 1,258 | 1,411 | 1,507 |
| Ayutthaya – Easy Tripod Paint | 25 | 13,513 | 14,400 | 14,920 |
| | 30 | 6,460 | 7,109 | 7,435 |
| | 35 | 3,185 | 3,557 | 3,742 |
| | 40 | 1,624 | 1,844 | 1,963 |

7.3. Rezultati

U ovom poglavlju bit će navedeni subjektivni rezultati provedenih testiranja. Prvi test, u kojem je pretpostavka bila da će biti veći broj pločica niže kvalitete i u kojem se mjerio utjecaj prijenosne brzine (*bandwidth*) na strujanje videa, pokazao je da se većinom dogodio *stalling* videa, nego što su se pločice promijenile s više kvalitete na nižu. Najbolji odaziv, tj. neprekidan rad videozapisa uz najmanje kašnjenje u prebacivanju rezolucije pločica je bilo u shemi popločavanja 3x3 i brzini prijenosa 50 Mbps. Najveći *stalling* i skoro nikakvo prebacivanje pločica iz više rezolucije u nižu se dogodio kod sheme popločavanja 12x4 i brzine prijenosa 5 Mbps. Razlog tome je to što se kod sheme popločavanja od više redaka i stupaca povećava broj segmenata za prijenos te samim time i potražnja za većom prijenosnom brzinom.

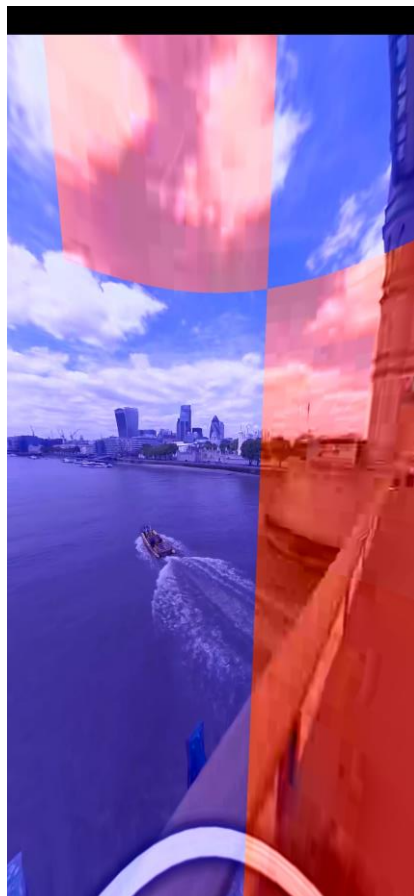
Nadalje, korespondencija između prosječne brzine prijenosa isporučene korisniku i prosječne brzine prijenosa dok se odvija komunikacija između poslužitelja i klijenta je manje trivijalna kod prilagodljivog strujanja 360-videa uz prostornu podjelu na pločice nego kod tradicionalnog strujanja videa. Na primjer, uzevši za primjer strujanje 360-videa koji koristi shemu popločavanja 12x4 pločica i vidno polje od devet pločica. Tijekom većeg dijela sesije strujanja prikazuje se vidno polje u najvećoj razlučivosti, dok su pločice izvan vidnog polja prikazane u najmanjoj rezoluciji. Ispada da je brzina isporučena korisniku (vidljiv dio korisničkog vidnog polja) ekvivalentna najvećoj rezoluciji (npr. u videozapisu *London on Tower Bridge* ta širina je bila 10Mbps), ali je ukupan *bitrate* videa, tj. prosječna potražnja za prijenosnom širinom tijekom sesije strujanja oko 5Mbps. To se događa zato što je veći dio videozapisa (koji nije vidljiv korisniku) dohvaćen u niskoj rezoluciji.

Sheme popločavanja također imaju utjecaj na subjektivan dojam korisnika. U slučaju sheme 3x3, sfera je podijeljena na 9 pločica koje su dosta velike u odnosu na korisničko vidno polje. Tada prilikom rotacije scene, ako korisnik rotira scenu, cijela pločica ili skup pločica će se prebaciti iz niske u visoku kvalitetu iako te pločice nisu u potpunosti i korisničkom vidnom polja. Ova shema daje najbrže rezultate, ali je iskustvena kvaliteta narušena jer velike pločice koje su niske rezolucije narušavaju ukupnu kvalitetu videozapisa. Za shemu popločavanja 12x4, sfera je podijeljena na 48 pločica i pločice su vrlo male u odnosu na cijelu sferu. To znači da korisnik kad rotira scenu, manji dijelovi sfere se prebacuju iz niže u višu kvalitetu, a to je mnogo ljepše nego kod sheme 3x3, ali uz cijenu performansi. Ova shema je dosta spora zato što je jako puno pločica i događa se mnogo HTTP zahtijeva između poslužitelja i klijenta. Stoga je optimalna shema 8x4. Ona je kombinacija dviju prethodno navedenih shema i pruža najbolje rezultate. **Slika 7.2** prikazuje uhvaćeni mrežni promet u programu Wireshark, gdje se vidi početak komunikacije između poslužitelja protokola HTTP i klijenta. Za komunikaciju je korišten protokol je TCP te se koristio filter za ljepši prikaz u Wiresharku, a filter samo ispisuje pakete koje prima ili šalje poslužitelj (IP adrese 192.168.0.29).

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|--------------|--------------|----------|--------|---|
| 49 | 40.714323 | 192.168.0.27 | 192.168.0.29 | TCP | 74 | 44690 → 8000 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TSval=36836118 TSecr=0 WS=256 |
| 50 | 40.714436 | 192.168.0.29 | 192.168.0.27 | TCP | 66 | 8000 → 44690 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 51 | 40.717872 | 192.168.0.27 | 192.168.0.29 | TCP | 54 | 44690 → 8000 [ACK] Seq=1 Ack=1 Win=87808 Len=0 |
| 52 | 40.718293 | 192.168.0.27 | 192.168.0.29 | HTTP | 224 | GET /test.mpd HTTP/1.1 |
| 53 | 40.719491 | 192.168.0.29 | 192.168.0.27 | TCP | 256 | 8000 → 44690 [PSH, ACK] Seq=1 Ack=171 Win=131072 Len=202 [TCP segment of a reassembled PDU] |
| 54 | 40.720071 | 192.168.0.29 | 192.168.0.27 | TCP | 1514 | 8000 → 44690 [ACK] Seq=203 Ack=171 Win=131072 Len=1460 [TCP segment of a reassembled PDU] |
| 55 | 40.720072 | 192.168.0.29 | 192.168.0.27 | TCP | 1514 | 8000 → 44690 [ACK] Seq=1663 Ack=171 Win=131072 Len=1460 [TCP segment of a reassembled PDU] |
| 56 | 40.720076 | 192.168.0.29 | 192.168.0.27 | TCP | 1514 | 8000 → 44690 [ACK] Seq=3123 Ack=171 Win=131072 Len=1460 [TCP segment of a reassembled PDU] |
| 57 | 40.720097 | 192.168.0.29 | 192.168.0.27 | TCP | 1514 | 8000 → 44690 [ACK] Seq=4583 Ack=171 Win=131072 Len=1460 [TCP segment of a reassembled PDU] |
| 58 | 40.720098 | 192.168.0.29 | 192.168.0.27 | TCP | 1514 | 8000 → 44690 [ACK] Seq=6043 Ack=171 Win=131072 Len=1460 [TCP segment of a reassembled PDU] |
| 59 | 40.720100 | 192.168.0.29 | 192.168.0.27 | TCP | 1514 | 8000 → 44690 [ACK] Seq=7503 Ack=171 Win=131072 Len=1460 [TCP segment of a reassembled PDU] |
| 60 | 40.720102 | 192.168.0.29 | 192.168.0.27 | TCP | 1514 | 8000 → 44690 [ACK] Seq=8963 Ack=171 Win=131072 Len=1460 [TCP segment of a reassembled PDU] |
| 61 | 40.720103 | 192.168.0.29 | 192.168.0.27 | TCP | 1514 | 8000 → 44690 [ACK] Seq=10423 Ack=171 Win=131072 Len=1460 [TCP segment of a reassembled PDU] |
| 62 | 40.720105 | 192.168.0.29 | 192.168.0.27 | TCP | 1514 | 8000 → 44690 [ACK] Seq=11883 Ack=171 Win=131072 Len=1460 [TCP segment of a reassembled PDU] |
| 63 | 40.724718 | 192.168.0.27 | 192.168.0.29 | TCP | 54 | 44690 → 8000 [ACK] Seq=171 Ack=203 Win=88832 Len=0 |
| 64 | 40.724766 | 192.168.0.29 | 192.168.0.27 | TCP | 1514 | 8000 → 44690 [ACK] Seq=13343 Ack=171 Win=131072 Len=1460 [TCP segment of a reassembled PDU] |
| 65 | 40.724923 | 192.168.0.27 | 192.168.0.29 | TCP | 54 | 44690 → 8000 [ACK] Seq=171 Ack=1663 Win=91648 Len=0 |
| 66 | 40.724960 | 192.168.0.29 | 192.168.0.27 | TCP | 1514 | 8000 → 44690 [ACK] Seq=14803 Ack=171 Win=131072 Len=1460 [TCP segment of a reassembled PDU] |
| 67 | 40.724961 | 192.168.0.29 | 192.168.0.27 | TCP | 1514 | 8000 → 44690 [ACK] Seq=16263 Ack=171 Win=131072 Len=1460 [TCP segment of a reassembled PDU] |
| 68 | 40.729389 | 192.168.0.27 | 192.168.0.29 | TCP | 54 | 44690 → 8000 [ACK] Seq=171 Ack=3123 Win=94720 Len=0 |
| 69 | 40.729416 | 192.168.0.29 | 192.168.0.27 | TCP | 1514 | 8000 → 44690 [ACK] Seq=17723 Ack=171 Win=131072 Len=1460 [TCP segment of a reassembled PDU] |
| 70 | 40.729417 | 192.168.0.29 | 192.168.0.27 | TCP | 1514 | 8000 → 44690 [ACK] Seq=19183 Ack=171 Win=131072 Len=1460 [TCP segment of a reassembled PDU] |
| 71 | 40.730028 | 192.168.0.27 | 192.168.0.29 | TCP | 54 | 44690 → 8000 [ACK] Seq=171 Ack=4583 Win=97536 Len=0 |
| 72 | 40.730076 | 192.168.0.29 | 192.168.0.27 | HTTP | 402 | HTTP/1.0 200 OK |
| 73 | 40.730764 | 192.168.0.27 | 192.168.0.29 | TCP | 54 | 44690 → 8000 [ACK] Seq=171 Ack=6043 Win=100352 Len=0 |
| 74 | 40.730868 | 192.168.0.27 | 192.168.0.29 | TCP | 54 | 44690 → 8000 [ACK] Seq=171 Ack=7503 Win=103424 Len=0 |
| 75 | 40.730869 | 192.168.0.27 | 192.168.0.29 | TCP | 54 | 44690 → 8000 [ACK] Seq=171 Ack=8963 Win=106240 Len=0 |
| 76 | 40.730870 | 192.168.0.27 | 192.168.0.29 | TCP | 54 | 44690 → 8000 [ACK] Seq=171 Ack=10423 Win=109312 Len=0 |
| 77 | 40.730870 | 192.168.0.27 | 192.168.0.29 | TCP | 54 | 44690 → 8000 [ACK] Seq=171 Ack=11883 Win=112128 Len=0 |
| 78 | 40.730873 | 192.168.0.27 | 192.168.0.29 | TCP | 54 | 44690 → 8000 [ACK] Seq=171 Ack=13343 Win=115200 Len=0 |

Slika 7.2 Uhvaćen mrežni promet prilagodljivog strujanja uz prostornu podjelu na pločice u Wiresharku

Drugi provedeni eksperiment je bio onaj u kojem su videozapisi prije samog HEVC kodiranja, kodirani filterom boje. U obzir su uzete tri boje: crvena, plava i zelena. Eksperiment je na ilustrativan način pokazao rad prilagodljivog strujanja uz prostornu podjelu na pločice. **Slika 7.3** prikazuje provedeni eksperiment. Naime, ovaj eksperiment je dao neočekivane rezultate. Kodirane su tri reprezentacije videozapisa s tri različite kvalitete i tri različite boje. Podijeljene su na DASH segmente, a shema popločavanja je bila 8x4. Očekivalo se da će se prikazati sve tri boje. Da će se plavo kodirani video prikazati u korisničkom vidnom polju, zeleno kodirani na rubovima vidnog polja, a crveno kodirani izvan korisničkog vidnog polja. Što se zaista dogodilo je to da su se prilikom strujanja prikazale samo najniža i najviša kvaliteta, tj. plava i crvena, dok se zelena nije prikazivala. Ideja zašto se to dogodilo je ta da dekodirani nije znao renderirati zeleno dekodirano strujanje, što je možda greška kodiranja samog strujanja ili greška prilikom renderiranja na Android pametnom uređaju. Potrebno je u budućem radu ovo bolje proučiti i vidjeti što se zaista dogodilo.



Slika 7.3 Drugi test prilagodljivog strujanja uz prostornu podjelu na pločice gdje se koristio filter boja

8. Zaključak i budući rad

Aplikacije za strujanje video sadržaja brzo se razvijaju. Kako bi se riješili izazovi ovakvih aplikacija, kao što je velika potražnja za mrežnim resursima, znanstvena zajednica i komercijalni pružatelji usluga predložili su razne tehnike za optimizaciju videozapisa od 360 stupnjeva. Međutim, ova složena interakcija između tehnika optimizacija 360-videa i promjenjivih mrežnih uvjeta nije trivijalna niti je dovoljno istražena. Također, besplatno dostupna rješenja za implementaciju prilagodljivog strujanja nisu još dovoljno razrađena, a ni dokumentirana.

Zbog toga su u ovom radu predložene i opisane implementacije koje koriste besplatno dostupne alate i omogućuju strujanje i reprodukciju videozapisa od 360 stupnjeva u OMAF standardu uz prostornu podjelu na pločice. OMAF standard omogućuje povećanje razlučivosti videozapisa unutar vidnog okvira žrtvujući rezoluciju područja koja nisu prikazana korisniku. Realizirane su tehnike koje optimiziraju strujanje 360-videa i omogućavaju uštedu mrežnih resursa. Kroz različite pristupe kodiranja i popločavanja te različite mrežne uvjete (širina prijenosnog pojasa), pokazano je da shema popločavanja 8x4 daje najbolji rezultate na subjektivan dojam korisnika. Također se na ilustrativan način pokazao rad samih tehnologija.

Za budući rad, bilo bi potrebno unaprijediti ovo rješenje na način da se strujanje 360-videa prikazuje u virtualnoj stvarnosti putem VR zaslona na glavi. Tada bi se provelo subjektivno istraživanje i analiza iskustvene kvalitete korisnika u različitim načinima kodiranja videa. Nadalje, potrebno je izvršiti opširniju evaluaciju predloženih optimizacijskih rješenja u promjenjivim uvjetima mreže. Konačno, planiraju se istražiti prednosti prilagodljivog strujanja putem protokola HTTP/2 i nadolazećeg mrežnog protokola HTTP/3, budući da paralelno preuzimanje pločica može dovesti do boljeg korištenja prijenosne širine. Dolaskom nove, pete generacije mobilnih mreža postojeća rješenja će se morati optimizirati kako bi što bolje radila prilikom velikih brzina prijenosa. Za kraj, potrebno je vidjeti drugačije projekcije videozapisa i njihov utjecaj na performanse i iskustvenu kvalitetu strujanja.

Literatura

- [1] C. Welch, "Facebook buying Oculus VR for \$2 billion", Ožujak 2014.,
Dostupno: <https://www.theverge.com/2014/3/25/5547456/facebook-buying-oculus-for-2-billion>
Datum pristupa: 13.04.2020.
- [2] CISCO. 2019. „Cisco Visual Networking Index: Forecast and Trends, Technical Report“, Cisco Systems, 2017.–2022.
- [3] X. Corbillon, G. Simon, A. Devlic, i J. Chakareski, „Viewport-adaptive navigable 360-degree video delivery“, In Proceedings of the IEEE International Conference on Communications (ICC 2017.)
Dostupno: <https://doi.org/10.1109/icc.2017.7996611>
- [4] „Don't touch that dial: How YouTube s bringing adaptive streaming to mobile, TVs“,
Dostupno: <https://gigaom.com/2013/03/13/youtube-adaptive-streaming-mobile-tv/>
Datum pristupa: 1.6.2020.
- [5] S. Afzal, C. Jiasi, and K. K. Ramakrishnan, "Characterization of 360-degree Videos.", Proceedings of the Workshop on Virtual Reality and Augmented Reality Network. ACM, 2017.
- [6] „4 things to know about VR before you buy a headset“,
Dostupno: <https://toast.gg/4-things-to-know-about-vr-before-you-buy-a-headset/>
Datum pristupa: 15.04.2020.
- [7] R. I. T. Da Costa Filho, W. Lautenschlager, N. Kagami, V. Roesler, i L. P. Gasparly, „Network Fortune Cookie: Using Network Measurements To Predict Video Streaming Performance And Qoe“, In Proceedings Of The IEEE Global Communications Conference (Globecom 2016.)
Dostupno: <https://doi.org/10.1109/glocom.2016.7842022>
- [8] R. I. T. Da Costa Filho, M. C. Luizelli, M. T. Vega, J. V. D. Hooft, S. Petrangeli, T. Wauters, F. De Turck, And L. P. Gasparly, „Predicting The Performance Of Virtual Reality Video Streaming In Mobile Networks“, In Proceedings Of The 9th ACM Multimedia Systems Conference ACM, New York, NY (MMSys 2018.),
Dostupno: <https://doi.org/10.1145/3204949.3204966>
- [9] Samsung Gear VR specifikacije:
Dostupno: <https://www.samsung.com/global/galaxy/gear-vr/specs/>
Datum pristupa: 19.05.2020.
- [10] MPEG OMAF specifikacije:
Dostupno: <https://mpeg.chiariglione.org/standards/mpeg-i/omnidirectional-media-format>
Datum pristupa: 19.05.2020.
- [11] Z. Chen, Y. Li, i Y. Zhang, „Recent advances in omnidirectional video coding for virtual reality: Projection and evaluation“, Signal Processing, Elsevier, 2018.,
Dostupno: <https://doi.org/10.1016/j.sigpro.2018.01.004>
- [12] M. Graf, C. Timmerer, i C. Mueller, „Towards bandwidth-efficient adaptive streaming of omnidirectional video over HTTP: Design, implementation, and evaluation“, In Proceedings of the 8th ACM Conference on Multimedia Systems Conference. ACM, New York, NY, (MMSys 2017.)
Dostupno: <https://doi.org/10.1145/3083187.3084016>

- [13] S. Petrangeli, V. Swaminathan, M. Hosseini, i F. De Turck, "An http/2-based adaptive streaming framework for 360 virtual reality videos," in Proceedings of the 2017 ACM on Multimedia Conference, New York, NY, USA: ACM, 2017.,
Dostupno: <http://doi.acm.org/10.1145/3123266.3123453>
- [14] G. Dimopoulos, I. Leontiadis, P. Barlet-Ros, and K. Papagiannaki, "Measuring video qoe from encrypted traffic," in Proceedings of the 2016 Internet Measurement Conference, New York, NY, USA: ACM, 2016.
Dostupno: <http://doi.acm.org/10.1145/2987443.2987459>
- [15] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet" IEEE Multimedia, 2011.,
Dostupno: <https://ieeexplore.ieee.org/document/6077864>
- [16] C. Concolato, J. L. Feuvre, F. Denoual, E. Nassor, N. Ouedraogo, i J. Taquet, "Adaptive Streaming of HEVC Tiled Videos Using MPEG-DASH" IEEE Transactions on Circuits and Systems for Video Technology, 2017.
- [17] O. A. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, i S. Y. Lim, "MPEG DASH SRD: Spatial relationship description" in Proceedings of the 7th International Conference on Multimedia Systems, New York, NY, USA: ACM, 2016.,
Dostupno: <http://doi.acm.org/10.1145/2910017.2910606>
- [18] F. Qian, L. Ji, B. Han, i V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks" in Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges, New York, NY, USA: ACM, 2016.,
Dostupno: <http://doi.acm.org/10.1145/2980055.2980056>
- [19] A. Rommel, „Fraunhofer HHI at Mobile World Congress with VR and 5G technologies“, Fraunhofer-Institut für Nachrichtentechnik, Heinrich-Hertz-Institut, HHI, 2017.
Dostupno: <https://www.innovations-report.com/html/reports/trade-fair-news/fraunhofer-hhi-at-mobile-world-congress-with-vr-and-5g-technologies.html>
- [20] E. Kuzyakov, „End-to-end optimizations for dynamic streaming“, Veljača 2017.,
Dostupno: <https://code.facebook.com/posts/637561796428084>
Datum pristupa: 19.05.2020.
- [21] C. Zhou, Z. Li, i Y. Liu, „A measurement study of Oculus 360 degree video streaming“, In Proceedings of the 8th ACM Conference on Multimedia Systems Conference. ACM, New York, NY, (MMSys 2017.)
- [22] M. Yu, H. Lakshman, i B. Girod, „A Framework to Evaluate Omnidirectional Video Coding Schemes“, 2015 IEEE International Symposium on Mixed and Augmented Reality, Fukuoka, 2015.
- [23] E. Kuzyakov i D. Pio, "Next-generation video encoding techniques for 360 video and VR", Siječanj 2016.
Dostupno: <https://engineering.fb.com/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/>
Datum pristupa: 19.05.2020.
- [24] F. Quan, B. Han, L. Ji, i V. Gopalakrishnan, „Optimizing 360 video delivery over cellular networks“, ATC '16: Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges, 2016.
- [25] M. Hosseini i V. Swaminathan, „Adaptive 360 VR video streaming: Divide and conquer“, In Proceedings of the IEEE International Symposium on Multimedia (ISM 2016.),
Dostupno: <https://doi.org/10.1109/ISM.2016.0028>

- [26] PSNR, Dostupno: <https://www.mathworks.com/help/vision/ref/psnr.html>
Datum pristupa: 19.05.2020.
- [27] SSIM, Dostupno: <https://www.mathworks.com/help/images/ref/ssim.html>
Datum pristupa: 19.05.2020.
- [28] R. Schatz, A. Sackl, C. Timmerer, i B. Gardlo. „Towards subjective quality of experience assessment for omnidirectional video streaming“, Ninth International Conference on Quality of Multimedia Experience (QoMEX), Svibanj 2017., Dostupno: <https://doi.org/10.1109/QoMEX.2017.7965657>
- [29] E. M. Kolasinski, „Simulator Sickness in Virtual Environments“, U.S. Army Research Institute for the Behavioral and Social Sciences, 1995.
- [30] F. Nielsen, „Surround video: a multihead camera approach“, The Visual Computer, 2005., Dostupno: <https://link.springer.com/article/10.1007/s00371-004-0273-z>
- [31] "Vuze Official site", Dostupno: <https://vuze.camera/camera/vuze-camera/>
Datum pristupa: 19.05.2020.
- [32] "Ricoh Theta S review“, CNET, CBS Interactive, Dostupno: <https://www.cnet.com/reviews/ricoh-theta-s-review/>
Datum pristupa: 12.05.2020.
- [33] S. Byford, "Kogeto iCONIC panoramic video coming to Galaxy Nexus and more", The Verge, 2012., Dostupno: <https://www.theverge.com/2012/1/6/2687245/kogeto-iconic-panoramic-lens-galaxy-nexus>
Datum pristupa: 12.05.2020.
- [34] "360-degree cameras: The best VR cams, no matter your budget". Pocket-lint, Dostupno: <https://www.pocket-lint.com/cameras/buyers-guides/137301-best-360-cameras-the-best-vr-and-360-video-cameras-no-matter-your-budget>
Datum pristupa: 12.05.2020.
- [35] Garmin VIRB kamera, Dostupno: <https://buy.garmin.com/hr-HR/HR/p/562010>
Datum pristupa: 12.05.2020.
- [36] GoPro Omni, Dostupno: https://gopro.com/en/us/news/vr_spherical_omni_rig_shipping
Datum pristupa: 12.05.2020.
- [37] GoPro Odyssey, Dostupno: <https://gopro.com/en/us/news/here-is-odyssey>
Datum pristupa: 12.05.2020.
- [38] YouTube Engineering and Developers Blog, ožujak 2017. Dostupno: <https://youtube-eng.googleblog.com/2017/03/improving-vr-videos.html>
Datum pristupa: 19.05.2020.
- [39] "How to Use 360° Equirectangular Panoramas for Greater Realism in Games", Dostupno: https://medium.com/@onix_systems/how-to-use-360-equirectangular-panoramas-for-greater-realism-in-games-55fadb0547da
Datum pristupa: 22.05.2020.
- [40] D. Pio, E. Kuzyakov, "Under the hood: Building 360 video", Dostupno: <https://engineering.fb.com/video-engineering/under-the-hood-building-360-video/>
Datum pristupa: 22.05.2020.

- [41] C. Brown, "Bringing pixels front and center in VR video", 2017.
Dostupno: <https://blog.google/products/google-vr/bringing-pixels-front-and-center-vr-video/>
Datum pristupa: 22.05.2020.
- [42] Wired, "Youtube 360 degree video", 2015., Dostupno:
<https://www.wired.com/2015/03/youtube-360-degree-video/>
Datum pristupa: 22.05.2020.
- [43] R. Amadeo, "YouTube's "VR180" format cuts down on VR video's prohibitive requirements",
Dostupno: <https://arstechnica.com/gadgets/2017/06/youtubes-vr180-format-makes-vr-video-more-accessible/>
Datum pristupa: 22.05.2020.
- [44] A. Strange, "YouTube's VR 180 and Daydream cameras bring immersive video to traditional creators", 2017.,
Dostupno: <https://mashable.com/2017/06/23/youtube-daydream-vr-180/?europa=true>
Datum pristupa: 19.05.2020.
- [45] B. Ayrey, "Introducing Facebook 360 For Gear VR", 2017.,
Dostupno: <https://about.fb.com/news/2017/03/introducing-facebook-360-for-gear-vr/>
Datum pristupa: 19.05.2020.
- [46] G. J. Sullivan, J. Ohm, W. Han i T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," in IEEE Transactions on Circuits and Systems for Video Technology, 2012,
Dostupno: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6316136>
- [47] Elemental Technologies, "HEVC DEMYSTIFIED, A Primer on the H.265 Video Codec", 2013., Dostupno: https://www.e-idsolutions.com/wp-content/uploads/2016/10/HEVC_Demystified.pdf
- [48] Podržani formati podataka na YouTubeu,
Dostupno: <https://support.google.com/youtube/troubleshooter/2888402?hl=en>
Datum pristupa: 08.06.2020.
- [49] Prilagodljivo strujanje,
Dostupno: <https://bitmovin.com/adaptive-streaming/>
Datum pristupa, 19.5.2020:
- [50] H. Schulzrinne, S. Casner, R. Frederick i V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", srpanj 2003., RFC 3550,
Dostupno: <https://tools.ietf.org/html/rfc3550#section-1.1>
- [51] H. Schulzrinne, A. Rao i R. Lanphier, "Real Time Streaming Protocol (RTSP)", travanj 1998, RFC 2326,
Dostupno: <https://tools.ietf.org/html/rfc2326>
- [52] S. Akhshabi, A. C. Begen; C. Dovrolis, "An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP", In Proceedings of the second annual ACM conference on Multimedia systems (MMSys 2011.), New York, NY, USA: ACM
- [53] A. Bentaleb, B. Taani, A. Begen, C. Timmermer, and R. Zimmermann, "A Survey on Bitrate Adaptation Schemes for Streaming Media over HTTP", In IEEE Communications Surveys & Tutorials (IEEE COMST), 2018.
- [54] C. Mueller, S. Lederer, C. Timmerer, "DASH at ITEC, VLC Plugin, DASHEncoder and Dataset", ACM Multimedia 2019, Nice, France, Listopad 2019.

- [55] C. Müller, S. Lederer and C. Timmerer, "An Evaluation of Dynamic Adaptive Streaming over HTTP in Vehicular Environments", MoVid '12: Proceedings of the 4th Workshop on Mobile Video, 2012.
Dostupno: <https://dl.acm.org/doi/10.1145/2151677.2151686>
- [56] G. Liz, "The Next Big Thing in Video: Adaptive Bitrate Streaming", 2010.
Dostupno: <https://gigaom.com/report/how-to-deliver-as-much-video-as-users-can-take/>
Datum pristupa: 19.05.2020.
- [57] Prilagodljivo strujanje s različitim brzinama prijenosa,
Dostupno: https://en.wikipedia.org/wiki/Adaptive_bitrate_streaming
Datum pristupa: 17.04.2020.
- [58] K. Spiteri, R. Uргаonkar, K. R. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos. IEEE INFOCOM, 2016.
- [59] Spiteri, Sitaraman and Sparacio, "From Theory to Practice: Improving Bitrate Adaptation in the DASH Reference Player", ACM Multimedia Systems Conference, Lipanj 2018.
- [60] MPEG-DASH standard,
Dostupno: <https://mpeg.chiariglione.org/standards/mpeg-dash>
Datum pristupa: 19.05.2020.
- [61] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, i P. Trangia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," IEEE Communications Surveys Tutorials, 2015.
- [62] J. Famaey, S. Latré, N. Bouten, W. Van de Meerssche, B. De Vleeschauwer, W. Van Leekwijck, F. De Turck, „On the merits of SVC-based HTTP Adaptive Streaming“, IFIP/IEEE International Symposium on Integrated Network Management (IM 2013.)
- [63] S. Akhshabi, A. C. Begen, C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP", In Proceedings of the second annual ACM conference on Multimedia systems (MMSys 2011.),
Dostupno: <https://dl.acm.org/doi/proceedings/10.1145/1943552>
- [64] Slika prilagodljivog strujanja uz prostornu podjelu na pločice
Dostupno: https://www.researchgate.net/figure/Tiled-FoV-mapping-of-a-users-3DoF-pose-in-the-EQR-projection-of-a-360-video-frame_fig1_329102051
- [65] OpenSignal, „State of Mobile Networks: USA“, Siječanj 2018.,
Dostupno: <https://opensignal.com/reports/2018/01/usa/state-of-the-mobile-network/>
Datum pristupa: 19.05.2020.
- [66] Jiang, Junchen, V. Sekar, and H. Zhang. "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive." IEEE/ACM Transactions on Networking (TON), 2014.
- [67] U. Pal i H. King, „Effect of UHD high frame rates (HFR) on DVB-S2 bit error rate (BER)“, SMPTE, 2015.
- [68] M. Woo, J. Neider, T. Davis, i D. Shreiner, „OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2 (3rd. ed.)“, Addison-Wesley Longman Publishing Co., Inc., USA, 1999.
- [69] Revised text of ISO/IEC FDIS 23090-2 Omnidirectional Media Format, Travanj 2018.

- [70] Specifikacije OMAF standarda,
Dostupno: <https://mpeg.chiariglione.org/standards/mpeg-i/omnidirectional-media-format>
Datum pristupa: 19.05.2020.
- [71] R. Ghaznavi-Youvalari i suradnici, "Comparison of HEVC coding schemes for tile-based viewport-adaptive streaming of omnidirectional video," 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP), Luton, 2017.
Dostupno: <https://ieeexplore.ieee.org/document/8122227>
- [72] A. Zare, A. Aminlou, and M. M. Hannuksela, „6K Effective Resolution with 4K HEVC Decoding Capability for OMAF-compliant 360° Video Streaming“, In Proceedings of the 23rd Packet Video Workshop (PV '18), Association for Computing Machinery, New York, NY, USA, 2018.
Dostupno: <https://doi.org/10.1145/3210424.3210425>
- [73] GPAC GitHub stranica:
Dostupno: <https://github.com/gpac/gpac/wiki/HEVC-Tile-based-adaptation-guide>
Datum pristupa: 22.03.2020.
- [74] Stranica za preuzimanje GPAC-a, Dostupno: <https://gpac.wp.imt.fr/downloads/>
Datum pristupa: 22.03.2020.
- [75] Online stranica za preuzimanje 360-videa,
Dostupno: <https://www.mettle.com/360vr-master-series-free-360-downloads-page/>
Datum pristupa: 22.03.2020.
- [76] Python http.server, Dostupno: <https://docs.python.org/3/library/http.server.html>
Datum pristupa: 22.03.2020.
- [77] GitHub Nokiatech Omaf,
Dostupno: <https://github.com/nokiatech/omaf>
Datum pristupa: 22.03.2020.
- [78] Visual Studio,
Dostupno: <https://visualstudio.microsoft.com/vs/whatsnew/>
Datum pristupa: 22.03.2020.
- [79] Android Studio,
Dostupno: <https://developer.android.com/studio>
Datum pristupa: 22.03.2020.
- [80] Wireshark,
Dostupno: <https://www.wireshark.org/>
Datum pristupa: 22.03.2020.
- [81] FFMPEG,
Dostupno: <https://ffmpeg.org/>
Datum pristupa: 22.03.2020.
- [82] Kvazaar GitHub stranica,
Dostupno: <https://github.com/ultravideo/kvazaar>
Datum pristupa: 22.03.2020.
- [83] Ultra Video Group,
Dostupno: <http://ultravideo.cs.tut.fi/#main>
Datum pristupa: 22.03.2020.
- [84] Net.Storm službena stranica,
Dostupno: <http://www.albedotelecom.com/pages/fieldtools/src/netstorm.php>
Datum pristupa: 15.06.2020.

Sažetak

Prilagodljivo strujanje videa od 360 stupnjeva

Prikaz videa od 360 stupnjeva putem tehnologije virtualne stvarnosti (engl. Virtual Reality, VR) omogućeno je mapiranjem sferičnog videa u 3D geometriju. Koristeći VR zaslon na glavi, korisnikova točka gledišta nalazi se u središtu sfere te se može pomicati oko triju osi (3 degrees of freedom, 3DoF). Strujanje 360-videa zahtjeva velike brzine prijenosa, što predstavlja izazov za mrežu. Zbog ograničenog mrežnog kapaciteta, potrebno je istražiti mogućnosti smanjenja i optimiranja brzine prijenosa videa. Određene tehnike baziraju se na praćenju korisnikovog područja gledanja (engl. Field of View, kratica: FoV) te prijenosa videa tako da je dio koji korisnik trenutno promatra prikazan u velikoj rezoluciji dok je dio koji korisnik ne gleda prikazan u manjoj rezoluciji.

Cilj ovog rada je istražiti i opisati mehanizme strujanja videa od 360 stupnjeva putem komunikacijske mreže. Proučena je i opisana norma *Omnidirectional Media Format* (OMAF), predložena od strane MPEG-a, koja standardizira format medija za razvoj aplikacija temeljenih na prikazu videa od 360 stupnjeva. Nadalje, implementiran je i evaluiran laboratorijski prototip kojim se ostvaruje prilagodljivo strujanje videa od 360 stupnjeva i prikaz putem zaslona pametnog telefona.

Ključne riječi: prilagodljivo strujanje, prilagodljivo strujanje uz prostornu podjelu na pločice, video od 360 stupnjeva, OMAF, HEVC, virtualna stvarnost

Summary

Adaptive 360-degree Video Streaming

360-degree video playback through Virtual Reality (VR) technology is made possible by mapping spherical video to 3D geometry. Using a VR head-mounted display (HMD), the user's point of view is at the center of the sphere and can be moved around three degrees of freedom (3DoF). 360-degree video streaming requires high bit rates, which is a challenge for the network. Due to the limited network capacity, it is necessary to explore the possibilities of reducing and optimizing the video transmission speed. Certain techniques are based on monitoring the user's field of view (abbreviation: FoV) and transmitting video so that the part currently viewed by the user is displayed in high resolution while the non-viewing part is displayed in lower resolution.

The aim of this thesis is to investigate and describe the mechanisms of 360-degree video streaming through a communication network. The Omnidirectional Media Format (OMAF) standard, proposed by MPEG, which standardizes the media format for the development of 360-degree video display applications, has been studied and described. Furthermore, a laboratory prototype was implemented and evaluated to provide adaptive 360 degree video stream and playback through the smartphone screen.

Keywords: adaptive streaming, adaptive tilebased streaming, 360 degree video, OMAF, HEVC, virtual reality

Popis slika

| | |
|---|----|
| Slika 1.1 Razlika između 3DoF-a i 6DoF-a (slika preuzeta s [6]) | 2 |
| Slika 2.1 Tehnologija strujanja uz prostornu podjelu na pločice (slika preuzeta s [19]) | 6 |
| Slika 3.1 Garmin VIRB 360 kamera (slika preuzeta s [35]) | 10 |
| Slika 3.2 Primjer pravokutne projekcije (slika preuzeta s [39]) | 11 |
| Slika 3.3 Primjer projekcije kocke (slika preuzeta s [39]) | 12 |
| Slika 3.4 Ilustracija kompenzacije gibanja i vektora pomaka | 15 |
| Slika 3.5 Redoslijed I-, P- i B- okvira | 16 |
| Slika 3.6 Primjer DC metode predikcije | 18 |
| Slika 3.7 Shema prilagodljivog strujanja putem protokola HTTP [57] | 20 |
| Slika 3.8 Ovisnost propusnosti mreže o preuzetim segmentima [57] | 20 |
| Slika 3.9 Koncept tehnologije prilagodljivog strujanja uz prostornu podjelu na pločice u VR-u (slika preuzeta s [64]) | 22 |
| Slika 3.10 Pregled rada sustava prilagodljivog strujanja uz prostornu podjelu na pločice | 24 |
| Slika 4.1 Primjeri RWP-a za pravokutnu projekciju; a) određivanje sadržaja koji ne pokriva cijelu sferu; b) kompenzacija prekomjernog uzorkovanja oko polova pravokutno projicirane slike [69] | 28 |
| Slika 5.1 Reprodukcija 360-videa <i>London on Tower Bridge</i> u MP4Client-u | 39 |
| Slika 5.2 Arhitektura sustava prilagodljivog strujanja uz prostornu podjelu na pločice | 41 |
| Slika 5.3 Tijek Kvazaar kodiranja i ispis u Windows PowerShellu | 45 |
| Slika 5.4 Primjer konfiguracijske JSON datoteke | 46 |
| Slika 5.5 Scena za odabir datoteke za strujanje (lijevo); Scena prilagodljivog strujanja uz prostornu podjelu na pločice (desno) | 47 |
| Slika 7.1 Postavljanje laboratorijskog eksperimenta | 52 |

Slika 7.2 Uhvaćen mrežni promet prilagodljivog strujanja uz prostornu podjelu na pločice u Wiresharku 56

Slika 7.3 Drugi test prilagodljivog strujanja uz prostornu podjelu na pločice gdje se koristio filter boja 57

Popis tablica

| | |
|--|----|
| Tablica 4.1 Vrste shema za videozapise s ograničenim pristupom | 30 |
| Tablica 4.2 MPD deskriptori za karakterizaciju svesmjernog videa | 31 |
| Tablica 4.3 Pregled profila OMAF videa | 32 |
| Tablica 4.4 Pristupi za strujanje ovisno o vidnom okviru temeljeno na prostornoj podjeli na pločice | 33 |
| Tablica 5.1 Detalji preuzetog videa | 43 |
| Tablica 7.1 Dobivene brzine prijenosa 3 korištena videozapisa za različite kvantizacijske parametre i za različite sheme popločavanja | 54 |