

Primerjava HEVC in VP9

David Valič¹, Marko Mež², Urban Burnik²

¹ Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Fakulteta za elektrotehniko

² Univerza v Ljubljani, Fakulteta za elektrotehniko,

E-pošta: dv6510@student.uni-lj.si

Abstract: The paper reviews and compares currently well-established video codecs HEVC and VP9 at several levels. The theoretical background and presentation of standard-specific compression stamps and procedures is followed by performance evaluation using characteristic video recordings with comments and conclusions.

1 Uvod

Danes se video vsebine širijo bolj kot še nikoli prej, predvsem zaradi porasta uporabe socialnih omrežij in orodij za takojšnje komunikacije. Da bi omogočili čim učinkovitejši pretok vsebin, je potreben tudi napredek v kompresiji videa. Danes sta uveljavljena kodeka HEVC (angl. *High Efficiency Video Codec*) in VP9. Cilj prispevka je primerjava obeh kodekov na različnih ravneh. Predstavljena je uporaba navedenih kodekov, sledi primerjava delovanja ter opis skupnih lastnosti in razlik. Prispevek se zaključuje s praktično primerjavo obeh kodekov.

2 Ozadje kodekov HEVC in VP9 ter njuna uporaba

Kodek HEVC oz. H.265 sta leta 2013 standardizirali organizaciji ITU-T Video Coding Experts Group in ISO/IEC JTC1 Moving Picture Experts Group v okviru delovne skupine Joint Collaborative Team on Video Coding. Gre za naslednika kodeka AVC (angl. *Advanced Video Codec*) in prinaša prihranke do 50 % in več na pasovni širini pri enakovredni kvaliteti, sicer pa so prihranki odvisni tudi od vsebine posnetka.

VP9 je razvilo podjetje Google in je bil prvotno objavljen leta 2013 kot prosto-dostopni format za kompresijo videa. Tudi sama implementacija kodeka je odprtokodna.

Ker je za uporabo kodeka HEVC potrebna licenca, je najbolj razširjen v namenskih multimedijских napravah, kot so kamere, pametni telefoni, televizije in računalniki. Za zasebno in spletno rabo pa je zaradi dostopnosti pogostejše v uporabi VP9. Največji uporabnik kodeka VP9 je skoraj zagotovo YouTube [5]. Tudi podjetje Netflix že od leta 2016 uporablja VP9 kot primarni kodek, od februarja 2020 pa ga za predvajanje na mobilnih napravah z operacijskim sistemom Android že nadomešča z naslednikom AV1 [9]. Slednji se danes uporablja tudi na Youtube-u.

3 Delovanje kodekov HEVC in VP9

Oba standarda za kompresijo temeljita na blokovnih operacijah v frekvenčnem prostoru. V osnovi je postopek dokaj podoben z nekaterimi razlikami v detajlih.

3.1 Osnovni princip delovanja kodekov

Preden se izvede katerokoli operacija, se posamična slika razdeli na ti. makrobloke, znotraj katerih se izvaja nadaljnje operacije. Makroblok je razdeljen na še manjše bloke, v okviru katerih se opravi predikcija. Obstajata dve različni predikciji:

- **Intra-predikcija** je predikcija znotraj istega okvirja videa in izkorišča podobnost med podatki v istem okvirju.
- **Inter-predikcija** je predikcija, ki upošteva zaporedne okvirje in s tem kompenzira gibanje.

Po dokončanem postopku predikcije se residualna slika kodira s transformacijskimi postopki. Po preslikavi v frekvenčni prostor se izvede utežena kvantizacija koeficientov, pri čemer se z utežmi določi stopnjo kompresije. Kvantizaciji sledi entropijsko kodiranje, ki končno velikost še dodatno zmanjša. Izhod iz kvantizatorja se dekvantizira in izvede se inverzna transformacija, ki da nazaj sliko, ki se prišteje rezultatu predikcije. Na celotnem okvirju se izvede post-procesiranje, ki omili blokovne artefakte.

3.2 Kodiranje HEVC

Makroblok v HEVC-u je predstavljen kot enota kodnega drevesa (angl. *Coding Tree Unit*) (CTU). Velikost CTU je lahko 16×16, 32×32 ali 64×64 pikslov. Večji CTU pripomore k učinkovitejšemu deljenju makroblokov pri videih visoke resolucije (4K in več). Večji bloki prinašajo deljenje CTU na 4 manjše bloke imenovane kodne enote (angl. *Coding Unit*) (CU). Vsak CU pa vsebuje še predikcijsko enoto (angl. *Prediction Unit*) (PU) ter transformacijsko enoto (angl. *Transform Unit*) (TU). CU se lahko nadaljnje deli na TU enote do najmanjše dovoljene vrednosti. V PU enotah so shranjeni podatki intra- in inter-predikcije (npr. vektorji premika). Deljenje CU na PU se lahko izvede na različne načine.

Intra-predikcija v HEVC-u se vedno izvaja na kvadratnih matrikah (blokkih). Predikcija uporablja podatke sosednjih pikslov v naslednjih smereh: levo, levo-gor, levo-dol, desno-gor, gor (smeri nad antidiagonalo); definiranih s PU deljenjem. HEVC ponuja 35 različnih načinov intra-predikcije: način št. 0, poimenovan tudi planarni (angl. *planar*), način št. 1 imenovan tudi DC (angl. *Direct Current*) ter še preostalih

33 kotnih načinov (načini od št. 2 do št. 34). DC način je najbolj preprost, saj so v njem vsi piksli v enem PU nastavljeni na povprečno vrednost sosednjih pikslov, medtem ko je planarni način najbolj procesorsko zahteven, saj se uporablja dvodimenzionalna linearna interpolacija pikslov. DC na robnih pikslih izvaja dodatno linearno filtriranje. Kotni načini pa so linearne interpolacije v različnih smereh glede na kot zasuka.

Za transformacijo se uporablja hitri algoritem za izračun celoštevilskih približkov diskretne kosinusne in sinusne transformacije (DCT tipa II in DST tipa VI).

Pri entropijskem kodiranju se uporablja aritmetični kodirnik imenovan sintaktični kontekstualno-adaptivni binarni aritmetični kodirnik (*angl. syntax-based context-adaptive binary arithmetic coder*) (SBAC), ki omogoča paralelno kodiranje in dekodiranje in se prilagaja znotraj samega okvirju.

Post-procesiranje pri HEVC-u se izvede v dveh korakih (oba koraka sta opcijaska). Prvi korak DBF (*angl. Deblocking Filter*) odpravlja blokovne artefakte. Drugi korak SAO (*angl. Sample Adaptive Offset*) kompenzira splošno izgubo na dekodirani CTU enoti [1].

Za inter-predikcijo se uporabljata dva referenčna seznama L0 in L1, vsak seznam lahko vsebuje 16 referenčnih okvirjev, od katerih je lahko največ 8 slik unikatnih. To pomeni, da se za predikcijo lahko uporabi isto sliko dvakrat, vendar z različnimi utežmi. Za predikcijo vektorja premika se gradi seznam kandidatov, za vsako predikcijo pa se zapiše indeks kandidata. Za predikcijo vektorjev premika se lahko uporabi dva načina: združitev ali napredna predikcija vektorjev premika. Za vsak PU se določi kateri način predikcije je uporabljen. Vektorji premika so definirani za področja 16×16 pikslov, kar pomeni, da je potrebno pri dekodiranju imeti v medpomnilniku matriko 16×16 [2].

3.3 Kodiranje VP9

Pri VP9 so makrobloki imenovani superbloki, ti so velikosti 64×64 pikslov. Deljenje blokov je zelo podobno kot pri HEVC-u. Superblok se lahko razdeli do blokov velikosti 4×4 piksele. Deljenje poteka ravno tako na predikcijske in transformacijske bloke. Za razliko od HEVC-a pa dovoljuje tudi pravokotne bloke za intra-predikcijo, saj se dva pravokotna bloka lahko smatra kot en kvadratni blok. Tako imata sosednja pravokotna bloka isti način predikcije. Za predikcijsko deljenje je mogoče uporabiti deljenja 2N×2N, N×N, 2N×N in N×2N. Predikcija na pod-blokih velikosti 4×4 je determinirana z

blokom 8×8 v katerega spadajo. Predikcija na večjih pod-blokih je determinirana posamično za vsak pod-blok.

Za intra-predikcijo se uporabljajo sosednji piksli v naslednjih smereh: levo, levo-gor, desno-gor in gor. VP9 za razliko od HEVC-a ponuja le 10 načinov intra-predikcije: DC, True Motion in osem kotnih načinov (horizontalni, vertikalni, koti 207°, 153°, 117°, 63° in 45°. Način DC je enak načinu kodeka HEVC, le brez dodatnega filtriranja robnih pikslov.

Pri kvantizaciji se uporablja celoštevilski približek DCT tipa II in DST tipa II za vse velikosti blokov (32×32, 16×16 in 8×8), razen za bloke 4×4 se uporablja DST tipa VI. Poleg DCT in DST pa se uporablja tudi asimetrična diskretna sinusna transformacija (ADST).

Za entropijsko kodiranje se uporablja 8-bitno aritmetično kodiranje imenovano "bool-coder". Za razliko od HEVC-a se verjetnost kodiranja ne prilagaja znotraj okvirja, ampak so spremembe zapisane v glavi okvirja kot logične vrednosti (kontekst okvirja).

Post-procesiranje je sestavljeno iz enega samega koraka za odpravljanje artefaktov, ki filtrira na mejah od 8 do 2 piksla na vsaki strani. Pri filtriranju sodeluje tudi detektor, ki zaznava enakomernost območja pikslov in s tem prilagaja velikost filtra.

Inter-predikcija pri VP9 uporablja kompenzacijo gibanja na 8 pikslih. V osnovi se uporabi en vektor premika za en blok, opcijsko pa se lahko dovoli tudi predikcijo z dvema vektorjema premika, ki ju potem povprečimo. Za vsak blok ali okvir se lahko uporabi enega od treh filtrov:

- navadni – Lagrangianov interpolacijski filter,
- ostri – interpolacijski filter, ki bazira na DCT in se uporablja okrog ostrih robov ter
- mehki – ne-interpolacijski filter [1].

Vektor premika kaže na enega izmed treh možnih referenčnih okvirjev, poimenovanih "Last", "Golden" in "AltRef". Vektorji premika so napovedani iz urejenega seznama kandidatov referenčnih vektorjev premika. Kandidati so sestavljeni iz do 8 sosednjih blokov iz iste referenčne slike. Sledijo še časovni prediktorji iz prejšnjega okvirja. Če se pri iskanju kandidatov seznam ne napolni, se izvede ponovno iskanje brez ujemanja z referenčno sliko. Če seznam še vedno ni poln, se napolni z vektorji brez premika (0, 0). V seznamu kandidatov je vedno 8 referenčnih slik, od teh so tri slike uporabljene za intra-predikcijo. Seznam kandidatov se posodablja sproti [2].

Tabela 1: Seznam videoposnetkov uporabljenih v praktični primerjavi ter njihove lastnosti

Ime videa	Ločljivost	Št. sličic na sekundo	Število okvirjev	Bitna hitrost	Format in barvno podvzorčenje	Velikost datoteke
Park Joy	1280×720	50	500	1106Mb/s	YUV 4:4:4	1,28GB
FourPeople	1280×720	60	601	664Mb/s	YUV 4:2:0	792Mb
Life	1920×1080	30	825	747Mb/s	YUV 4:2:0	2,38GB
Shields	1280×720	50	504	553Mb/s	YUV 4:2:0	664MB

4 Primerjava kodiranja videov s HEVC in VP9 kodekom

Za primerjavo kodeka HEVC in VP9 je bil izveden praktični eksperiment kodiranja različnih videov. Za primerjavo so bili iz arhiva nekomprimiranih posnetkov <https://media.xiph.org/video/derf/> izbrani videoposnetki, ki jih podaja Tabela 1.

Opisi posnetkov:

- Na posnetku "Park Joy" gre za kamero ki se pomika v desno smer, tako da se ozadje premika z gibanjem kamere.
- Na posnetku "FourPeople" gre za statično kamero, ki je usmerjena na mizo, za katero sedijo štiri osebe, ki si podajajo dokumente.
- Posnetek "Life" je edini računalniško generiran posnetek – gre za animacijo več statičnih scen, v katerih se kamera pomika v polkrožni smeri.
- Posnetek "Shields" je precej podoben posnetku "Park Joy", le da se kamera giba v levo smer.

Za izvedbo kodiranja je bilo uporabljeno orodje FFmpeg verzije 20200403-52523b6, ki lahko kodira s kodekom HEVC s pomočjo knjižnice libx265 [3] in s kodekom VP9 s knjižnico libvpx-vp9.

Posamezni posnetek je bil kodiran s ciljnim bitnimi hitrostmi 2 Mb/s, 5 Mb/s, 10 Mb/s, 25 Mb/s, 75 Mb/s in 150 Mb/s. FFmpeg za kodiranje z VP9 z determinirano bitno hitrostjo priporoča dvopasovno kodiranje [4], zato je bilo to izbrano tudi za kodiranje s kodekom HEVC. Vsa kodiranja so bila izvedena na prenosnem računalniku Asus VivoBook S510U s procesorjem Intel® Core™ i7-8550U 1.8GHz z integrirano grafično kartico Intel® UHD Graphics 620, 8GB RAM in operacijskim sistemom Windows 10 Home 64-bit.

Za primerjavo rezultatov kodiranja se je za vsak video in ciljno bitno hitrost primerjal čas trajanja kodiranja, in ocenjevala se je kvaliteta videa z metrikami "Peak Signal to Noise Ratio" (PSNR) in "Structural Similarity Index" (SSIM). PSNR je razmerje med vrhovi signala in šuma. Izračunan je na podlagi srednje kvadratne napake (*angl. Mean Square Error*). Ker PSNR ne daje najboljših rezultatov glede na človeško percepcijo je dobro upoštevati še SSIM. SSIM je metrika, ki upošteva lokalno luminanco, kontrast in strukturo za določanje kvalitete [5]. Oba parametra sta podana v dB.

Ker se za primerjavo rezultatov uporabljajo PSNR in SSIM, je potrebno pri kodiranju upoštevati nastavitve posebej za PSNR in posebej za SSIM. Rezultati so bili izračunani z orodjem FFmpeg. Ker ta izračuna PSNR in SSIM vrednosti za vsak okvir, je bilo pri rezultatih vzeto povprečje vseh okvirjev.

5 Rezultati

5.1 Rezultati in primerjava procesa kodiranja

Tabela 2 predstavlja povprečni čas, potreben za kodiranje videoposnetka za posamezne ciljne bitne hitrosti. Opazno je, da je kodiranje s HEVC-om časovno bolj učinkovito kot z VP9. Razlika v učinkovitosti je v razponu od 2x pa tudi do 5x in več. Razlog za tako veliko razliko je tudi procesor, ki je bil uporabljen za kodiranje. Intelovi procesorji osme generacije so bolj optimizirani za kodiranje s kodekom HEVC in uporabljajo več virov pri kodiranju. Kodiranje s kodekom HEVC zna izkoristiti od 2x do 3x več procesorske moči v primerjavi z VP9. Izvedena je bila tudi primerjava zahtevnosti dekodiranja. Kodek HEVC potrebuje približno 10 % več procesorske moči kot VP9 za dekodiranje istega posnetka pri isti bitni hitrosti. Zaključiti se lahko, da je HEVC učinkovitejši pri kodiranju, a manj učinkovit pri dekodiranju.

5.2 Rezultati in primerjava objektivno merjene kvalitete videoposnetkov

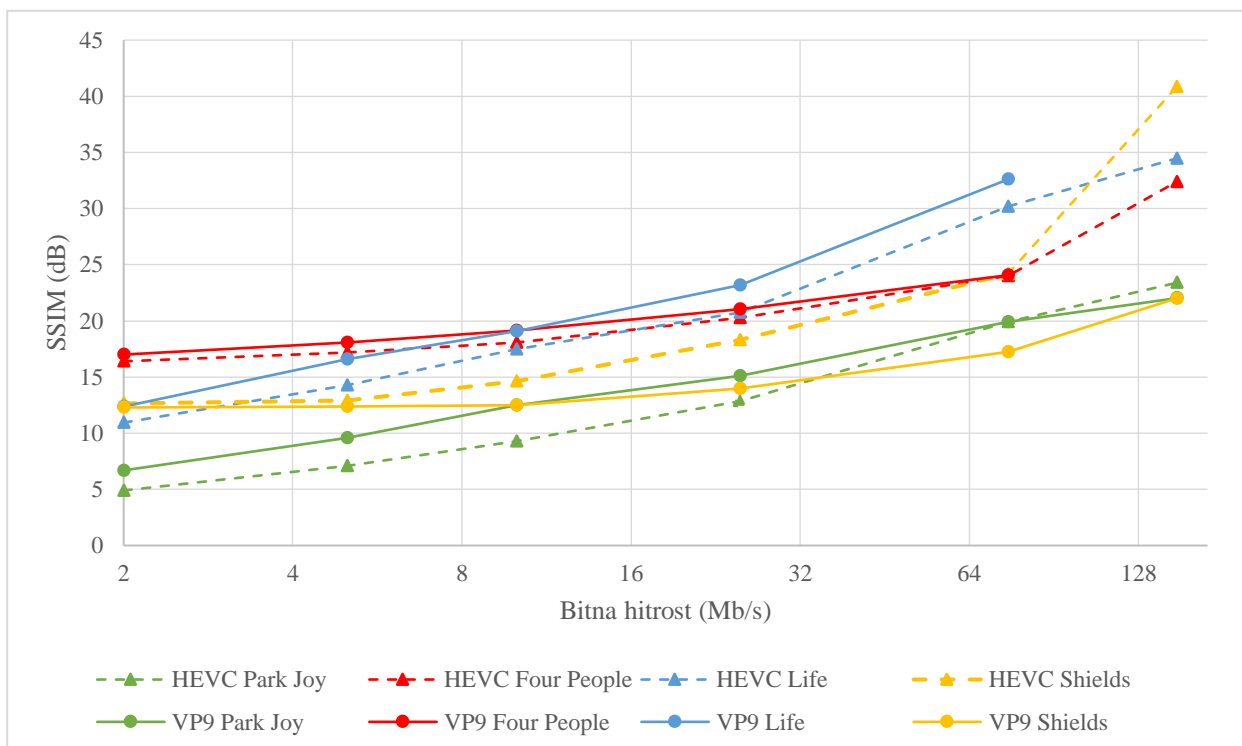
Sledi primerjava objektivno določene kvalitete videoposnetkov. Rezultati so bili pridobljeni tako za PSNR in SSIM, a ob podobnem poteku Slika 1 predstavlja le potek SSIM. Na splošno vsi rezultati kažejo, da sta si kodeka HEVC in VP9 pri isti bitni hitrosti dokaj primerljiva. Glede na posnetek "Park Joy" velja, da daje SSIM pri nižjih bitnih hitrostih boljše rezultate za kodek VP9, medtem ko je pri višjih bitnih hitrostih malenkost boljši HEVC. Medtem ko je pri posnetkih "Four People" in "Life" konstanto boljši VP9. Pri posnetku "Life" je to še bolj izrazito, zato se lahko sklepa, da je VP9 bolj optimiziran za računalniško generirane posnetke, kjer ni toliko šuma, kot pri resničnih posnetkih. Predvideva se lahko tudi, da se VP9 bolje obnese na posnetkih višje ločljivosti. Zanimiva je tudi razlika v rezultatih posnetka "Shields", kjer se HEVC izkaže za boljšega pri vseh bitnih hitrostih, iz tega primera lahko sklepamo, da je predikcija gibanja enega objekta pri HEVC-u precej učinkovitejša kot predikcija gibanja več objektov.

Če se dobljene rezultate primerja z ostalimi raziskavami [1, 5, 6, 7] in poljudnimi eksperimenti [8] je opaziti, da rezultati pridobljeni v letu 2013 [1] kažejo na to, da daje HEVC konstanto boljše rezultate – videoposnetki so pri isti bitni hitrosti višje kvalitete. Raziskavi iz leta 2016 [5, 6] sta že nekoliko drugačni od tiste iz 2013, tukaj so rezultati kodiranja s HEVC in VP9 zelo primerljivi. Novejši rezultati iz leta 2018 [8] in 2019 [7] pa so podobni rezultatom tega prispevka, kjer je VP9 boljši pri nizkih pasovnih širinah. Da se sklepati, da je VP9 v času od njegovega izida do danes prejel več izboljšav kot HEVC. Razlog je lahko tudi zaprtost kodeka HEVC, zaradi katerega je FFmpeg knjižnico za njegovo uporabo težje vzdrževati.

Tabela 2: Čas kodiranja s kodekoma HEVC in VP9 za različne ciljne bitne hitrosti – povprečje vseh videov

Ciljna bitna hitrost	150 Mb/s	75 Mb/s	25 Mb/s	10 Mb/s	5 Mb/s	2 Mb/s
Čas kodiranja HEVC	256 s	195 s	114 s	72 s	53 s	40 s
Čas kodiranja VP9	704 s	678 s	437 s	330 s	274 s	193s

Slika 1: Potek odvisnosti SSIM od bitne hitrosti za posnetke kodirane s kodekoma HEVC in VP9.



6 Zaključek

Glede na dobljene rezultate se zaključí, da pri isti bitni hitrosti kodek VP9 dosega boljše rezultate, vendar je pri uporabi potrebno upoštevati tudi zahtevnost kodiranja, saj traja kodiranje z VP9 precej dlje kot s HEVC-om. Odločitev za izbor je odvisna od uporabe, kje se bo video kodiral in kje predvajal.

Ker so bili v prispevku primerjani samo štirje različni videi, se težko sklepa, kateri kodek je boljši za katero vsebino ter kakšno ločljivost. Za tako primerjavo bi bilo potrebno primerjati večje število videoposnetkov, kjer bi več posnetkov imelo podobno vsebino. Rezultate posnetkov s podobno vsebino bi se združili v skupino in bi nato bile primerjane skupine med seboj.

Glede na to, da je od leta 2013, ko sta bila izdana kodeka minilo že kar nekaj let, bi bilo koristno v primerjavo vključiti tudi naslednika VP9, to je AV1. V prihodnosti pa je tudi VVC, naslednik HEVC-a. Oba novejša kodeka obljubljata še večje pridobitve pri videih visoke ločljivosti 4K in 8K.

Literatura

- [1] M. P. Sharabayko, O. G. Ponomarev, R. I. Chernyak: Intra Compression Efficiency in VP9 and HEVC; Applied Mathematical Sciences, Vol. 7, 2013, no. 137, 6803 – 6824, HIKARI 2013.
- [2] S. Akramullah: Digital Video Concepts, Methods and Metrics, Apres open 2014.
- [3] H.265/HEVC Video Encoding Guide, FFmpeg, dostopno na

<https://trac.ffmpeg.org/wiki/Encode/H.265> [Poskus dostopa 18.4.2020].

- [4] FFmpeg and VP9 encoding guide, FFmpeg, dostopno na <https://trac.ffmpeg.org/wiki/Encode/VP9> [Poskus dostopa 18.4.2020]-
- [5] J. D. Cock, A. Mavlankar, A. Moorthy, A. Aaron: A Large-Scale Video Codec Comparison of x264, x265 and libvpx for Practical VOD Applications, 2016, Netflix - Video Algorithms
- [6] J. Kufa, L. Polak, T. Kratochvil: HEVC/H.265 vs. VP9 for Full HD and UHD Video: Is There Any Difference in QoE?; 58th International Symposium ELMAR-2016, Zadar Croatia, IEEE 2016.
- [7] Z. Li, Z. Duanmu, W. Liu, Z. Wang: AVC, HEVC, VP9, AVS2 or AV1? — A Comparative Study of State-of-the-Art Video Encoders on 4K Videos; Springer Nature Switzerland AG 2019, Springer 2019.
- [8] AV1 vs VP9 vs AVC (h.264) vs HEVC (h.265) - 4 part comparison, Texpion, dostopno na <https://www.texpion.com/2018/07/av1-vs-vp9-vs-avc-h264-vs-hevc-h265-1-lossless.html> [Poskus dostopa 19.4.2020]
- [9] Netflix Now Streaming AV1 on Android, Netflix Technology Blog, Medium 2020, dostopno na <https://netflixtechblog.com/netflix-now-streaming-av1-on-android-d5264a515202> [Poskus dostopa 19.4.2020]