

Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra radioelektroniky

Platforma pro analýzu adaptivních metod online distribuce videosekvencí

Bc. Petr Hodač

Školitel: doc. Ing. Stanislav Vítek, Ph.D.

Obor: Audiovizuální technika a zpracování signálů

Zaměření: Elektronika a komunikace

Srpen 2020

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Hodač** Jméno: **Petr** Osobní číslo: **355362**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra radioelektroniky**
Studijní program: **Elektronika a komunikace**
Specializace: **Audiovizuální technika a zpracování signálů**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Platforma pro analýzu adaptivních metod online distribuce videosekvencí

Název diplomové práce anglicky:

Platform for Analysis of Adaptive Methods of Online Video Distribution

Pokyny pro vypracování:

- 1) Prostudujte metody objektivního testování kvality videosekvencí. Zaměřte se na systémy distribuce videosekvencí s proměnlivým bitovým tokem.
- 2) Na základě studie vytvořené v bodě (1) navrhnete platformu, která umožní kódování videosekvencí různými kodéry pro účely objektivní analýzy adaptivních metod. Platforma umožní uživateli automatizované nastavení vybraného kodéru na základě stanovených kritérií a aplikaci vybraných objektivních metod hodnocení kvality.
- 3) V rámci práce připravte testovací videosekvence pro ověření funkcionality platformy.

Seznam doporučené literatury:

- [1] DUANMU, Zhengfang; REHMAN, Abdul; WANG, Zhou. A quality-of-experience database for adaptive video streaming. IEEE Transactions on Broadcasting, 2018, 64.2: 474-487.
- [2] WIEN, Mathias. High efficiency video coding. Coding Tools and specification, 2015, 133-160.
- [3] ZHANG, Zhewei, et al. A new rate control scheme for video coding based on region of interest. IEEE Access, 2017, 5: 13677-13688.

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Ing. Stanislav Vítek, Ph.D., katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **17.09.2019**

Termín odevzdání diplomové práce: **14.08.2020**

Platnost zadání diplomové práce: **19.02.2021**

doc. Ing. Stanislav Vítek, Ph.D.
podpis vedoucí(ho) práce

doc. Ing. Josef Dobeš, CSc.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Za odborné vedení mé diplomové práce, velkou míru trpělivosti a ochoty, rychlost, lidský přístup a také za cenné a velmi podnětné rady při zpracovávání práce děkuji vedoucímu práce doc. Ing. Stanislavu Vítkovi, Ph.D.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 1. srpna 2020

Abstrakt

Cílem této diplomové práce je navrhnout a implementovat testovací platformu pro experimenty s adaptivním streamingem. Platforma umožňuje uživateli renderovat adaptivně kódované videosekvence s ohledem na nastavení jednoho z klíčových parametrů a současně automaticky vyhodnocuje vybrané metriky objektivního hodnocení kvality obrazu. V rámci práce byla ověřena funkčnost celé platformy na vybraných videosekvencích.

Klíčová slova: Stream, Adaptivní stream, Objektivní metody hodnocení kvality, Kodér, Kvalita obrazu

Školitel: doc. Ing. Stanislav Vítek, Ph.D.

Abstract

This diploma thesis aims to design and implement a test platform for experiments with adaptive streaming. The platform allows the user to render adaptively coded video sequences considering the settings of one of the crucial parameters and at the same time, automatically evaluates selected metrics of objective assessment of image quality. The thesis verifies the functionality of the entire platform on selected video sequences.

Keywords: Streaming, Adaptive Streaming, Objective Quality Assessment Methods, Codek, Picture Quality

Title translation: Platform for Analysis of Adaptive Methods of Online Video Distribution

Obsah

1 Úvod	1	3.4 Běh těchto softwarů	20
2 Distribuce audiovizuálního obsahu	3	4 Návrh systému pro analýzu adaptivního streamu	22
2.1 Techniky adaptivního streamování	8	4.1 Požadavky	22
2.2 Zdrojová komprese audiovizuálního obrazu	11	4.2 Návrh systému	23
3 Metody hodnocení kvality obrazu	14	5 Nástroje	27
3.1 Subjektivní hodnocení kvality obrazu	14	5.1 Vytvoření adaptivního streamu .	27
3.2 Objektivní hodnocení kvality obrazu	15	5.2 Prezentace výsledků	28
3.3 Nástroje pro objektivní hodnocení kvality obrazu	18	5.3 Testování	29
3.3.1 Video Quality Estimator	18	5.4 Užitečné příkazy pro práci s videosekvencemi	33
3.3.2 FFmpeg	19	6 Experimenty	35
3.3.3 MSU Video Quality Measurement Tool	20	6.1 Výukové video	35
		6.2 Vliv délky segmentu na výslednou kvalitu videa	39
		6.3 Vliv zaostření scény na výslednou kvalitu videosekvence	42

7 Závěr	45
Literatura	47
A Kompilace použitých softwarů	50
B Seznam zkratek	52
C Obsah přiloženého média	55

Obrázky

2.1 Vysílací řetězec České televize [18]	4	4.1 Renderování všech nastavení kodéru	24
2.2 Unicast vysílání	4	4.2 Skládání různých částí do finálního streamu	24
2.3 Video on Demand	5	4.3 Výpočet metrik	25
2.4 RTMP stream	6	4.4 Postupné renderování a porovnávání	25
2.5 Adaptivní stream [25]	7	4.5 Výpočet metrik	26
2.6 YouTube nastavení kvality [12] . .	8	5.1 PSNR pro různá nastavení bufferu	29
2.7 Playlisty HLS [27]	9	5.2 SSIM pro různá nastavení bufferu	30
2.8 MPEG-DASH [26]	9	5.3 VIF pro různá nastavení bufferu	30
2.9 Přehled využívání daných technologií [19]	10	5.4 PSNR pro různá nastavení bufferu	31
2.10 Group of Pictures [16]	11	5.5 SSIM pro různá nastavení bufferu	31
2.11 YouTube VP9 [17]	13	5.6 VIF pro různá nastavení bufferu	32
3.1 Plná reference při vyhodnocování [7]	15	5.7 Grafování v prostředí MATLAB	34
3.2 Schéma kombinace SSIM [7]	16	6.1 Sestava pro natáčení experimentu	36
		6.2 PSNR pro různá nastavení bufferu	37

6.3 SSIM pro různá nastavení bufferu	37
6.5 Problémový přechod snímků 2200	38
6.6 Problémový přechod snímků 3100	38
6.4 VMAF pro různá nastavení bufferu	38
6.7 Sestava pro natáčení experimentu	40
6.8 PSNR pro VMAF 70 pro segment 1, 3, 5, 10 s	40
6.9 SSIM pro VMAF 70 pro segment 1, 3, 5, 10 s	41
6.10 VIFP pro VMAF 70 pro segment 1, 3, 5, 10 s	41
6.11 Sestava pro natáčení experimentu	42
6.12 Vliv zaostření na kvalitu scény s min VMAF 80 na PSNR.....	43
6.13 Vliv zaostření na kvalitu scény s min VMAF 80 na SSIM	44
6.14 Vliv zaostření na kvalitu scény s min VMAF 80 na VIF.....	44

Tabulky

2.1 Doporučené přenosové rychlosti h264	6
3.1 Bitové rychlosti pro 50p RAW ..	21



Kapitola 1

Úvod

Distribuce streamovaného audiovizuálního obsahu uživatelům se stala v posledních letech velkou částí celkového objemu přenesených dat na internetu. Dle studie Sandvine [1] v roce 2019 bylo 60 % internetového provozu právě streamování. Zvláště k objemu přenesených dat přispívá zvyšující se kvalita přenášených obrazových dat. Mohlo by se zdát, že sledování videí nebo záznamů na největších platformách jako Twitch, YouTube a Facebook není streamování, ale je tomu tak – jedná se o adaptivní streaming. Jde o velké množství přenesených dat, ve kterém se bude čím dál více zvyšovat kvalita obsahu, a to nejen v rozlišení, ale i v dynamickém rozsahu scény a také snímkové frekvenci.

Další rychle rostoucí segment distribuce video obsahu ukázala krizová situace související s pandemií nemoci covid-19, kdy produkovalo a konzumovalo výrazně více osob prostřednictvím konferenčních hovorů a také vzdálené výuky na školách. Společnost Netflix musela v té době omezit bitový tok o 25 % [20]. Aktuálně to již takový problém není, ale lze předpokládat, že bezkontaktní a odosobněný trend se bude dále navyšovat a také se budou postupně stupňovat nároky uživatelů na kvalitu audiovizuálního obsahu dodávaného v reálném čase. To vše se projeví na vytížení internetových linek a firmy zajisté začnou nabízet hardware a služby zaměřené na virtuální propojování lidí více než

doposud.

Tento trend bude mít pozitivní dopady nejen na dopravu a věci s ní spojené, ale také na celkový čas uživatelů těchto služeb. Aby byl dopad na datové spoje co nejmenší a výsledná kvalita co nejlepší, tak roste zájem o tuto problematiku. Práce se jí zabývá ve 2. kapitole

Analýza. Analýza audiovizuálního obsahu, který byl streamován v reálném čase, je velmi problematická, jelikož nejsou k dispozici zdrojová data a ani není zřejmé, jak byla tato data vytvořena. V případě opakovaného streamování daného videa může být situace o trochu lepší, protože zdrojové video existuje. Nicméně neexistuje mnoho nástrojů, které by generovaly adaptivní stream přesně dle našich požadavků nebo potřeb. Této problematice se věnuje kapitola 3.

Stav poznání. Kvalitou diváckého zážitku se zabývá článek [3], v němž autoři tvoří databázi videí, která byla streamovaná pomocí platformy DASH. Další z článků [6] se navíc zabývá vyhodnocováním kvality diváckého zážitku přes platformu DASH i se srovnáním s výsledky subjektivních testů. Tato práce se ale věnuje platformě nezávislého systému a také si klade za cíl generovat vlastní sekvence splňující některé požadavky. Optimalizaci daného typu videa pro adaptivní streaming také najdeme v článku [5], kde je možné vidět, že je možné ušetřit přibližně 10 % bitového toku. Vlivem změny bitového toku v daném typu videa se zabývá společnost Netflix v článku na svém blogu [13]. Je jasně patrné, že je zapotřebí zabývat se různým nastavením kodéru pro různá videa. Existuje celá řada softwarů, které se profesionálně zabývají vyhodnocením kvality obrazu pro vyrenderované video a originál. Těmto softwarům se bude více věnovat kapitola 3. Bohužel většina z nich je velmi drahá a navíc neumožňuje generování videosekvencí dle daných parametrů zadaných uživatelem. Renderování videosekvencí a také výpočty objektivního hodnocení kvality trvají velmi dlouho. Také je k tomu třeba mnoho mechanické práce, a to jak při generování, tak vyhodnocování. Tato práce si tedy klade za cíl navrhnout platformu pro experimentování s různým nastavením adaptivního streamingu.

Kapitola 2

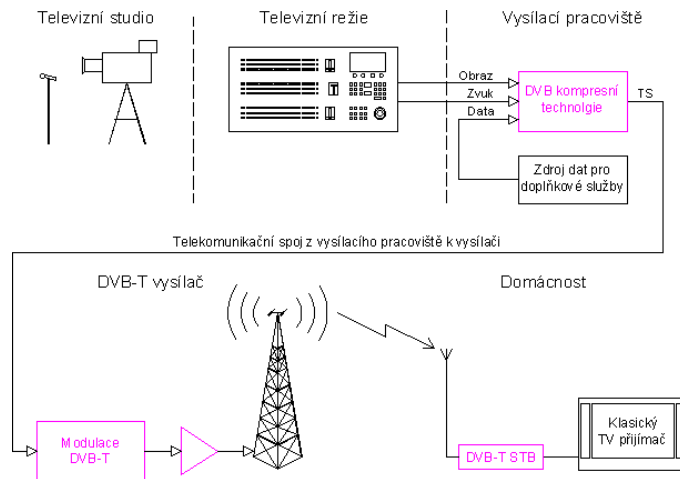
Distribuce audiovizuálního obsahu

V současnosti se audiovizuální obsah téměř vždy distribuuje pomocí digitálního vysílání. Na rozdíl od minulosti, kdy se vysílal obsah v analogové podobě. Způsob dopravy audiovizuálního díla k divákům je možné rozdělit podle druhu přenosu signálu na:

- Pozemní – signál se šíří pomocí pozemních vysílačů bezdrátově,
- Kabelový – signál se šíří pomocí fyzického L1 propoje,
- Satelitní – signál se šíří pomocí družic na oběžné dráze bezdrátově,
- IPTV – signál se šíří pomocí protokolu IP multicast v dané podsíti.

Vysílací řetězec. Všechna tato vysílání je možné zařadit do kategorie broadcast vysílání. Pro ně platí, že všichni diváci přijímají jeden signál, který je vysílán pouze jednou. Tento signál s různým zpožděním dorazí na přijímače diváků a je přehrán. Není tedy možné vracet se v čase. Někteří poskytovatelé IPTV nabízejí službu aktuálního vysílání a jednoho až dvou zpožděných vysílání téhož signálů (typicky o hodinu a dvě). Tento způsob ale zvyšuje

2. Distribuce audiovizuálního obsahu



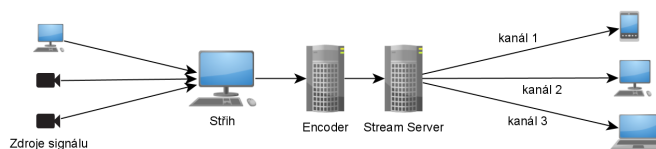
Obrázek 2.1: Vysílací řetězec České televize [18]

konstantní multicastový datový tok trojnásobně, což může být u většího počtu kanálů problém. Také to vyžaduje rychlost připojení mezi 100 a 1 000 Mb/s.

Druhým typem přenosu je:

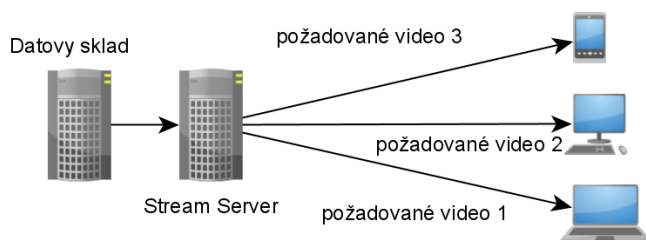
- Internetová televize – signál se šíří pomocí sítě internet každému divákovi zvlášť.

Tuto metodu nazýváme unicast. V této metodě se přenáší každému uživateli data zvlášť, což zatěžuje datovou linku. V případě jejího sdílení s ostatními uživateli může omezovat ostatní i našeho diváka, pokud klesne rychlost připojení pod datový tok videa – tedy video není možné přehrát.



Obrázek 2.2: Unicast vysílání

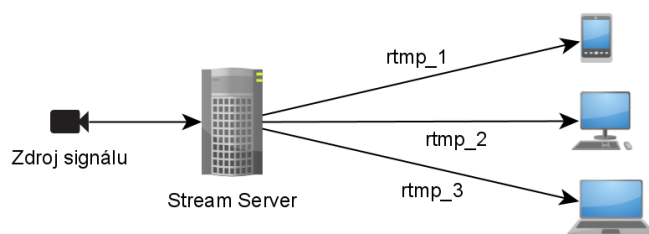
Video on Demand. Tato metoda distribuce audiovizuálního obsahu využívá unicastový přenos. Mezi její výhody patří, že každý divák může zhlédnout dané vysílání kdykoliv. Stream server má tyto pořady uloženy ve vyrovnávací paměti nebo v datovém skladu a poskytuje je divákům, když o ně mají zájem. Tato metoda je datově náročná, protože celkový datový tok je součet datových toků všech streamů. To může být problém v distribuční cestě k divákovi a čím blíž k němu to je, tím větší nastane problém, protože dojde k jak zahlcení sítě, tak k přetížení serverů. Tento problém se řeší pomocí rychlejších síťových karet v serverech či zaváděním vyššího počtu streamovacích serverů, mezi které se zátěž rozděluje.



Obrázek 2.3: Video on Demand

Stream. Streamem rozumíme unicastový přenos digitální videosekvence k divákovi. Jeho kvalita je hlavně daná bitovým tokem a rozlišením.

Real-Time Messaging Protocol – RTMP. Tento velmi rozšířený protokol pro zasílání zpráv v reálném čase je navržen na streamování audiovizuálních dat s nízkou latencí. Používá potvrzování zasílání zpráv pomocí TCP z IP protokolu. Existují také varianty, které jsou tunelovány přes HTTP a nebo šifrovány pomocí SSL.



Obrázek 2.4: RTMP stream

Adaptivní stream. Při pohledu do tabulky přenosových rychlostí pro kodek h264, kde je uvedena hardwarová akcelerace dekódování, je zřejmé, že nelze vyhovět všem divákům daného streamu na různých zařízeních s různou rychlostí připojení k internetu. Proto je nutné, aby platformy pro streamování obsahu nabízely různé kvality daných streamů, které jsou rozděleny do krátkých, typicky 5s sekvencí. Mezi těmito sekvencemi je možné přepínat na základě rychlosti připojení nebo velikosti okna přehrávače. Tyto krátké sekvence jsou typicky bufferovány, což způsobuje zpoždění, ale zlepšuje divácký zážitek v případě změny rychlosti připojení nebo krátkodobého výpadku či rušení.

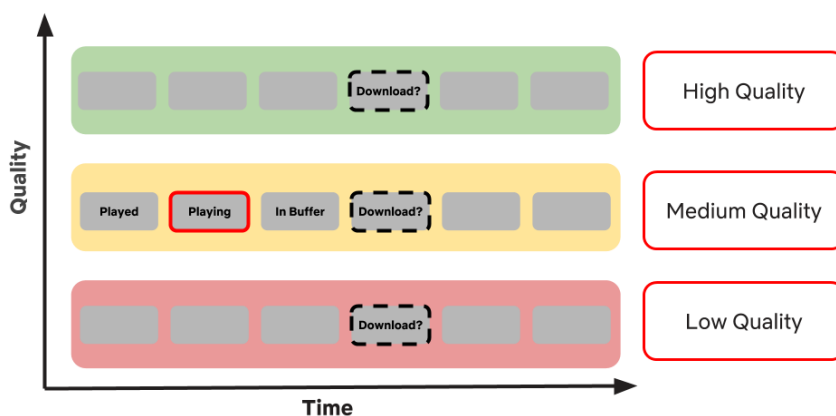
Kvalita adaptivního streamu. Stejně jako u obyčejného streamu je kvalita dána rozlišením a bitovým tokem, ale navíc záleží ještě na jednom parametru a tím je délka bloku. Tato hodnota určuje, jak rychle se stream adaptuje na lepší nebo horší kvalitu. Z uživatelského pohledu by měl tedy tento úsek být co nejkratší, ale z pohledu náročnosti na přenos a účinnost komprese je zase zajímavé mít blok delší.

Rozlišení	Přenosová rychlost	Přenosová rychlost
	fps (30)	fps (60)
2160p	35–45 Mb/s	53–68 Mb/s
1440p	16 Mb/s	24 Mb/s
1080p	8 Mb/s	12 Mb/s
720p	5 Mb/s	7,5 Mb/s
480p	2,5 Mb/s	4 Mb/s
360p	1 Mb/s	1,5 Mb/s

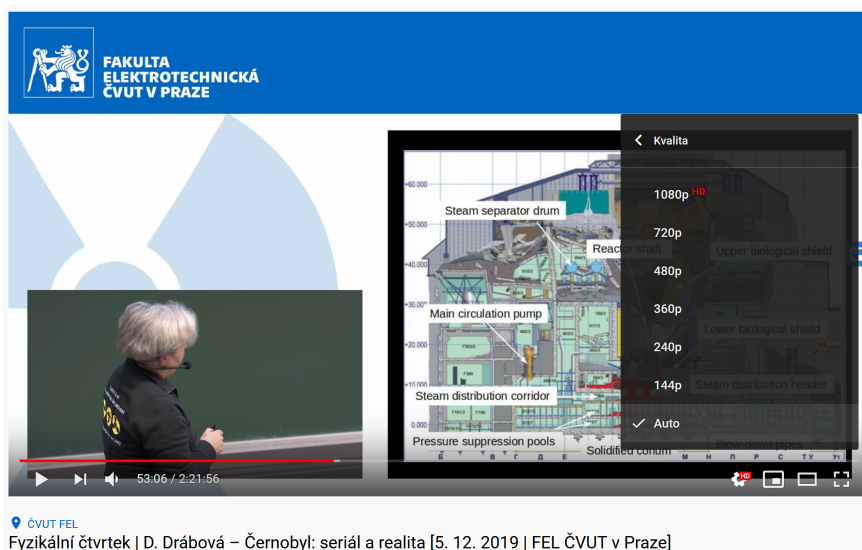
[2]

Tabulka 2.1: Doporučené přenosové rychlosti h264

Proč adaptivní? V televizní technice bývá přijímač na jednom místě a typicky se řeší dvě rozlišení – HD a Full HD. Pokud je dostatečný odstup signálu od šumu, daný kanál funguje, když není, tak nefunguje. Diváci na největších streamovacích platformách sledují videa nebo streamy na nejrůznějších zařízeních, především na těch mobilních, na kterých se při pohybu mění kvalita připojení k internetu. Adaptivní stream se správným přehrávačem umí na tyto změny reagovat pouze snížením kvality zážitku daného diváka, a to jen po dobu, kdy je to nutné. Pro uživatele s dostatečným připojením k internetu a se zobrazovačem o stejném rozlišení jako zdroj signálu nabízí plnou kvalitu, a tím i dobrý divácký zážitek. Některé služby ovšem nabízí vyšší kvalitu pouze za poplatek.



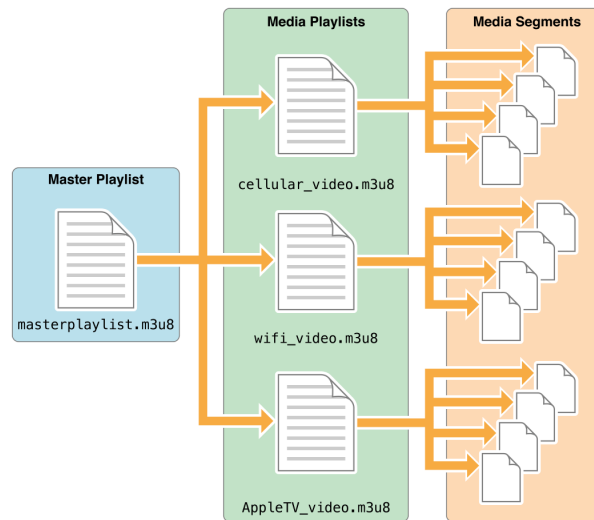
Obrázek 2.5: Adaptivní stream [25]



Obrázek 2.6: YouTube nastavení kvality [12]

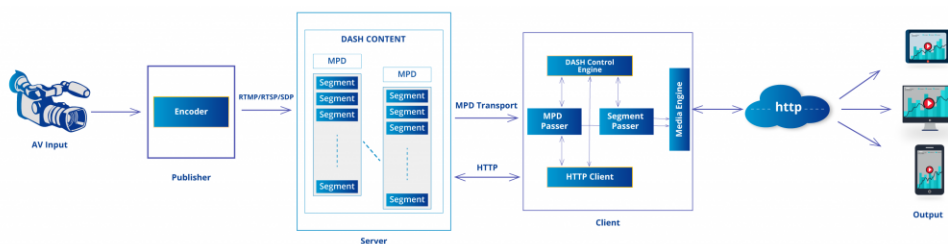
2.1 Techniky adaptivního streamování

Apple HTTP Live Streaming – HLS. HLS byl navržený již v roce 2009 společností Apple. Přes několik návrhů přešel do formy RFC [4]. Základem je zakódování videa do více rozlišení a rozdělení všech těchto rozlišení na několika sekundové části buď v MPEG-2 TS, nebo v MPEG4. Ty jsou pak vloženy do playlistu, ze kterého jsou přehrávány dle číselného pořadí. Typická délka segmentu je 10s.



Obrázek 2.7: Playlists HLS [27]

Dynamic Adaptive Streaming over HTTP – MPEG-DASH. Tato adaptivní technologie z dílny skupiny MPEG závisí na použitém kodeku, umožňuje tedy použít různé druhy jako h264, h265, VP9 a další. Tento typ se nazývá kodek agostnic. Výsledný stream se skládá z menších částí než u HLS, proto může lépe reagovat na změny. Bohužel není podporovaný v HTML5, ale existuje open-source implementace klienta, a to i v programovacím jazyce C++. Také ho nepodporují zařízení Apple.

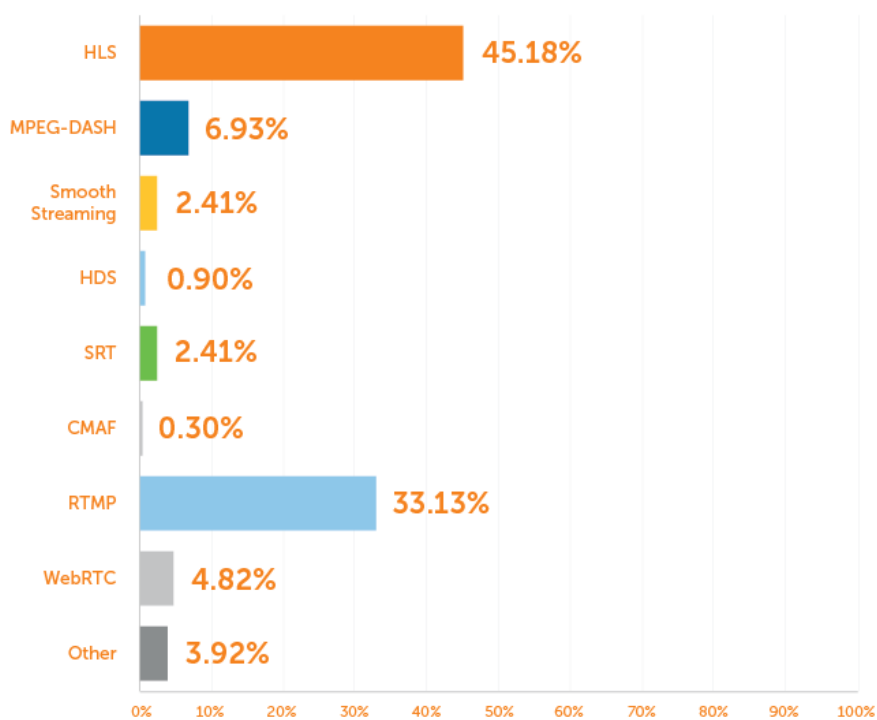


Obrázek 2.8: MPEG-DASH [26]

Adobe HTTP Dynamické streamování. Firma Adobe disponuje řešením pro adaptivní streaming, které je sice multiplatformní a podporuje kodeky

h264 a VP6, ale je třeba mít nainstalovaný Flash Player 10.1 a novější. Samotný vydavatel ale v roce 2020 plánuje ukončit podporu tohoto softwaru.

Smooth Streaming. Microsoft také přišel s technologií pro adaptivní streamování přes HTTP, a to jako rozšíření svého produktu IIS Media Services. Pro klientský příjem vyžaduje aplikaci SilverLight, která není multiplatformní, nicméně je možné omezené funkcionality dosáhnout s aplikací Smooth Streaming Porting Kit.



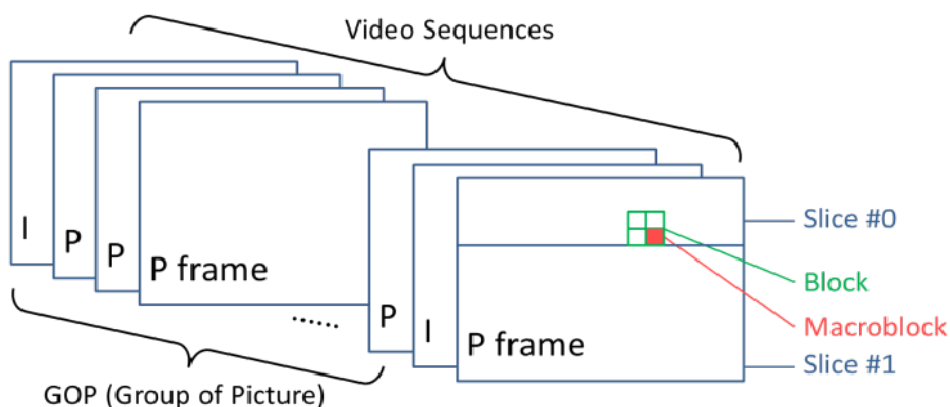
Obrázek 2.9: Přehled využívání daných technologií [19]

2.2 Zdrojová komprese audiovizuálního obrazu

Komprese audiovizuálního materiálu má za cíl redukovat redundanci a irelevanci. Jedná se tedy o odstranění duplicitních dat a informací, které nejsou postřehnutelné lidským okem. Takovéto kompresi se říká perceptuálně bezeztrátová. Není tedy poznat rozdíl mezi originálem a komprimovaným výsledkem.

MPEG-2. MPEG-2 je jeden z nejstarších kompresních algoritmů používaný v digitální televizní technice DVB-T a na DVD. Stejně jako komprese JPEG je založen na segmentaci bloku po 8×8 pixelech s použitím diskrétní kosínové transformace (DCT). Typický datový tok byl 12–18 Mb/s pro 1080p50 [22].

Group of Pictures – GOP. GOP je skupina po sobě jdoucích snímků typu I, B nebo P. Typicky se zadává pomocí hodnot M a N , kde M značí vzdálenost P snímků a N říká vzdálenost I snímků. Například tedy pro $M = 4$, $N = 12$ bude GOP vypadat následovně – IBBBPBBBBPBBBI. I je snímek, který je nezávislý na ostatních snímcích. P snímek je prediktivní, který je spočítán za základě predikce pohybu makrobloků za pomoci směrových vektorů z I snímku. B snímek je rozdílový mezi I a P snímkem.



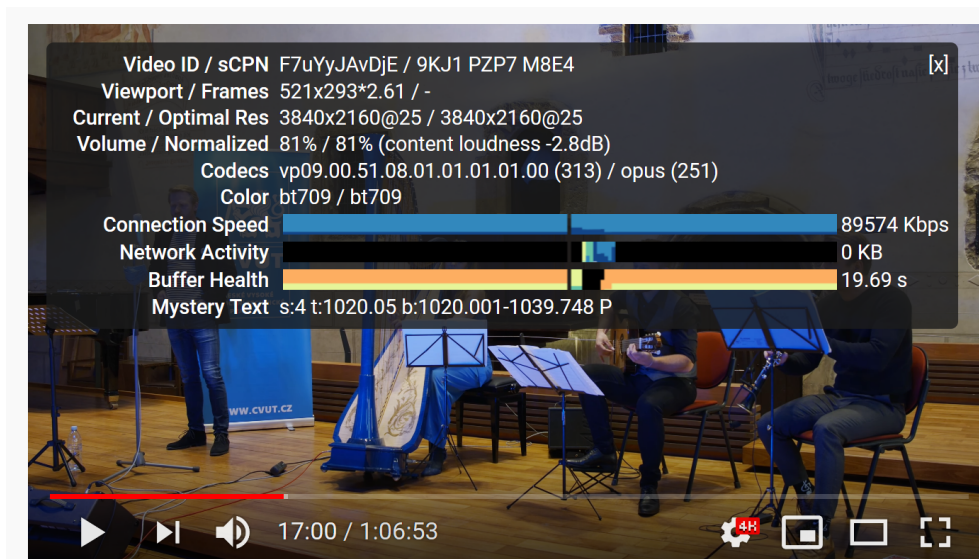
Obrázek 2.10: Group of Pictures [16]

h264 / MPEG-4 AVC. Je nejspíše nejrozšířenější kompresní algoritmus současnosti. Oproti svému předchůdci přináší stejnou obrazovou kvalitu s polovičním datovým tokem. Hlavním důvodem úspory bitového toku je prodloužení struktury GOP, vylepšení entropického kódování a 7 typů makrobloků, které mohou mít každý svůj vektor pohybu [15]. Hardwarová akcelerace výpočtu tohoto kodeku je běžnou součástí většiny procesorů a zařízení určených pro práci s videem.

h265 / HEVC. Tento kodek je nástupce kodeku h264. Jak je zvykem u MPEG standardů, tak opět přináší poloviční bitový tok při zachování obrazové kvality. Je relativně výpočetně náročný, nicméně jeho hardwarová akcelerace je v masivním měřítku levná záležitost, i když je licencovaný. Najdeme ho v DVB-T2 vysílání nebo jeho hardwarový encoder v zařízeních, co natáčí ve vyšších rozlišeních a jsou finančně limitovány zápisy na médium, jako například drony. Bohužel hardwarová akcelerace dekodingu na počítačích a běžných zařízeních je částečně možná, ale nefunguje rovnou.

VP8. VP8 je svobodný kodek od firmy Google, který je částečně srovnatelný s h264, nicméně se tak nerozšířil jako právě MPEG-4. Když se srovná s h264, tak VP8 je prakticky ve všech parametrech horší s drobnou výjimkou a tou je 4K rozlišení. U tohoto rozlišení má v bitovém toku nepatrně lepší výsledky [24].

VP9. Tento kodek je nástupce kodeku VP8 a konkuruje kodeku h265. Je možné s ním setkat právě na platformě YouTube. Velká výhoda tohoto formátu je, že má nativní podporu v HTML5.



Obrázek 2.11: YouTube VP9 [17]

AV1. Toto je aktuálně nejnovější kodek, který je založen na VP9 a je vyvíjen Aliancí pro otevřená média (OAMedia). Ta je složena z firem Amazon, Apple, ARM, Cisco, Facebook, Google, IBM, Intel Corporation, Microsoft, Mozilla, Netflix a NVIDIA. Je zaměřen právě na streamovaná média a dosahuje velmi dobrých výsledků. Bohužel je extrémně výpočetně náročný a hardwarová akcelerace ještě není hotova. V době psaní této práce dosahuje encode 4K klipu s délkou 1 min přibližně 40 min výpočtu. Ale tento kodek je cesta nezatížená licenčními poplatky a vypadá velmi slibně.

Kapitola 3

Metody hodnocení kvality obrazu

Kvalita výsledného vyrenderovaného videa je důležitou metrikou pro hodnocení kompresí audiovizuálních dat. Video bude totiž mnohonásobně viděno diváky a mnohdy, kdy po čase již nezůstanou velké zdrojové soubory, bude také to jediné, co nám po letech z daného videa zůstane. Proto při využití perceptuálně ztrátové komprese je vždy nutné dbát na její výslednou kvalitu.

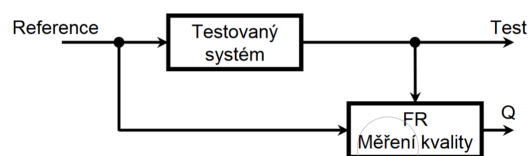
3.1 Subjektivní hodnocení kvality obrazu

Nejlepší metoda pro hodnocení kvality obrazu je bezesporu ta subjektivní, kterou provádí větší skupina pozorovatelů a která je správně vyhodnocena. Jelikož ale různí pozorovatelé mohou mít velmi odlišný pohled na kvalitu videa, je na správné vyhodnocení třeba použít vhodnou statistiku. Bohužel metoda subjektivního hodnocení je velmi drahá a časově náročná, proto se používají metody objektivního hodnocení kvality.

3.2 Objektívni hodnocení kvality obrazu

Metody objektívniho hodnocení kvality obrazu dávají jasné číselné výsledky, které lze porovnávat a ve kterých lze hledat korelace se subjektivními testy. Nejlepších korelací by měly bezesporu dosahovat ty, které reflektují lidské vidění a extrakci scény. Jedná se o matematické charakteristiky rozdílu mezi pixely nebo strukturální podobností.

Třídy metod hodnocení kvality obrazu. Pro vyhodnocení objektívniích metrik kvality lze používat plnou, částečnou nebo žádnou informaci o původním nezkrasleném snímku. U metod částečné reference hodnocení kvality je obtížné správně nastavit příznaky, proto bude lepší v této práci použít právě plnou referenci, kde lze očekávat nejlepší výsledky.



Obrázek 3.1: Plná reference při vyhodnocování [7]

Mean Squared Error – MSE. Tento parametr je střední kvadratická chyba mezi pixely originálního obrazu a zkrasleného obrazu [23].

$$MSE = \frac{1}{n} \frac{1}{m} \sum_{i=1}^n \sum_{j=1}^m [R(i, j) - D(i, j)]^2 \quad (3.1)$$

- $R(i, j)$ – referenční snímek
- $D(i, j)$ – zkraslený snímek
- $n \times m$ – rozlišení snímku
- b – bitová hloubka

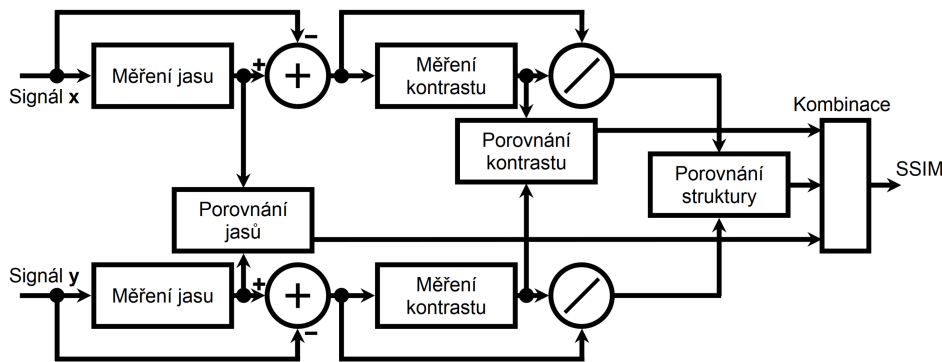
Signal to Noise ratio – SNR.

$$SNR_{dB} = 10 \log_{10} \left(\frac{\frac{1}{n} \frac{1}{m} \sum_{i=1}^n \sum_{j=1}^m R^2(i, j)}{MSE} \right) \quad (3.2)$$

Mean Squared Error – PSNR. PSNR je špičkový odstup signálu k šumu, tedy maximální energie signálu ku střední kvadratické chybě (MSE) [7]. Výsledné hodnoty jsou v decibelech a PSNR dvou stejných obrazů není definováno, jelikož dva totožné obrazy mají MSE rovno nule.

$$PSNR_{dB} = 20 \log_{10} \left(\frac{2^b - 1}{\sqrt{MSE}} \right) \quad (3.3)$$

Structural Similarity – SSIM. Metrika SSIM je výrazně přesnější v odhadu kvality než metoda PSNR. Porovnává místní vzory intenzit pixelů. Vzniká kombinací třech metrik.



Obrázek 3.2: Schéma kombinace SSIM [7]

Pro podobnost v jasů.

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (3.4)$$

$$\mu_x = \sum_{i=1}^n x_i \quad (3.5)$$

Pro podobnost v kontrastu.

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (3.6)$$

$$\sigma_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \mu_x)^2} \quad (3.7)$$

Pro podobnost ve struktuře.

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (3.8)$$

$$\sigma_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y)} \quad (3.9)$$

Výsledné SSIM je možné spočítat dle vzorce (3.10):

$$SSIM = l^\alpha(x, y)c^\beta(x, y)s^\gamma(x, y) \quad (3.10)$$

Visual Information Fidelity – VIF. Metoda VIF je založena na předpokladu, že hlavní funkcí lidského vidění je extrakce informace ze scény a že komprese snižuje přirozenost dané scény. Výsledná hodnota VIF je poměr vzájemné informace mezi zkreslenou a nezkraslenou větví.

$$VIF = \frac{\sum iI(C_i, F_i)}{\sum iI(C_i, F_i)} \quad (3.11)$$

Video Multi-Method Assessment Fusion – VMAF. Technika VMAF pro objektivní hodnocení kvality od firmy Netflix [11] je založena na korelaci výsledku měření s datasetem vygenerovaným pomocí neuronové sítě natrénované pomocí výsledku velkého množství subjektivních testů. Metoda velmi dobře koreluje s výsledky subjektivních testů. Tato metoda se používá na

videa s rozlišením 2160p nebo 1080p dle použitého datasetu a její výsledky jsou pro vzdálenosti 3H u 1080p a 1,5H u 2160p. U 4K rozlišení se jedná zatím pouze o predikci. Pro ostatní rozlišení je třeba pro správný výpočet video znovu vyrenderovat do rozlišení modelu. Tato metrika vychází z výsledků subjektivních testů na různých typech obsahů, kde diváci odpovídali „bad“, „poor“, „fair“, „good“ a „excellent“, kde hodnota 20 odpovídá „bad“ a pak roste po 20. Za dobrou můžeme považovat hodnotu 70. U této metriky se objevuje zajímavé vyhodnocení skóre jednotlivých snímků, a to harmonický průměr, který je stejně jako lidské vnímání citlivý na malé hodnoty kvality. Tyto malé hodnoty by se v obyčejném průměru (AM) ztratily, ale i pár nízkých hodnot dost ovlivní divácký zážitek.

Harmonický průměr – HM. Tento průměr je podíl počtu hodnot se součtem jejich převrácených hodnot [14].

$$H = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}} \quad (3.12)$$

$$AM(85, 90, 87, 91) = 88.25 \quad HM(85, 90, 87, 91) = 88.19$$

$$AM(90, 98, 1, 91) = 67.25 \quad HM(90, 87, 1, 91) = 3.87$$

3.3 Nástroje pro objektivní hodnocení kvality obrazu

3.3.1 Video Quality Estimator

Tento komerční software [21] je zaměřen nejen na vyhodnocování objektivních metrik kvality obrazu, ale i na zobrazování videí vedle sebe. Umí vyhodnotit PSNR, APSNR, MSAD, MSE, SSIM, VQM, VMAF a VMAF Phone či VIF.

Nicméně vytvořit v něm adaptivní sekvence není možné. Na webu nelze dohledat cenu za licenci a ani jejich podpora ji nechtěla sdělit.

3.3.2 FFmpeg

Tento velmi rozšířený multiplatformní software s otevřeným kódem¹ nejen pro renderování videa umí počítat pomocí filtrů dvě základní metriky – PSNR a SSIM [10]. S rozšířením od společnosti Netflix je možné také počítat skóre VMAF, typicky mezi originálem a vyrenderovaným videem².

Výpočet VMAF. Po kompilaci FFmpegu je možné tedy napočítat metriky VMAF, SSIM a PSNR. Stačí k tomu zdrojové a vyrenderované video, které ani nemusí být převedeno do YUV prostoru, protože převod zvládne FFmpeg při samotném výpočtu. Nastavení parametrů, které chceme vypočítat do sekce `libvmaf = název = hodnota` a oddělovačem parametrů, je:

- `phone model` – výchozí hodnota `false`,
- `psnr - ssim - msssim`,
- `pool - min`, harmonic mean výchozí hodnota je `mean`.

Listing 3.1: Ukázka výpočtu hodnot

```
ffmpeg \  
-i vysledek.mp4 \  
-i zdroj.mov \  
-lavfi libvmaf='psnr=1:pool=harmonic_mean' -f null -  
  
Start calculating VMAF score...  
frame= 274 fps=4.4 q=-0.0 Lsize=N/A time=00:00:11.00  
Exec FPS: 4.326957
```

¹<https://github.com/FFmpeg/FFmpeg>

²<https://github.com/Netflix/vmaf.git>

```
VMAF score (harmonic_mean) = 96.416817
PSNR score (harmonic_mean) = 44.530273
```

3.3.3 MSU Video Quality Measurement Tool

Tento komerční software [8] nabízí objektivní analýzu obrazových dat, a to jak pro full reference, tak pro single reference porovnávání. Jedná se o velmi názorný software pro hodnocení různě komprimovaných obrazových dat mezi sebou samými a originálem. Následně porovná výsledky mezi sebou rovnou s grafy a spustitelnými video sekvencemi a to vše dokáže exportovat do formátu JSON.

Nevýhody. Bohužel v bezplatné verzi umí pouze videa do SD kvality. Mezi další nevýhody patří, že grafická aplikace není multiplatformní a také nenabízí žádné možnosti automatizace nebo skriptování. V softwaru není možné tvořit sekvence, u kterých by se měnilo základní rozlišení nebo bitový tok. Licence VQMT 12 s roční podporou stojí 27 000 korun a stejně by z ní pro tuto práci šlo použít pouze příkazovou řádku pro Linux, kterou je ale možné ve starší verzi 1.1 najít [9] a v nezměněné verzi pro vědecké účely používat. Nová verze nám nabízí zlepšení rychlosti výpočtu a výpočet VMAF, který ale umí spočítat FFmpeg s rozšířením přímo od firmy Netflix.

Listing 3.2: Ukázka výpočtu hodnot VQMT

```
vqmt original.yuv render.yuv ROZ_y ROZ_x \
FRAMES CHROMA Vysledne_hodnoceni
PSNR SSIM VIFP MSSSIM
```

3.4 Běh těchto softwarů

Je nutné se připravit na to, že při analýze audiovizuálních dat ve větších rozlišeních a hlavně v nekomprimovaném světě YUV se bude jednat o procesorově

Rozlišení		Bitový tok [Mb/s]
640	360	276,48
854	480	491,904
1280	720	1105,92
1920	1080	2488,32
3840	2160	9953,28

Tabulka 3.1: Bitové rychlosti pro 50p RAW

a hlavně paměťově náročné operace, jak je možné vidět v tabulce. Všechny aplikace, které vyžadují vstupy v YUV prostoru, budou tak problematické. Při větším nasazení aplikace bude vhodné zvolit takové softwary pro porovnávání, co samy převedou video do YUV až při spuštění výpočtu, a to ideálně rovnou přímo do operační paměti serveru.

Kapitola 4

Návrh systému pro analýzu adaptivního streamu

4.1 Požadavky

Před návrhem systému na generování a objektivní analýzu obrazu je nutné stanovit požadavky, které má splňovat. Systém by měl umožňovat generovat výsledek adaptivního streamu dle předem daných nastavení kodéru. Jedná se tedy o video, které by uživatel viděl, kdyby měl v dané časy danou rychlost připojení ke streamovacímu serveru. V této práci se vůbec nebude uvažovat linková vrstva síťového spojení a bude to jen spoj s definovanou garantovanou přenosovou rychlostí.

Vstupy a výstupy. Tento systém by měl poté umět porovnat vytvořené sekvence s originálním videem v maximální kvalitě a spočítat parametry: PSNR, SSIM, VIF, VMAF. Systém pro výpočet a porovnání by měl být do budoucna lehký rozšiřitelný o další parametry nebo další programy pro výpočet objektivních parametrů obrazu. Z proměnných parametrů je třeba reflektovat nastavitelnou délku segmentu a minimální hodnoty PSNR, SSIM,

VIF, VMAF pro každý tento segment. Výstupem tedy budou finální grafy a video splňující uživatelské požadavky.

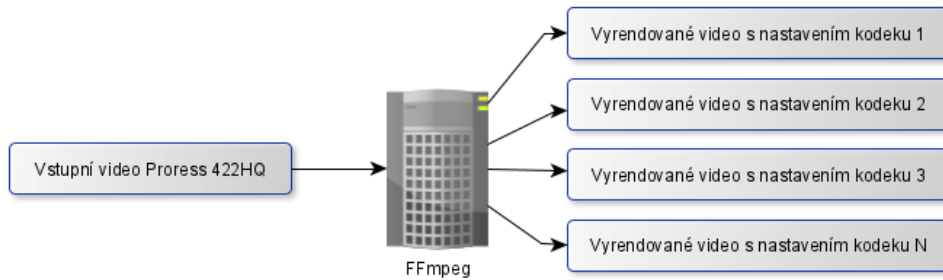
Software a licence. Použitý software by měl být volně šiřitelný nebo alespoň použitelný zdarma pro vědecké účely. Výsledná aplikace by měla být modulární tak, aby bylo možné v budoucnu přidat jiné kodéry nebo softwary pro výpočet objektivních metrik hodnocení kvality.

Budoucí rozšiřitelnost. Software by měl být navržen tak, aby bylo možné v budoucnu použít jiný software na vyhodnocování kvality obrazu, například o studenský projekt v rámci předmětu z magisterské etapy oboru Audiovizuální technika a zpracování signálů. Software by mělo být možné použít i na případné ukázky nebo jednoduché úkoly v rámci některých laboratorních cvičení zaměřených na kvalitu obrazu.

4.2 Návrh systému

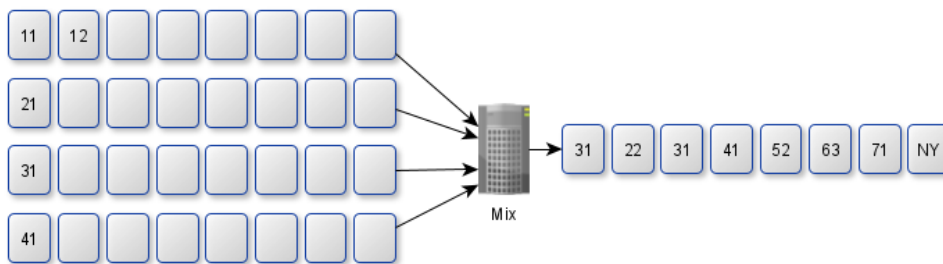
Motivace. Při záznamu adaptivního streamu na internetu je možné sice stahovat segmenty všech kvalit, ale bohužel nelze určit, jakým nastavením byly zakódovány, a ani většinou nelze získat zdrojová data v jakékoliv podobě. Proto je nutné vyrobit si vlastní streamy s různým nastavením kodéru, rozdělit je a vše pak zpětně složit dle předem daných pravidel, a to vše pro další analýzu.

Renderování dle nastavení kodéru. Pro renderování vícero videosekvencí pro další analýzu je použito video ve formátu Apple ProRes 422 HQ. Jelikož se jedná o formát, v němž umí většina moderních kamer a rekordérů nahrávat, a zároveň je tím zamezeno vzniku chyb v testovacím systému. Každý snímek je komprimovaný zvlášť, čímž je zabráněno problémům s různou mezinímkovou kompresí jiných formátů jako h264 a h265 u zdrojových dat pro vyhodnocení objektivních metrik kvality.



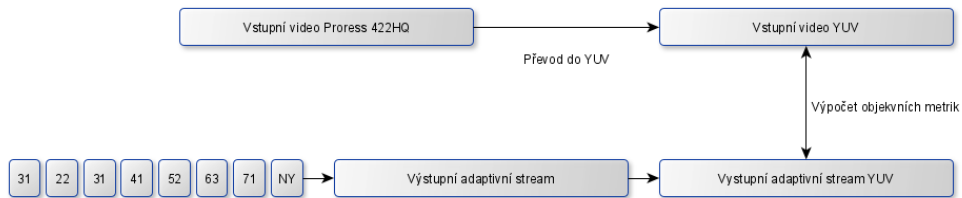
Obrázek 4.1: Renderování všech nastavení kodéru

Skládání dle nastavení uživatele. Již vyrenderované videosekvence se rozdělí do krátkých sekvencí dle nastavení uživatele. Typicky se bude jednat o jednotky sekund. Poté je nutné vyřešit poskládání zadané uživatelem.



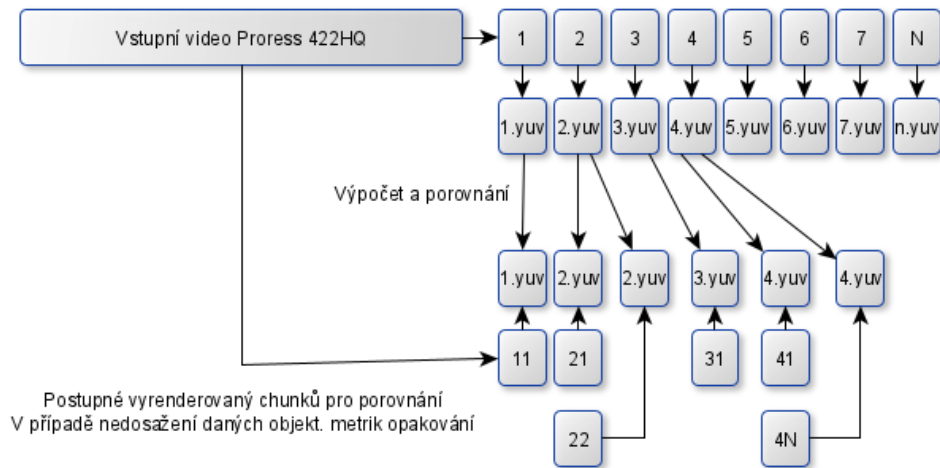
Obrázek 4.2: Skládání různých částí do finálního streamu

Skládání různých rozlišení. Může se stát že bude třeba za sebou skládat segmenty s různým rozlišením, což by pro finální výpočty nedávalo smysl a ani by to nereflektovalo zobrazení běžné na platformách YouTube nebo Twitch. Na nich při snížení rozlišení dojde ke zvětšení na stejnou velikost a rozlišení okna přehrávače. Z výše zmíněného vyplývá, že bude nutné videa s menším rozlišením před složením přeškálovat do rozlišení originálu.



Obrázek 4.3: Výpočet metrik

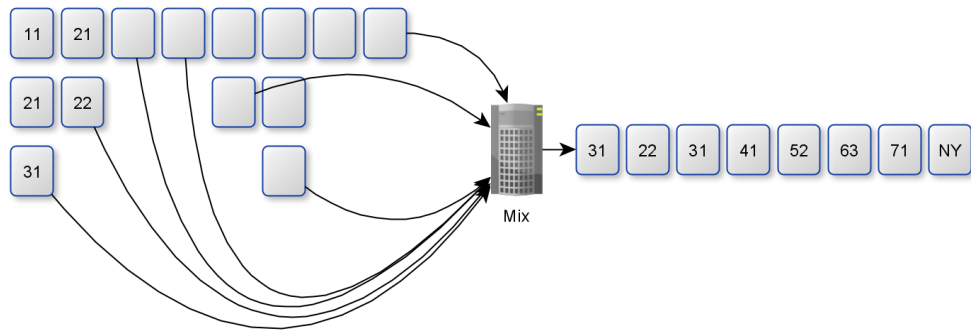
Skládání dle nastavení minimálních hodnot od uživatele. Výsledné video je možné také chtít zadat pomocí minimální hodnoty metriky objektivního hodnocení kvality, které bude splňovat každý segment vyrenderovaného videa. Jedná se o metriky PSNR, SSIM, VIF a VMAF. Do videosekvence tak budou zařazeny pouze ty části, co danou metriku splňují.



Obrázek 4.4: Postupné renderování a porovnávání

Ovládání skriptu. Skript by měl být plně automatizovatelný, čímž se rozumí textový konfigurační soubor nebo seznam parametrů na příkazové řádce.

Prezentace výsledku. Prezentace by měla probíhat formou grafu výsledných objektivních metrik mezi vyrenderovaným výsledkem a zdrojovými daty.



Obrázek 4.5: Výpočet metrik

Zdrojová data těchto grafů by měla být strojově čitelná a exportovatelná pro další práci například v prostředí MATLAB. Také by mělo být možné porovnat mezi sebou více nastavení, a to ideálně všechny.

Videa. Výsledná vyrenderovaná videa by mělo být možno spustit pomocí přehrávače, a to i ta převedená do světa YUV.

Kapitola 5

Nástroje

Zvolený software. Aby bylo možné vyhovět požadavkům na výsledný software, je zvoleno prostředí GNU/Linux, a to konkrétně distribuce Debian ve virtualizované podobě. Vzhledem k rozšířenosti na fakultě a dostupnosti studentům, co používají prostředí Windows nebo Apple, je jako prostředí pro virtualizaci vybrán Oracle Virtual Box. Výsledné řešení není distribuováno jako kontejner kvůli celkové složitosti a občasným změnám konfigurace. Jako kodér videa je pro svou rozšířenost a univerzálnost vybrán FFmpeg.

5.1 Vytvoření adaptivního streamu

gen_target_value.sh. Na generování částí adaptivního streamu dle zadaných parametrů je vytvořen nástroj `gen_target.sh`. Tento nástroj čeká jako argumenty název videa, metriku, kterou má splňovat, minimální hodnotu průměru této metriky, kterou musí splňovat v každém segmentu, délku segmentu a konfigurační soubor. Pro případ problému kvůli špatnému nastavení kodéru lze nalézt v souboru `DEBUG`, jak byl kodér spuštěn, a případně odladit problémy. Výsledná adaptivní videosekvence bude ve stejném adresáři a bude

se jmenovat stejně jako původní video, přibude k němu ovšem `_adaptiv`

Listing 5.1: Příklad spuštění

```
./gen_target.sh Capture0002.mov vmaf 70 5 setup.cfg
```

Listing 5.2: Příklad souboru setup.cfg

```
-c:v libx264 -b:v 1200k -maxrate 1200k -minrate 1200k -bufsize 1200k
-c:v libx264 -b:v 1600k -maxrate 1600k -minrate 1600k -bufsize 1600k
-c:v libx264 -b:v 2000k -maxrate 2000k -minrate 2000k -bufsize 2000k
-c:v libx264 -b:v 4000k -maxrate 4000k -minrate 4000k -bufsize 4000k
-c:v libx264 -b:v 8000k -maxrate 8000k -minrate 8000k -bufsize 8000k
-c:v libx264 -b:v 12000k -maxrate 12000k -minrate 12000k -bufsize 12000k
-c:v libx264 -b:v 16000k -maxrate 16000k -minrate 16000k -bufsize 16000k
-c:v libx264 -b:v 20000k -maxrate 20000k -minrate 20000k -bufsize 20000k
```

gen_target.sh. Pokud nechceme nastavovat parametry kodéru a máme zájem pouze o adaptivní videosekvenci co splňuje dané, kritérium. Můžeme použít modul `gen_target.sh`. Kde zadáváme pouze název souboru, technologii, hodnotu a délku segmentu.

5.2 Prezentace výsledků

gen_graph.sh. Pro prezentaci výsledku byl zvolen program Gnuplot, který byl obalen skriptem pro automatické vygenerování všech grafů. Výsledné grafy jsou obrázky ve formátu PNG a v rozlišení 1920 × 1080.

Listing 5.3: Generování grafů

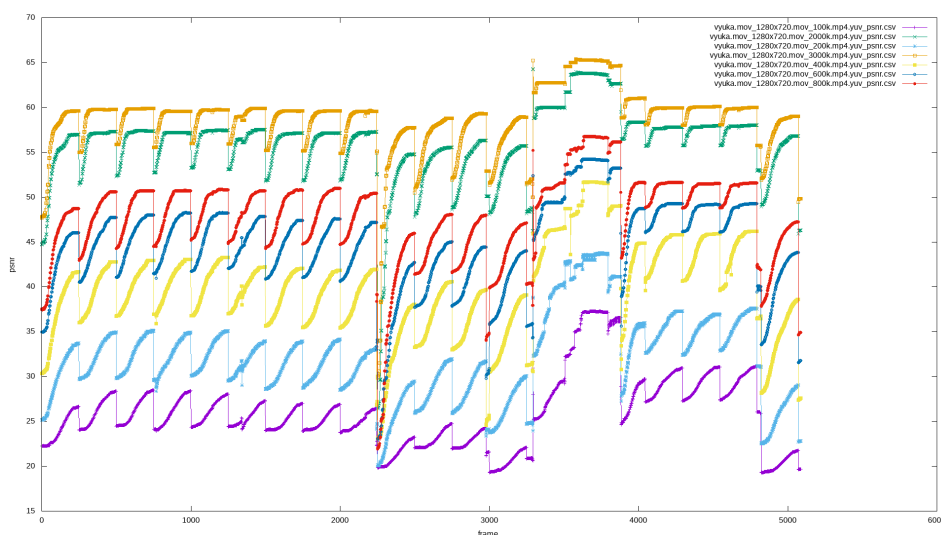
```
#!/gen_graph.sh nazev-video nazev-souboru
./gen_graph.sh vyuka.mov _overeni
```

5.3 Testování

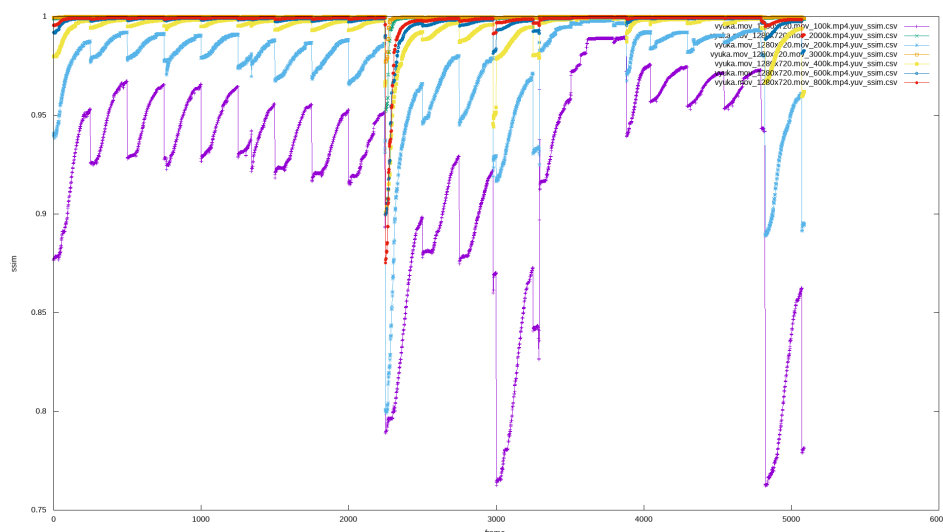
Platformu bylo třeba otestovat, zda pracuje dle očekávání.

Ověření – zlepšení objektivních metrik s rostoucím bitovým tokem. V této části je potřeba ověřit, že platforma pro větší bitový tok vygeneruje kvalitnější videosekvenci s lepším objektivním hodnocením kvality. Na níže uvedených obrazcích je jasně patrné, že se zvyšujícím se bitovým tokem rostou hodnoty PSNR, SSIM i VIF.

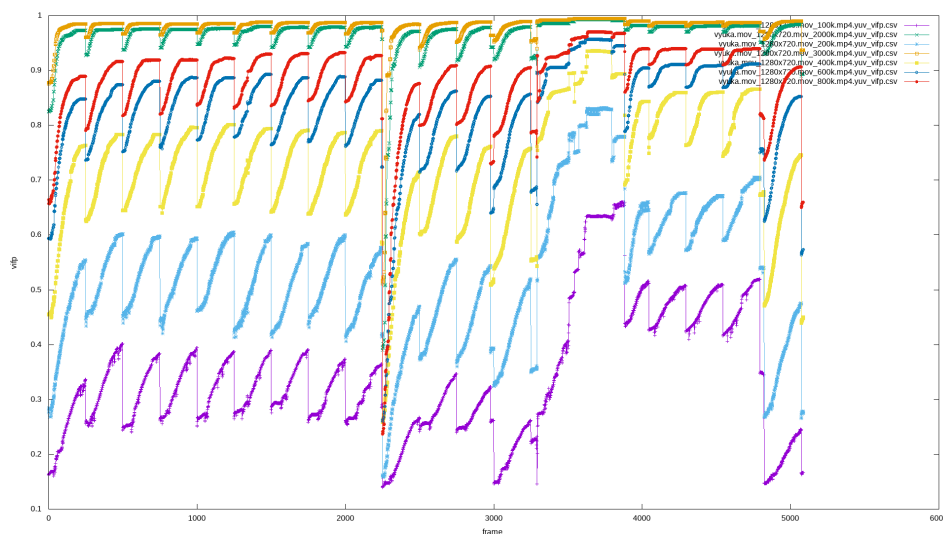
Výkon. Systém je velmi pomalý, a to hlavně proto, že vyhodnotit 60s video trvá 6 min. Proces renderování segmentu je oproti vyhodnocení relativně rychlý, a to přibližně čtyřikrát méně než doba trvání segmentu. Pro videosekvenci dlouhou 90s trvá kompletní analýza pro 8 různých nastavení přibližně 55 min a zabere 150 GB místa na disku. Jedná se o experimentální platformu takže výkon není tolik důležitý. Aktuálně systém běží na AMD RYZEN 9 9700X, 32GB RAM, 6T SATA3 7.2k.



Obrázek 5.1: PSNR pro různá nastavení bufferu



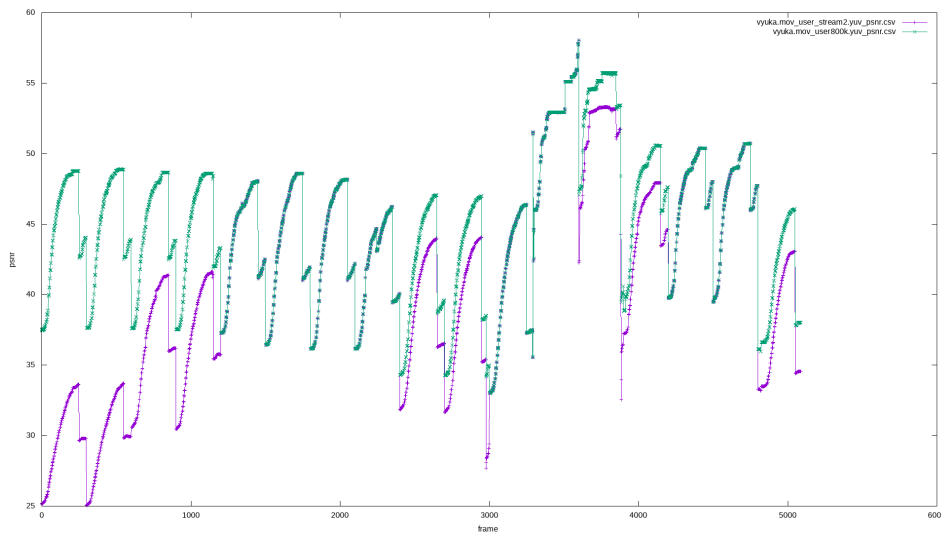
Obrázek 5.2: SSIM pro různá nastavení bufferu



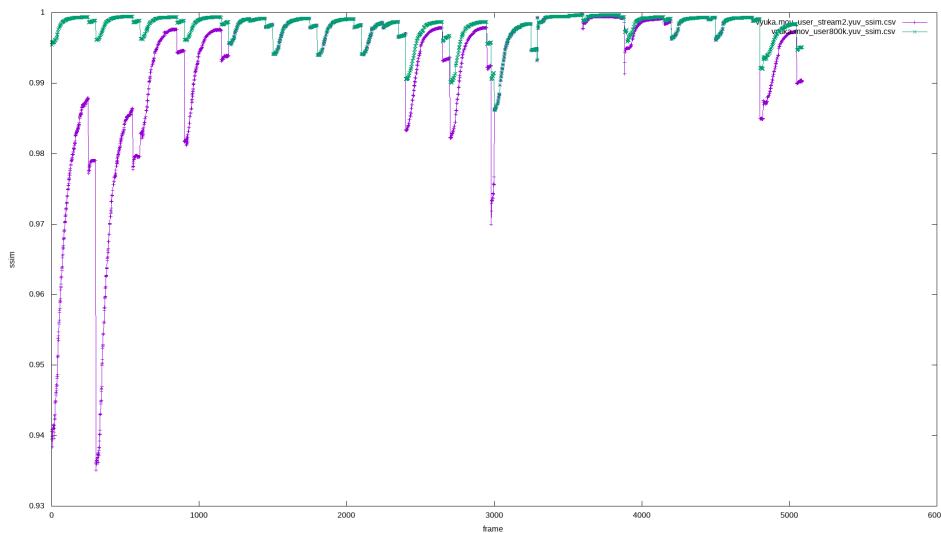
Obrázek 5.3: VIF pro různá nastavení bufferu

Ověření – vliv skládání segmentů na výsledek. Je třeba ověřit že platforma dává správné výsledky pro stream složený ze segmentů různých kvalit. Je tedy vyrenderována první videosekvence, která má celou dobu stejné parametry, a

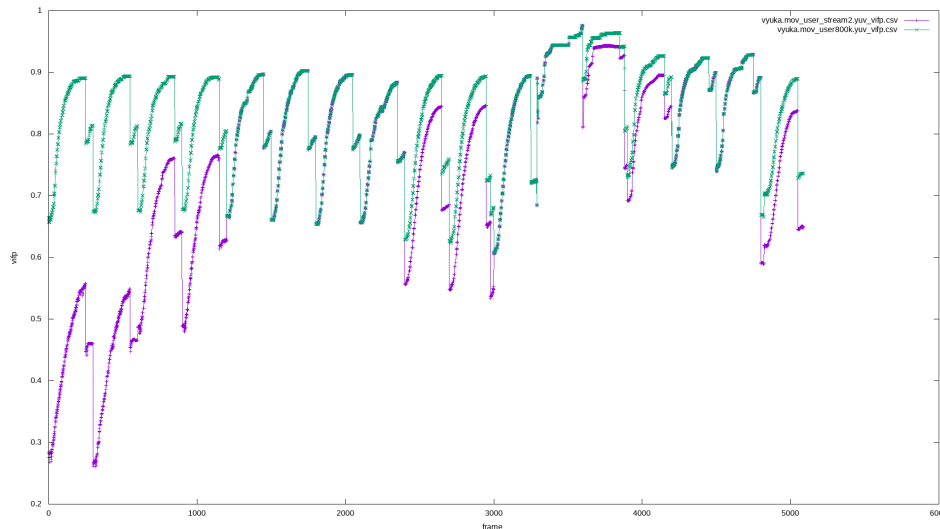
druhá, která se skládá z takového nastavení jen v některých částech. Na níže uvedených výsledcích je jasně patrné, že se výsledky fialového adaptivního streamu přesně shodují v těch segmentech, kde je použito stejné video. V ostatních segmentech jsou dosažené výsledky nižší, jelikož v nich byl použit menší bitový tok, a to pro všechny metriky.



Obrázek 5.4: PSNR pro různá nastavení bufferu



Obrázek 5.5: SSIM pro různá nastavení bufferu



Obrázek 5.6: VIF pro různá nastavení bufferu

gen_adap_stream.sh. Modul pro vytvoření adaptivního streamu, dle vybraní kvalit uživatel. Tato kvalita se zapisuje číselnou hodnotou vzestupně dle kvality pomocí číselné hodnoty do souboru q.txt.

Listing 5.4: q.txt

```
22446666556655665566556655666666666661
```

gen_adaptiv_mp4.sh. Modul pro spojování částí videí, které nejsou v YUV.

gen_ffvmaf.sh. Modul, který používá pro vyhodnocení objektivních metrik výhradně ffmpeg-vmaf.

gen_stats.sh. Modul, který převádí výsledky modulu gen_ffvmaf.sh do stejného formátu jako ostatní.

5.4 Užitečné příkazy pro práci s videosekvencemi

Listing 5.5: GOP struktura

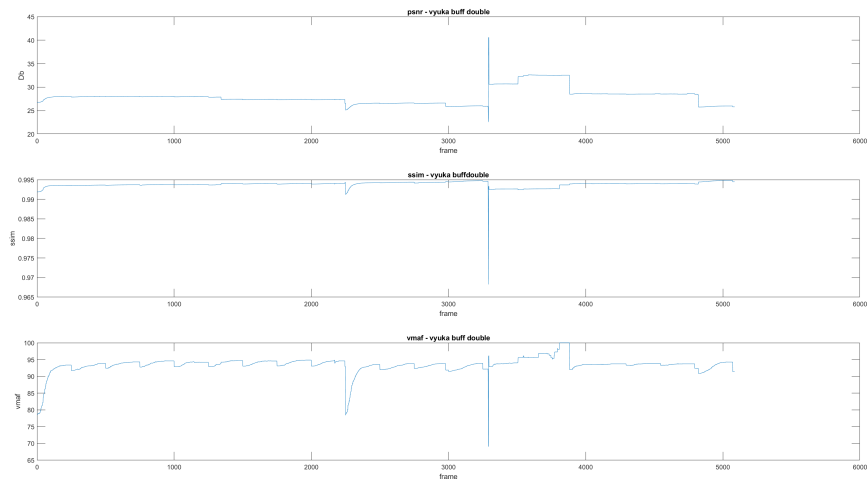
```
ffprobe -show_frames soubor.mp4 | grep pict_type \
| head -200 | cut -d= -f2 | tr "\n" ,
I ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,
P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,
P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,
P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,
P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,
P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,
P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,P ,B,B,B,
P ,B,B,B,P ,B,B,B,
```

Listing 5.6: Import a zobrazení naměřených dat v prostředí MATLAB

```
PSNR = readtable('vyuka_1920_720_1080_buff_double_stats_psnr');
SSIM = readtable('vyuka_1920_720_1080_buff_double_stats_ssim');
VMAF = readtable('vyuka_1920_720_1080_buff_double_stats_vmaf');

subplot(311)
plot(PSNR.frame, PSNR.vyuka_1920_720_1080_buff_double_stats_psnr)
title 'psnr - vyuka buff double', ylabel 'Db', xlabel 'frame'
subplot(312)
plot(SSIM.frame, SSIM.vyuka_1920_720_1080_buff_double_stats_ssim)
title 'ssim - vyuka buffdouble', ylabel 'ssim', xlabel 'frame'
subplot(313)
plot(VMAF.frame, VMAF.vyuka_1920_720_1080_buff_double_stats_vmaf)
title 'vmaf - vyuka buff double', ylabel 'vmaf', xlabel 'frame'
```

5.4. Užitečné příkazy pro práci s videosekvencemi



Obrázek 5.7: Grafování v prostředí MATLAB

Listing 5.7: Přehrání raw YUV videa v prostředí Windows

```
"C:\Program Files\VideoLAN\VLC\vlc.exe" --demux rawvideo \  
--rawvid-fps 50 \  
--rawvid-width 1920 \  
--rawvid-height 1080 \  
--rawvid-chroma I422 \  
data.yuv
```

Listing 5.8: Přehrání raw YUV videa v prostředí Linux

```
vlc --demux rawvideo \  
--rawvid-fps 50 \  
--rawvid-width 1920 \  
--rawvid-height 1080 \  
--rawvid-chroma I422 \  
data.yuv
```

Kapitola 6

Experimenty

6.1 Výukové video

Jednoduchý experiment se věnuje vlivu nastavení velikosti bufferu FFmpegu na kvalitu odstreamovaného výukového videa. Jedná se o video obrazovky, kde je otevřené okno s příkazovým řádkem interpretu bash. Toto video je zajímavé hlavně tím, že se většinu času mění pouze malá část scény, ale později dochází ke skokové změně, kdy se vše posune směrem nahoru a dole přibude úplně nová část. Tato problematika je zajímavá hlavně v momentě přípravy výukových videí pro programovací a administrační předměty nebo pro streamování videí do projektů zaměřených na inovaci výuky. Jelikož se bude jednat o velmi dlouhá, typicky 90min videa.

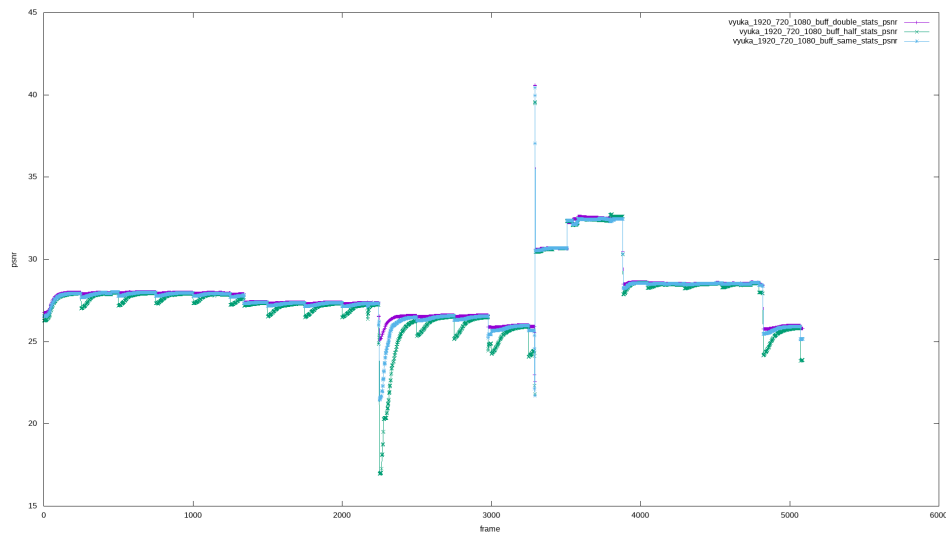
Konfigurace. Jako zdroj signálu byl použit notebook, který simuluje učitelské zařízení připojené ke streamovacímu řešení. Výstup signálu 1080p60 z notebooku byl z mini display portu zredukován na HDMI 1.4 a připojen k zařízení Blackmagic Video Assist 3G pro záznam zdrojového videa ve formátu ProRes HQ.



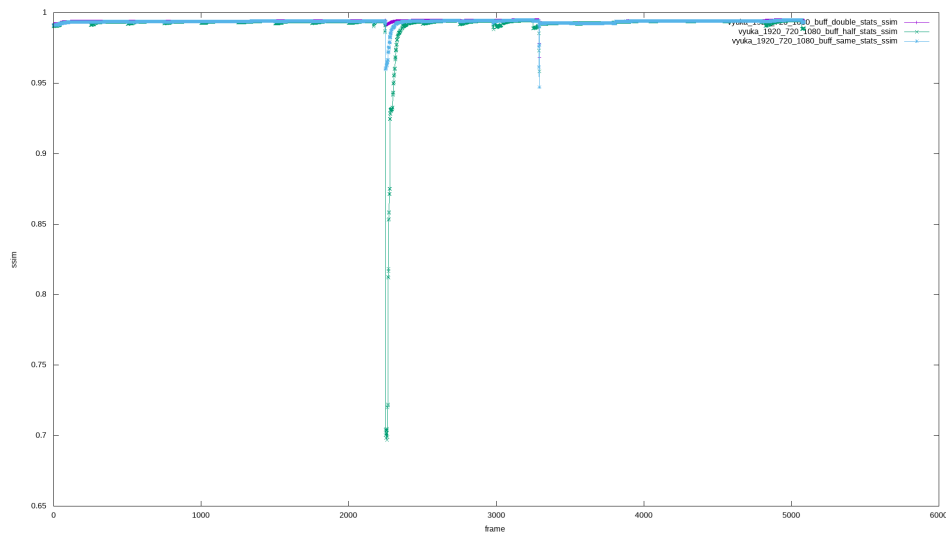
Obrázek 6.1: Sestava pro natáčení experimentu

Simulace. Zachycené video ve formátu ProRes bylo přeškálováno na rozlišení HD 1280×720 , ale bylo zatím ponecháno ve formátu ProRes HQ, aby bylo dosaženo minimální ztrátovosti. Toto video bylo překódováno do formátu h264 s maximálním datovým tokem 800 kb/s a rozlišením Full HD (1920×1080). Tento celý řetězec simuluje reálný adaptivní stream, který celý divák sledoval v druhém nejlepším poskytovaném rozlišení a měl zobrazen na fullscreen na Full HD monitoru. Tento experiment simuluje tedy 3 různá nastavení kodéru FFmpeg, ve kterém se mění velikost bufferu na polovinu, celek a dvojnásobek maximálního bitového toku.

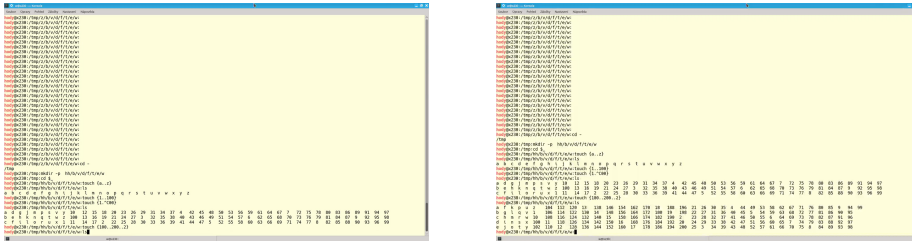
Výsledky. Na výsledných grafech PSNR, SSIM a VMAF je vidět, že k největšímu zlepšení dochází při nastavení velikosti bufferu na stejnou hodnotu jako maximální bitový tok pro tuto konfiguraci. Zároveň zde byl kompresní poměr fenomenální, a to 1:100.



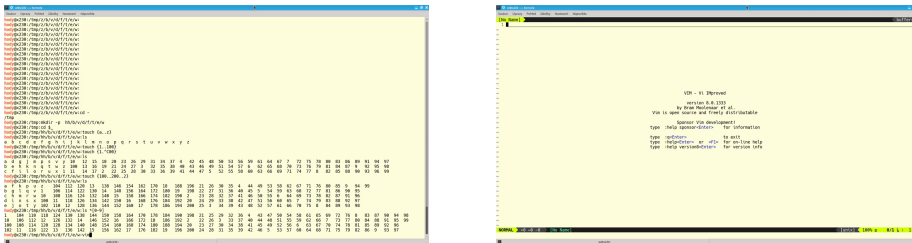
Obrázek 6.2: PSNR pro různá nastavení bufferu



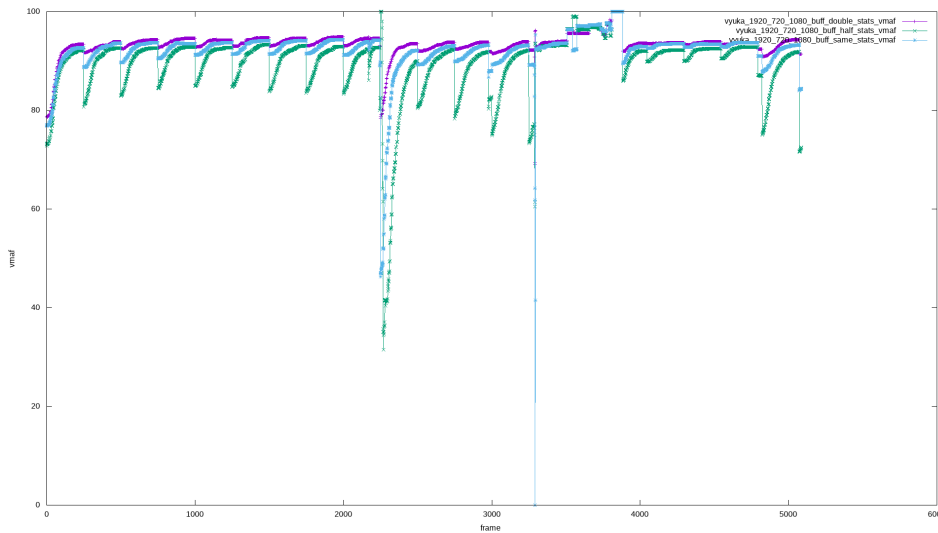
Obrázek 6.3: SSIM pro různá nastavení bufferu



Obrázek 6.5: Problémový přechod snímků 2200



Obrázek 6.6: Problémový přechod snímků 3100



Obrázek 6.4: VMAF pro různá nastavení bufferu

6.2 Vliv délky segmentu na výslednou kvalitu videa

Tento experiment se zabývá vlivem délky segmentu na výslednou kvalitu obrazu při definované minimální hodnotě metriky VMAF a to je 70. Pro tento experiment bylo natočeno video z počítačové hry Word of Tanks, která je často streamována na platformě Twitch nebo YouTube. Jedná se o video s vysokými detaily a velmi rychlou změnou ve scéně. Jako délky segmentu byly zvoleny časy 1, 3, 5, 10 sekund.

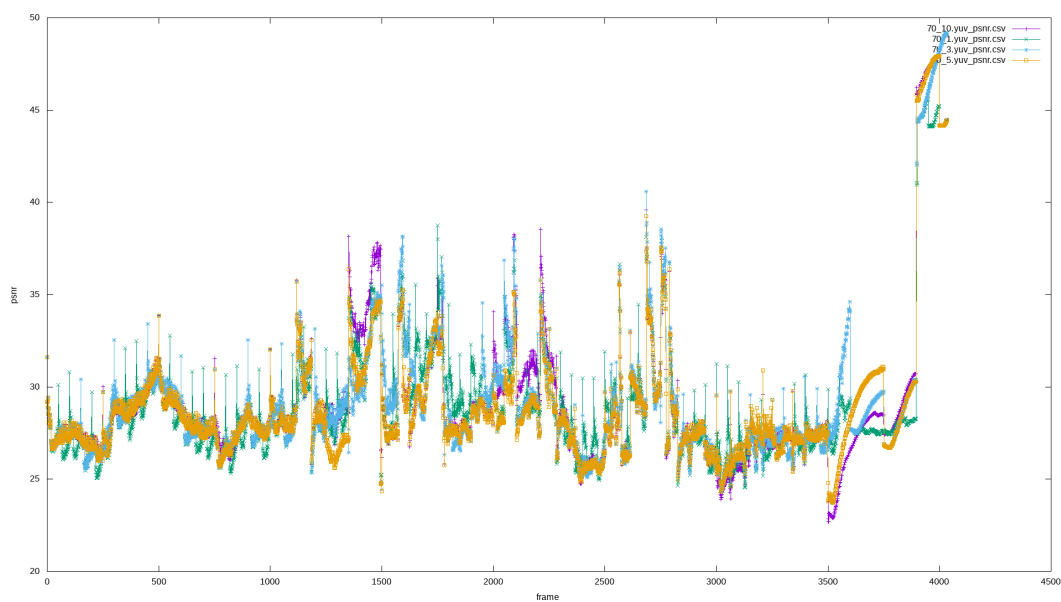
Konfigurace. Jako zdroj signálu byl použit počítač, který simuluje učitelské zařízení připojené ke streamovacímu řešení. Výstup signálu 1080p50 je zapojen přímo přes HDMI 1.4 k zařízení Blackmagic Video Assist 3G pro záznam zdrojového videa ve formátu ProRes HQ.

Výsledky. Z grafů můžeme vidět že velikost segmentu hraje roli ve výsledných hodnoceních hlavně v částech videosekvence, ve kterých není tolik detailů a nedochází k rychlým pohybům. Tam vychází lépe delší segment. Toto video je ale velmi dynamické a lze říci, že většinou je nejlepší 5s interval.

6.2. Vliv délky segmentu na výslednou kvalitu videa

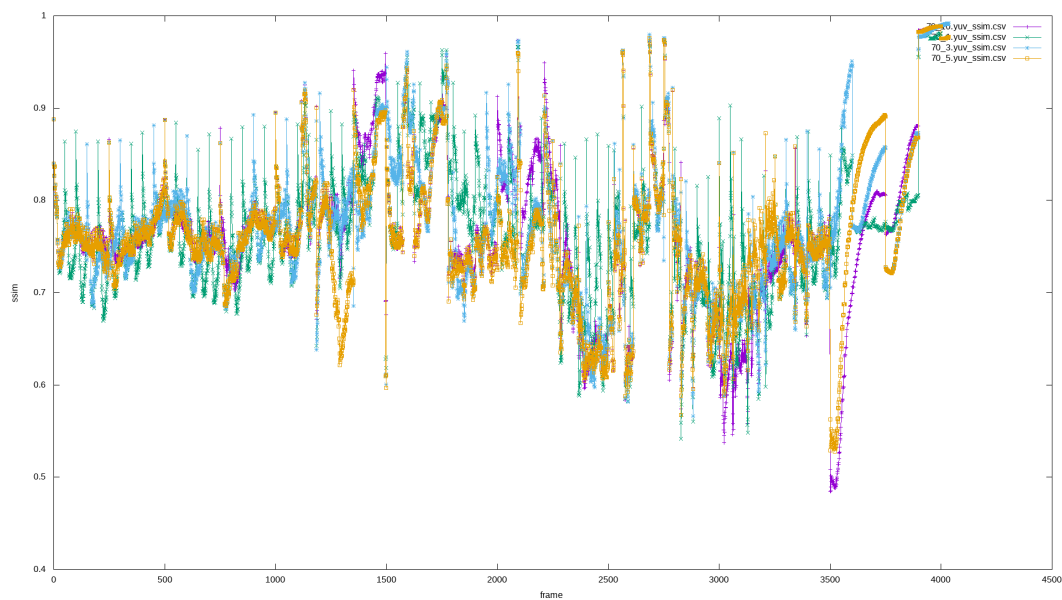


Obrázek 6.7: Sestava pro natáčení experimentu

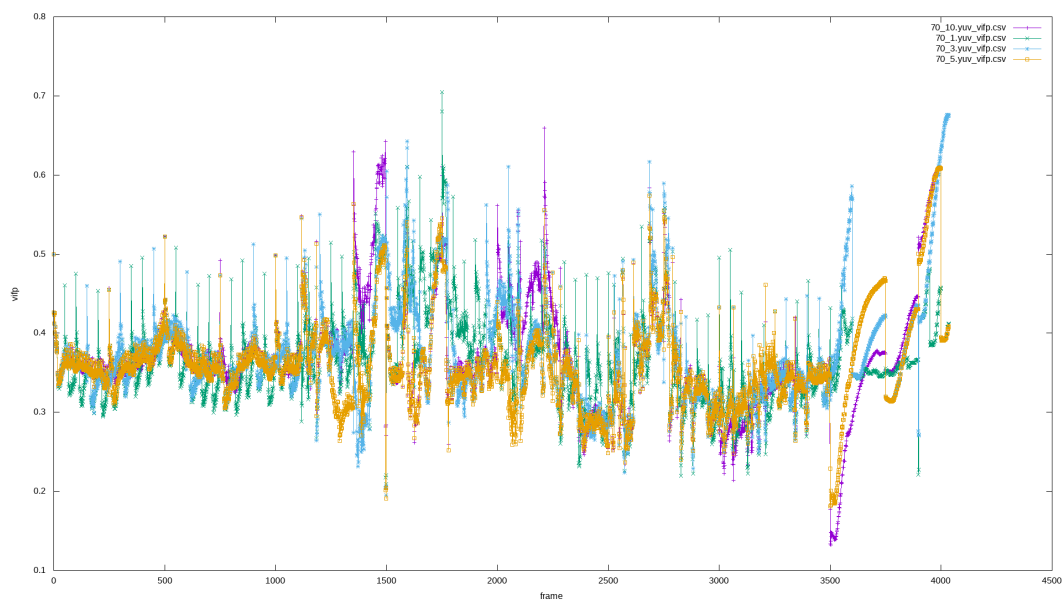


Obrázek 6.8: PSNR pro VMAF 70 pro segment 1, 3, 5, 10 s

6.2. Vliv délky segmentu na výslednou kvalitu videa



Obrázek 6.9: SSIM pro VMAF 70 pro segment 1, 3, 5, 10 s



Obrázek 6.10: VIFP pro VMAF 70 pro segment 1, 3, 5, 10 s

6.3 Vliv zaostření scény na výslednou kvalitu videosekvence

V tomto experimentu se zjišťuje, jak ovlivňuje zaostření scény výslednou kvalitu této sekvence pro dodržení alespoň VMAF 80 pro každý fragment scény.

Konfigurace. Na natáčení byla použita kamera Blackmagic 4K Production s Full Frame objektivem Sigma Art 24-70 F2.4 s vypnutou stabilizací. Kamerou je zabrán park, kde se postupně celá scéna zaostřuje. Doba trvání segmentu je 2 s.



Obrázek 6.11: Sestava pro natáčení experimentu

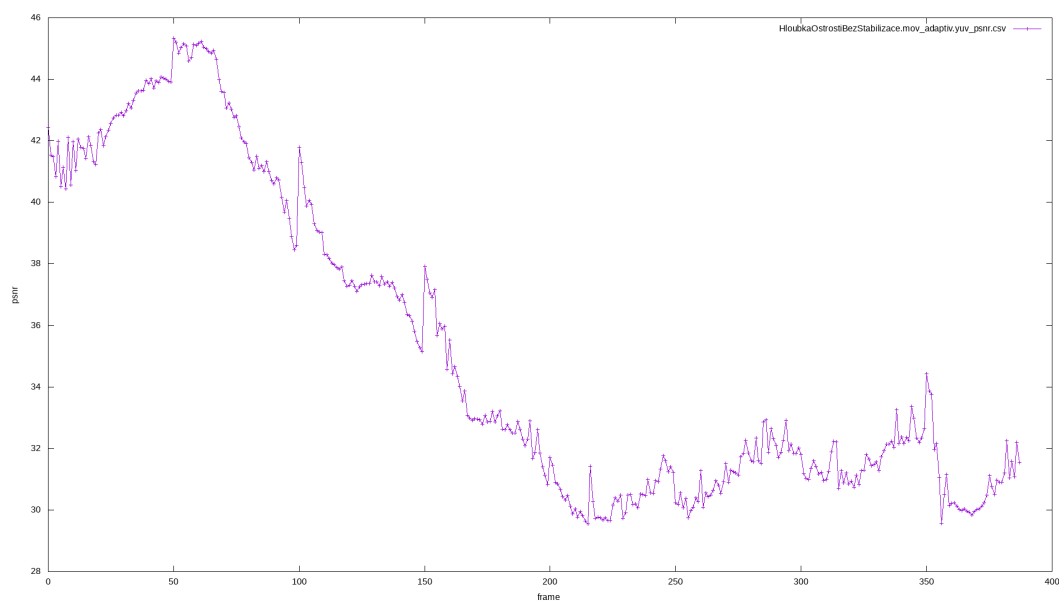
Výsledky. Na začátku, kde kvůli zaostření scény nejsou žádné detaily, běží komprese 600 kb/s a stejně dosahuje výborných výsledků ve všech měřeních

6.3. Vliv zaostření scény na výslednou kvalitu videosekvence

metrikách. S vyšší hloubkou ostrosti se zvyšuje bitový tok a výrazně se snižují hodnoty všech 3 metrik. Toto bylo očekáváno a výsledek tomu odpovídá.

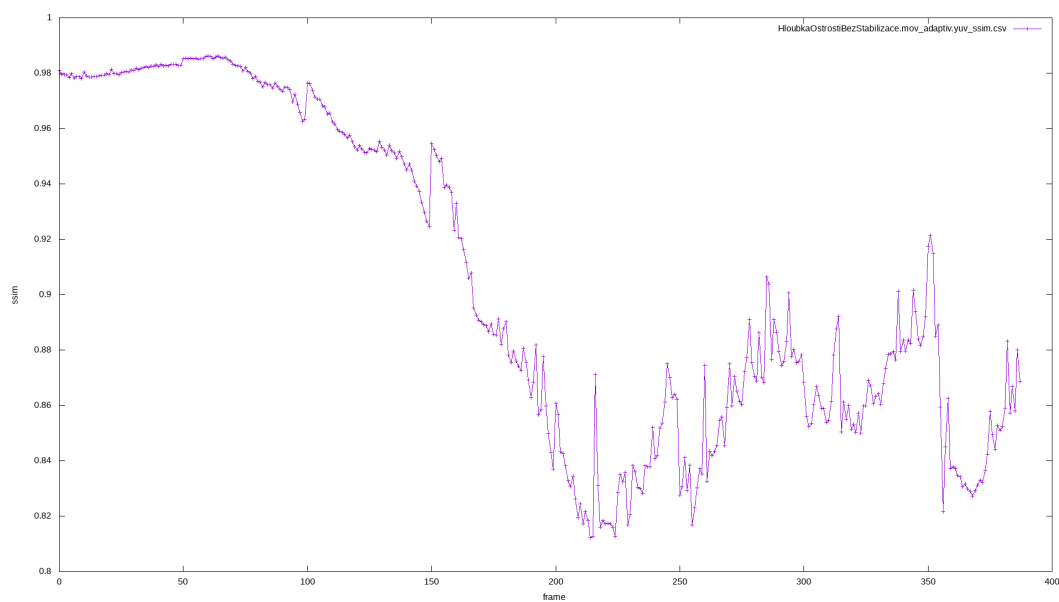
Listing 6.1: Příklad souboru před složením

```
zaostreniBezStabilizace.mov_1920x1080.mov_0_49.mov_600k.mp4.yuv  
zaostreniBezStabilizace.mov_1920x1080.mov_50_99.mov_1200k.mp4.yuv  
zaostreniBezStabilizace.mov_1920x1080.mov_100_149.mov_1200k.mp4.yuv  
zaostreniBezStabilizace.mov_1920x1080.mov_150_199.mov_1600k.mp4.yuv  
zaostreniBezStabilizace.mov_1920x1080.mov_200_249.mov_2000k.mp4.yuv  
zaostreniBezStabilizace.mov_1920x1080.mov_250_299.mov_1600k.mp4.yuv  
zaostreniBezStabilizace.mov_1920x1080.mov_300_349.mov_1600k.mp4.yuv  
zaostreniBezStabilizace.mov_1920x1080.mov_350_388.mov_2000k.mp4.yuv
```

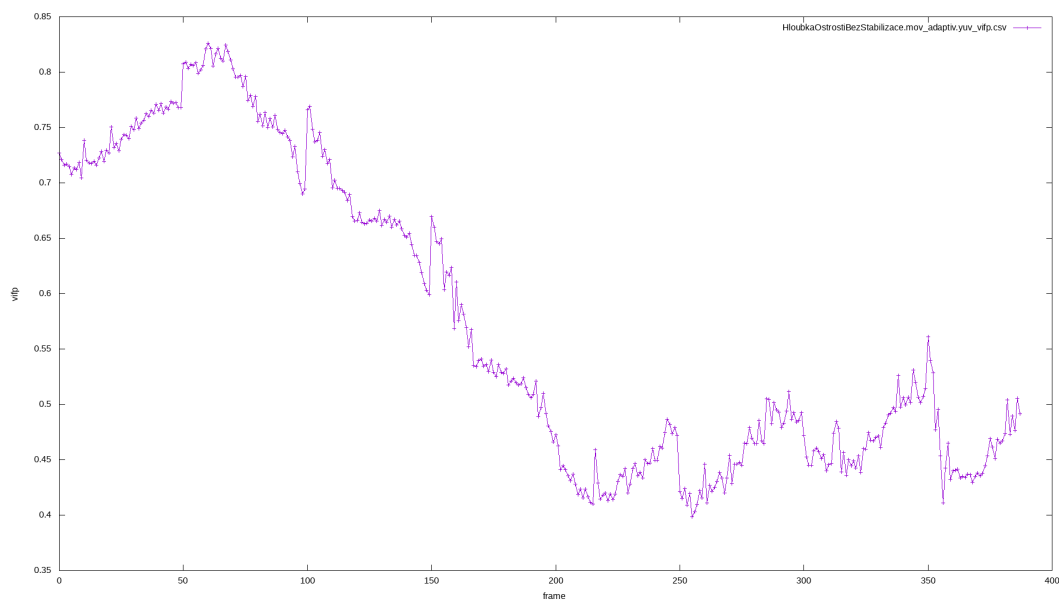


Obrázek 6.12: Vliv zaostření na kvalitu scény s min VMAF 80 na PSNR

6.3. Vliv zaostření scény na výslednou kvalitu videosekvence



Obrázek 6.13: Vliv zaostření na kvalitu scény s min VMAF 80 na SSIM



Obrázek 6.14: Vliv zaostření na kvalitu scény s min VMAF 80 na VIF



Kapitola 7

Závěr

V této práci jsou rozebrány základní techniky distribuce digitálního video obsahu a vysvětleny metriky objektivního hodnocení kvality obrazu. Z těchto získaných informací se povedlo stanovit důležité požadavky na navrhovanou platformu.

Povedlo se navrhnout a implementovat platformu, která umí tvořit a hodnotit videosekvence na základě uživatelského nastavení, a to jak bitového toku, tak metrik PSNR, SSIM a VMAF. Platforma je lehce upravitelná v případě použití jiných softwarů na objektivní hodnocení kvality obrazu. Také není závislá na použitém kodeku, takže je možné porovnávat různé kodeky mezi sebou. Dále není podstatná ani snímková frekvence testované videosekvence, jelikož je na to platforma připravena.

Funkčnost systému se povedla ověřit na několika experimentech pomocí připravených testovacích videosekvencí různých typů, kde práce došla k očekávaným výsledkům.

Tuto platformu je možné jednoduše používat na demonstraci i při výuce v laboratoři v předmětu B2M37KASA – Komprese obrazů a signálů. Také je

možné vytvořit videosekvence pro subjektivní testování kvality obrazu dle nastavených kritérií.

Výsledné videosekvence je možné přehrávat pomocí přehrávače VLC. Bitový tok těchto raw videosekvencí je sice velký, ale díky vhodnému návrhu je dostupný přímo z operačního systému, který hostuje tuto platformu jako virtuální server. To umožňuje přímý přístup k souboru a není ho třeba stahovat po 100/1000 Mb/s síti.

Možné varianty rozšíření. Zpětně je patrné, že uložení veškerých informací o videosekvencích do názvu není tak praktické. Existuje totiž omezení na délku 255 znaků [28]. Aktuálně dosahuje délka nejdelšího souboru 142 znaků. To bude hlavně problém u videí, co mají délku větší než několik hodin. Toto by také vyřešilo, že soubor nesmí obsahovat znak „_“, jelikož se používá na interní parsování.

Dále by bylo možné rozšířit výstupní vyhodnocená data o informaci, o který snímek z GOP se jednalo. Rovněž by bylo zajímavé pro zlepšení celkového vjemu diváka všude používat harmonický průměr místo aritmetického.

System by dále mohl mít také komponentu, která by hádala na základě předchozí vyrenderované části, jaké nastavení použít pro další část. Tím by se částečně ušetřil výkon stroje, na kterém platforma běží, a tím pádem by systém generoval výsledky rychleji.



Literatura

- [1] *Stream traffic [online]. [cit. 12.06.2020]. dostupné z, <https://www.sandvine.com/global-internet-phenomena-report-2019>.*
- [10] *Ffmpeg filters [online]. [cit. 21.05.2020]. dostupné z, <https://trac.ffmpeg.org/wiki/FilteringGuide>.*
- [11] *Netflix vmaf [online]. [cit. 01.05.2020]. dostupné z, <https://netflixtechblog.com/vmaf-the-journey-continues-44b51ee9ed12>.*
- [12] *Fel youtube [online]. [cit. 01.08.2020]. dostupné z, https://www.youtube.com/watch?v=T4rNFNgt_IE.*
- [13] *Netflix title optimazilation [online]. [cit. 12.06.2020]. dostupné z, <https://netflixtechblog.com/per-title-encode-optimization-7e99442b62a2>.*
- [14] *Harmonic mean [online]. [cit. 28.06.2020]. dostupné z, <https://www.askiitians.com/iit-jee-progressions-and-series/harmonic-mean/>.*
- [15] *H264 [online]. [cit. 22.07.2020]. dostupné z, <https://www.itu.int/rec/T-REC-H.264-200503-S/en/>.*

- [16] *H264* [online]. [cit. 22.07.2020]. dostupné z, https://www.researchgate.net/figure/H264-GOP-and-the-coding-mechanism-for-Baseline-profile-5_fig3_303535868.
- [17] *Vp9 – youtube cvut* [online]. [cit. 23.07.2020]. dostupné z, <https://www.youtube.com/watch?v=F7uYyJAvDjE>.
- [18] *Vysílací řetězec České televize* [online]. [cit. 1.05.2020]. dostupné z, https://img.ceskatelevize.cz/boss/image/contents/technika/obr/tv_retezec.gif.
- [19] *Wowza video report 2019* [online]. [cit. 12.06.2020]. dostupné z, <https://www.wowza.com/blog/2019-video-streaming-latency-report>.
- [2] *Doporučené přenosové rychlosti* [online]. [cit. 12.05.2020]. dostupné z, <https://support.google.com/youtube/answer/1722171?hl=cs>.
- [20] *Netflix reduce bit stream* [online]. [cit. 12.06.2020]. dostupné z, <https://www.bbc.com/news/technology-51968302>.
- [21] *Video quality estimator* [online]. [cit. 12.07.2020]. dostupné z, <https://www.elecard.com/products/video-analysis/video-quality-estimator>.
- [22] *Mpeg 2. vs h264 bitrate*, <http://www.gpds.ene.unb.br/databases/2012-UNB-Varium-Exp/Exp3-Delft/00-report-alexandre/Papers---Searched/HDTVSubjectiveQualityOfH264vsMPEG-2WithAndWithoutPacketLoss.pdf>.
- [23] *Mse vs ssim*, <https://www.ijser.org/researchpaper/MSE-Vs-SSIM.pdf>.
- [24] *Comparative analysis of vp8 and h264*, https://www.researchgate.net/publication/261074990_Detailed_Comparative_Analysis_of_VP8_and_H264.
- [25] *Adaptivní stream* [online]. [cit. 12.05.2020]. dostupné z, <https://netflixtechblog.com/using-machine-learning-to-improve-streaming-quality-at-netflix-9651263ef09f>.
- [26] *Mpeg-dash* [online]. [cit. 15.05.2020]. dostupné z, <https://www.synopi.com/mpeg-dash/>.

- [27] *Hierarchy playlistu [online]. [cit. 13.05.2020]. dostupné z, <https://developer.apple.com/library/archive/referencelibrary/GettingStarted/AboutHTTPLiveStreaming/about/about.htm>.*
- [28] *filespecsyntax [online]. [cit. 14.08.2020]. dostupné z, https://www.ibm.com/support/knowledgecenter/SSEQVQ_8.1.9/client/c_cmd_filespecsyntax.html.*
- [3] *A quality-of-experience database for adaptive video streaming [online]. [cit. 12.06.2020]. dostupné z, <https://ece.uwaterloo.ca/~zduanmu/tbc2018qoe/paper/tbc2018sqoe.pdf>.*
- [4] *Hls rfc specification [online]. [cit. 11.05.2020]. dostupné z, <https://tools.ietf.org/html/rfc8216>, Citováno: 2020-02-02.*
- [5] *Optimizing video encoding for adaptive streaming over http, <https://ieeexplore.ieee.org/abstract/document/6227439/>.*
- [6] *Automated qoe evaluation of dynamic adaptive streaming over http, <https://ieeexplore.ieee.org/abstract/document/6603211/>.*
- [7] *Psnr [online]. [cit. 15.06.2020]. dostupné z, https://moodle.fel.cvut.cz/pluginfile.php/196197/mod_resource/content/0/b2m37kasa_05_v2.00_190318.pdf.*
- [8] *Vqmt [online]. [cit. 11.04.2020]. dostupné z, https://www.compression.ru/video/quality_measure/vqmt_download.html#start.*
- [9] *Vqmt sw [online]. [cit. 11.04.2020]. dostupné z, <https://github.com/Rolinh/VQMT>.*

Příloha A

Kompilace použitých softwarů

Listing A.1: Kompilace virtualbox-tools

```
apt install build-essential dkms linux-headers-$(uname -r)
mkdir -p /mnt/cdrom
mount /dev/cdrom /mnt/cdrom
cd /mnt/cdrom
sh ./VBoxLinuxAdditions.run --nox11
```

Listing A.2: Instalace Gnuplot

```
apt install gnuplot
```

Listing A.3: Kompilace VQMT

```
apt install libopencv-dev
su user
git clone https://github.com/Rolinh/VQMT.git
cd VQMT/
cmake .
```

Listing A.4: Kompilace filtru libvmaf

```
git clone https://github.com/Netflix/vmaf.git
cd ./libvmaf
sudo apt install ninja-build meson
```

```
sudo apt install libvmaf-dev >=1.3.9
sudo cp ./vmaf/model/vmaf_v0.6.1.pkl* \
cesta_k_modelu
meson build --buildtype release
ninja -vC build
```

Listing A.5: Kompilace FFmpeg s filtrem VMAF

```
git clone https://github.com/FFmpeg/FFmpeg.git
cd ./FFmpeg
./configure --disable-x86asm --enable-libvmaf \
--enable-version3 --enable-libx264 --enable-gpl
make
make install
```



Příloha B

Seznam zkratek

- **AM:** Aritmetický průměr
- **APSNR:** Aligned Peak Signal to Noise Ratio
- **AV1:** AOMedia Video 1
- **CMAF:** Common Media Application Format
- **DCT:** Diskrétní kosinová transformace
- **DVB-T:** Digital Video Broadcasting – Terrestrial
- **DVD:** Digital Versatile Disc
- **GOP:** Group of Pictures
- **HD:** High Definition
- **HDMI:** High-Definition Multimedia Interface
- **HEVC:** High Efficiency Video Coding
- **HM:** Harmonický průměr
- **HTML:** Hypertext Markup Language

- **HTTP**: Hypertext Transfer Protocol
- **HLS**: HTTP Live Streaming
- **IP**: Internet Protocol
- **IPTV**: Internet Protocol Television
- **JPEG**: Joint Photographic Experts Group
- **JSON**: JavaScript Object Notation
- **MPEG**: Moving Picture Experts Group
- **MPEG-4 AVC**: MPEG-4 Advanced Video Coding
- **MPEG-DASH**: MPEG Dynamic Adaptive Streaming over HTTP
- **MPEG-TS**: MPEG Transport Stream
- **MSAD**: Mean Sum of Absolute Differences
- **MSE**: Mean Squared Error
- **PNG**: Portable Network Graphics
- **PSNR**: Peak Signal to Noise Ratio
- **RFC**: Request for Comments
- **RTMP**: Real-Time Messaging Protocol
- **SD**: Standard Definition
- **SNR**: Signal to Noise Ratio
- **SRT**: Secure Reliable Transport
- **SSIM**: Structural Similarity
- **SSL**: Secure Sockets Layer
- **TCP**: Transmission Control Protocol
- **VIF**: Visual Information Fidelity
- **VMAF**: Video Multimethod Assessment Fusion

- **VQM**: Video Quality Measurement
- **VQMT**: Video Quality Measurement Tool
- **WebRTC**: Web Real-Time Communication



Příloha C

Obsah přiloženého média

- .
- + hodacpet_data.tar.gz
- + Capture0002.mov
- + hodacpet_data.tar.gz
- + RozostrenoBezStabilice.mov
- + SpatnaStabilizaceGOPRO.mov
- + tanky.mov
- + virtual.ova
- + virtual.pass
- + vyuka.mov