

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

PLATFORMA PRO SUBJEKTIVNÍ HODNOCENÍ VIDEO SEKVENCÍ

PLATFORM FOR SUBJECTIVE EVALUATION OF VIDEO-SEQUENCES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Tomáš Srnec

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Číka, Ph.D.

BRNO 2016

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Tomáš Srnec

ID: 164783

Ročník: 3

Akademický rok: 2015/16

NÁZEV TÉMATU:

Platforma pro subjektivní hodnocení video sekvencí

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte standard ITU-T P.910, který definuje metody pro subjektivní měření kvality videosekvencí. Připravte si sérii nekomprimovaných videosekvencí, které zkomprimujete moderními video kodeky pomocí knihovny FFMPEG. Navrhněte a vytvořte aplikaci na stolní PC a prostředí, ve kterém budete jednotlivé videosekvence kvalitativně testovat. Vyberte vzorek lidí, se kterými provedete subjektivní test dle vybrané metodiky z ITU-T P.910 a výsledky zhodnoťte.

DOPORUČENÁ LITERATURA:

[1] REC. ITU-T P.910 (04/2008). Subjective video quality assessment methods for multimedia applications. -: International Telecommunication Union, 2008.

Termín zadání: 1.2.2016

Termín odevzdání: 1.6.2016

Vedoucí práce: Ing. Petr Číka, Ph.D.

Konzultant bakalářské práce:

doc. Ing. Jiří Mišurec, CSc., předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato bakalářská práce se zabývá subjektivním hodnocením kvality videosekvencí. V první kapitole jsou obecně popsány použité moderní kodeky H.264, H.265, VP8 a VP9. V další části se rozebírají čtyři metody pro subjektivní hodnocení videosekvencí dle normy ITU-T P.910. Praktická část obsahuje kódování tří vybraných videí, do čtyř kodeků, pro čtyři rozlišení. Výstupem je JavaFX aplikace, ve které účastníci testování zhlédli a ohodnotili připravená videa. Jejich hodnocení se v reálném čase odesílalo na MYSQL server a přímo v aplikaci se vyhodnotilo do přehledných graů. Po zhodnocení výsledků byl určen jako nejlepší kodek VP9 před kodeky H.265, H.264 a VP8.

KLÍČOVÁ SLOVA

Kvalita videa, Srovnání kodeků, H.264, H.265, VP8, VP9, FFmpeg, JavaFX, MYSQL

ABSTRACT

This bachelor thesis is focused on subjective video quality assessment. Used modern codecs such as H.264, H.265, VP8 and VP9 are described in first chapter. In the next part of the thesis, four methods of the subjective video assessment are being called, according to Recommendation ITU-T P.910. The practical part includes encoding of four videos, into four resolutions, for four codecs. Output of the thesis is JavaFX application, capable of playing used videos for participants of test, who are making judgment. Their results are real-time sent to MySQL server and directly in application evaluated into bar charts. According to our results, the best codec is VP9, before codec H.265, H.264 and VP8.

KEYWORDS

Video Quality, Codec comparison, H.264, H.265, VP8, VP9, FFmpeg, JavaFX, MYSQL

SRNEC, Tomáš *Platforma pro subjektivní hodnocení videosekvencí*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 50 s. Vedoucí práce byl Ing. Petr Číka, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Platforma pro subjektivní hodnocení videosekvencí“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Petru Číkovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

podpis autora(-ky)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

Výzkum popsany v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....

podpis autora(-ky)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

| | |
|---|-----------|
| Úvod | 11 |
| 1 Použité video kodeky | 12 |
| 1.1 H.264 | 12 |
| 1.2 H.265 | 14 |
| 1.3 VP8 | 17 |
| 1.4 VP9 | 18 |
| 2 Použitý software | 19 |
| 2.1 FFmpeg | 19 |
| 2.2 Java | 19 |
| 2.2.1 JavaFX | 19 |
| 2.3 MySQL | 20 |
| 2.3.1 MySQL Workbench | 20 |
| 3 Metody subjektivního hodnocení | 22 |
| 3.1 Charakteristika scény | 22 |
| 3.2 Testovací metody | 22 |
| 3.2.1 Absolute category rating (ACR) | 22 |
| 3.2.2 Absolute category rating with hidden reference (ACR-HR) | 23 |
| 3.2.3 Degradation category rating (DCR) | 24 |
| 3.2.4 Pair comparison method (PC) | 25 |
| 3.3 Porovnání metod | 26 |
| 3.4 Referenční podmínky | 27 |
| 3.5 Zpracování a přehrávání videa | 27 |
| 3.6 Účastníci testu | 28 |
| 3.6.1 Pokyny pro účastníky a školení | 28 |
| 4 Praktická část | 29 |
| 4.1 Popis použitých zařízení | 29 |
| 4.2 Realizace MySQL serveru | 29 |
| 4.2.1 Použité příkazy | 29 |
| 4.3 Výběr metody | 30 |
| 4.4 Testované videosekvence | 30 |
| 4.4.1 Proces kódování | 31 |
| 4.5 Testovací aplikace | 33 |
| 4.5.1 Ovládání a funkce aplikace | 33 |
| 4.6 Průběh testování | 38 |

| | | |
|----------|--|-----------|
| 5 | Výsledky testování | 39 |
| 5.1 | Rozlišení 1080p | 39 |
| 5.2 | Rozlišení 720p | 40 |
| 5.3 | Rozlišení 480p | 41 |
| 5.4 | Rozlišení 360p | 42 |
| 5.5 | Celkové výsledky | 43 |
| 5.6 | Porovnání výsledků | 44 |
| 6 | Závěr | 45 |
| | Literatura | 46 |
| | Seznam symbolů, veličin a zkratk | 48 |
| | Seznam příloh | 49 |
| A | Příloha | 50 |
| A.1 | Datová komprese použitých kodeků | 50 |

SEZNAM OBRÁZKŮ

| | | |
|-----|---|----|
| 1.1 | H.264 Proces zpracování videa | 12 |
| 1.2 | Struktura HEVC kodéru a dekodéru | 15 |
| 2.1 | Prostředí MySQL Workbench | 21 |
| 3.1 | Možné nastavení ACR metody | 22 |
| 3.2 | Možné nastavení DCR metody | 25 |
| 3.3 | Možné nastavení PC metody | 26 |
| 4.1 | Úvodní obrazovka | 33 |
| 4.2 | Začátek testování | 34 |
| 4.3 | Průběh testování | 35 |
| 4.4 | Video station2 | 36 |
| 4.5 | Video crowd_run | 36 |
| 4.6 | Video ducks_take_off | 37 |
| 4.7 | Video old_town_cross | 37 |
| 5.1 | Výsledky hodnocení u jednotlivých videí pro rozlišení 1920x1080 | 39 |
| 5.2 | Výsledky hodnocení u jednotlivých videí pro rozlišení 1280x720 | 40 |
| 5.3 | Výsledky hodnocení u jednotlivých videí pro rozlišení 854x480 | 41 |
| 5.4 | Výsledky hodnocení u jednotlivých videí pro rozlišení 640x360 | 42 |
| 5.5 | Celkové výsledky hodnocení u jednotlivých videí | 43 |

SEZNAM TABULEK

| | | |
|-----|--|----|
| 1.1 | Zmenšení průměrného bit ratu u HEVC v porovnání s H.264/MPEG-4 [6] | 14 |
| 4.1 | Ukázka jednoho řádku v databázi | 29 |
| 4.2 | Doporučené hodnoty bit ratu pro daná rozlišení | 31 |
| 4.3 | Použité profily kodeků | 31 |
| 4.4 | Použité příkazy | 32 |
| 5.1 | Výsledky pro rozlišení 1920x1080 | 39 |
| 5.2 | Výsledky pro rozlišení 1280x720 | 40 |
| 5.3 | Výsledky pro rozlišení 854x480 | 41 |
| 5.4 | Výsledky pro rozlišení 640x360 | 42 |
| 5.5 | Celkové výsledky pro všechna rozlišení | 43 |
| A.1 | Kompresa dat. | 50 |

ÚVOD

Objem dat přenášený na celém světě internetovým protokolem IP roste každý rok o desítky až stovky procent. Společnost Cisco předpovídá do roku 2019 pětinasobný nárůst objemu. Přenosy videa budou představovat 80% celosvětového datového provozu z toho dvě třetiny dat budou připadat na mobilní připojení.[1] Díky tomu se kladou stále větší nároky na vývoj nových kodeků, které musí umět při zachování stejné kvality a velikosti zpracovávat videa ve větším rozlišení, než jejich předchůdci.

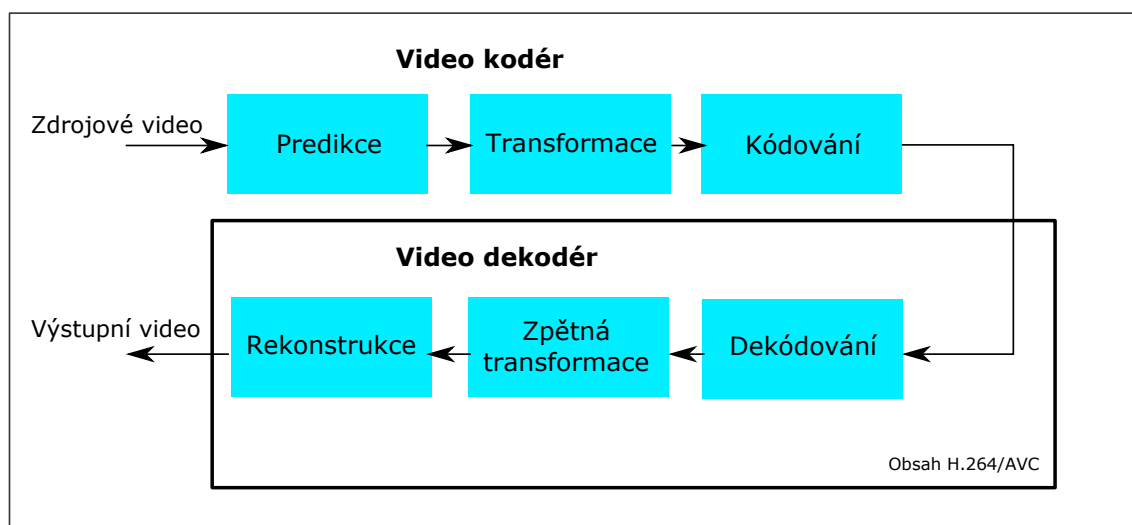
Tato práce si bere za cíl seznámit čtenáře se dvěma používanými kodeky a jejich nástupci. Konkrétně H.264, H.265 z rodiny MPEG-4 a kodeky VP8, VP9, které vyvíjí společnost Google. Dále popsat metody subjektivního hodnocení videosekvencí, při použití těchto kodeků, dle normy ITU-T P.910 a pomocí počítačové aplikace otestovat na vybraném vzorku lidí. Z výsledků se určí nejlepší kodek pro měřená rozlišení a celkový nejlepší kodek.

1 POUŽITÉ VIDEO KODEKY

1.1 H.264

H.264 je standart pro video kompresi, proces, který transformuje digitální video formát na nový, s menší velikostí. Video komprese je základním prvkem pro dnešní systémy jako například pozemní digitální vysílání (DVB-T), DVD-Video, videokonference nebo internetový streaming. Vytvoření jednotného standartu zajišťuje podporu na více zařízeních od různých výrobců. Hlavní částí je kodér, který konvertuje video do komprimovaného formátu a dekodér, jenž dělá přesný opak. [2]

Norma H.264/AVC (Advanced Video Coding) je dokument vydaný organizacemi ITU-T (International Telecommunication Union) a ISO/IEC (International Organisation for Standardisation / International Electrotechnical Commission). Definuje syntaxi pro kompresi a metody dekodéru pro vznik zobrazitelné videosekvence. Standart ovšem neudává, jak by měl vypadat kodér. Záleží pro jakou konkrétní implementaci se výrobce rozhodne. Více o obsahu normy H.264 na obrázku 1.1.



Obr. 1.1: H.264 Proces zpracování videa

Kódování

Kodér zpracovává snímky videa v makrobloky o velikosti 16x16 pixelů. Jednotlivé makrobloky se sdruží do snímku. Makrobloky se vytváří buď predikcí z předchozích, už zakódovaných dat, ze současného snímku (interpredikce), nebo z dat ostatních snímků, které se už kódování podrobily (intrapredikce). Predikce makrobloků využívá různé techniky odhadu pohybu. Výpočty se provádějí pomocí pohybových

vektorů. Tyto operace jsou výpočetně nejnáročnější částí kódování, neboť je třeba počítat s každým pixlem uvnitř makrobloku. O tom jestli se makroblok zakóduje v módu inter nebo intra, rozhoduje Rate–Distortion (míra zkreslení). Po zvolení módu se vybírá vhodný algoritmus. Používanými algoritmy jsou například Diamond search method a Hexagon search method (oba pro interpredikci). Jejich rozdíl je v rozmístění kontrolovaných pixelů. Pokud algoritmy naleznou aktuální makroblok v okolí referenčního, vypočtou mu pohybový vektor a při dalším počítání se používají pouze souřadnice tohoto vektoru, čímž se šetří data.

Vytvořený predikovaný makroblok se odečte od aktuálně zpracovávaného. Tento proces se nazývá kompenzace pohybu. Následně se provede diskrétní kosinová transformace (DCT) a entropické kódování. Výstupem DCT je soubor koeficientů, každý o dané hodnotě. V dalším kroku se koeficienty vydělí celým číslem (kvantovací parametr), z důvodu snížení přesnosti. V takto vzniklém bloku je pak většina nulových s několika nenulovými koeficienty. Čím menší kvantovací parametr volíme, tím přesnější obraz dostáváme, za cenu menší komprese.

Dekódování

Video dekodér přijme komprimovaný H.264 bitový proud, dekoduje každou část popsanou výše a extrahuje informace. Tyto informace použije při zpětném kódování a znovuvytvoření snímků. Kvantovací parametry jsou přepočteny. Každý koeficient je vynásoben hodnotou, aby výsledný blok byl co nejvěrnější originálu. Vzniklé bloky se dále skládají do makrobloků a snímků.

Profily

Pro konkrétní aplikace a požadavky různých zařízení jsou v normě H.264 definovány tzv. profily. Podle profilu dokáže dekodér určit, jaké parametry má při dekodování nastavit. [3]

Baseline profile (Základní profil) se používá pro aplikace, u kterých nevádí vyšší ztrátovost dat. Konkrétně se jedná hlavně o videokonference a mobilní zařízení. Vzorkování videa YCbCr je nastaveno na 4:2:0 při bitové hloubce 8 bitů.

Extended profile (rozšířený profil) je určen pro streamování videa. Dokáže tedy zvládat vyšší kompresi a má navíc funkce pro dokonalejší síťový přenos. Snímky SI a SP umožňují měnit bitové rychlosti. Od tohoto profilu se již používají B snímky.

Main profile (hlavní profil) slouží jako standart pro pozemní televizní DVB vysílání. Oproti předchozím profilům je zavedeno kódování typu CABAC (Context Adaptive Binary Arithmetic Coding), namísto CAVLC (Context Adaptive Variable

Length Coding). Neumožňuje ale vysílání ve vysokém rozlišení, proto byl roku 2004 vytvořen profil High.

High profile (vysoký profil) přináší řadu vylepšení. Vzorkování videa se zvedlo z původních 4:2:0 na 4:2:2 nebo dokonce na 4:4:4. Bitová hloubka vzrostla na maximálních 14 bitů. Profil High je určen pro videa ve vysokém a velmi vysokém rozlišení, od čeho se odvíjí i vyšší nároky na výpočetní techniku a dobu zpracování. Prakticky je použit například na Blu-ray discích nebo jako DVB HDTV vysílání. Rozšířený profil High 4:4:4 Predictive Profile navíc umožňuje prediktivní bezztrátové kódování či kódování každého snímku jako tří oddělených barevných ploch.

Mimo profily jsou definovány také takzvané úrovně (levels). Úrovně nastavují stupeň, na kterém může ještě daný profil pracovat. Nastavují například maximální rozlišení, snímkovací frekvenci, maximální bitovou rychlost nebo počet makrobloků za sekundu. Úrovně jsou rozděleny do 5 kategorií s několika podkategoriemi. Vždy platí, že dekodér, který zpracovává jednu úroveň, musí být schopen zpracovat všechny úrovně nižší. Maximální podporované rozlišení je 4096x2304 při 56,3 snímcích za sekundu.

1.2 H.265

Kodek H.265, označovaný také jako HEVC (High Efficiency Video Coding), vychází ze svých předchůdců jako H.264/AVC nebo ještě dřívějšího MPEG-2 Video. Byl vytvořen stejně jako H.264 organizacemi ISO/IEC a ITU-T. V lednu 2013 vyšla jeho první verze a oficiálně přijatý jako standart byl téhož roku v dubnu. V současné době se nachází ve třetí vývojové verzi.

Oproti svému předchůdci nabízí až 60% úsporu dat, při zachování stejné kvality[5]. Porovnání v tabulce 1.1.

Redukce bit ratu oproti kodeku H.264

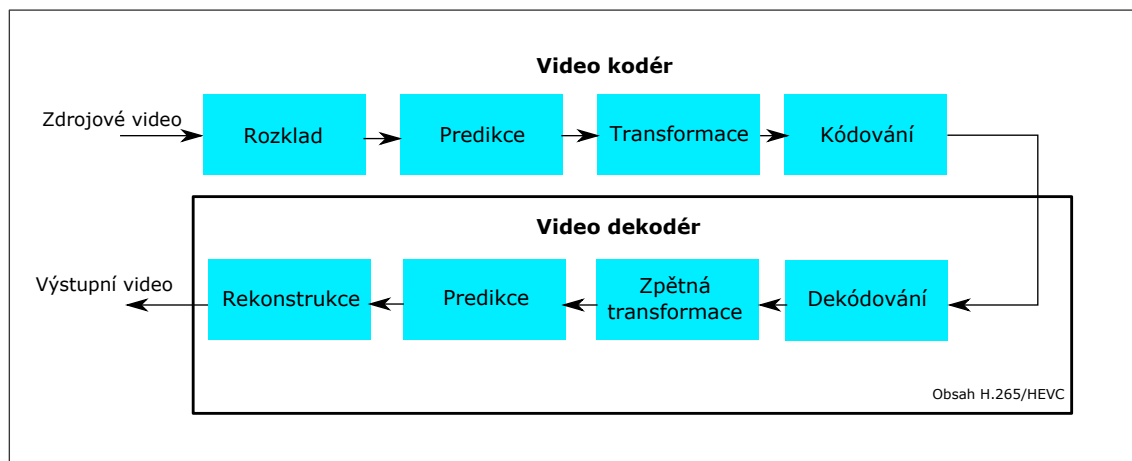
| Použitý kodek | 480p | 720p | 1080p | 4K UHD |
|---------------|------|------|-------|--------|
| HEVC | 52% | 56% | 62% | 64% |

Tab. 1.1: Zmenšení průměrného bit ratu u HEVC v porovnání s H.264/MPEG-4 [6]

Další důležitá zlepšení:

- Flexiblnější rozkládání na bloky.
- Flexibilita v predikci a transformaci velikosti bloků.
- Sofistikovanější interpolace a deblokovací filtr
- Sofistikovanější predikce, signalizace režimů a pohybových vektorů.
- Funkce na podporu efektivního paralelního zpracování.

Standart HEVC definuje syntaxi a formát komprimovaného videa, včetně metod dekódování a komprese. Návrh dekodéru není stejně jako u H.264 definován. Na obrázku 1.2 je znázorněna struktura kodéru a dekodéru.



Obr. 1.2: Struktura HEVC kodéru a dekodéru

Rozklad

Kodek HEVC umožňuje vysoce flexibilní rozklad video sekvence. Každý snímek nebo obrázek je rozložen na dlaždice a/nebo pláty. Tyto podbloky se dále rozkládají do takzvaných Coding Tree Units (CTUs). CTU je základní jednotkou kódování, obdobně jako byly u H.264 makrobloky. Maximální velikost je 64x64 pixelů. CTU se dá dále rozdělit na Coding Units (CUs) [7].

Predikce

Coding Unit (CU) je rozdělena na jeden nebo více predikčních bloků (PB). Každý z nich je predikován použitím buď intra nebo inter predikce.

Intra predikce: Každý PB se predikuje z dat sousedních bloků ve stejném snímku užitím DC predikce (průměrná hodnota PB), rovinné predikce (vlození rovných ploch do PU) nebo přímé predikce (extrapolace ze sousedních údajů).

Inter predikce: Každý PB se predikuje z dat jednoho nebo dvou referenčních snímků (může být před nebo za současným snímkem) použitím pohybové vyrovnané predikce. Pohybové vektory mají až čtvrtinové rozlišení.

Transformace a kvantování

Všechna zbytková data po predikci jsou transformována použitím Diskrétní Cosinové Transformace (DCT) nebo Diskrétní Sinové Transformace (DST). Jeden nebo více

bloků transformace o velikosti 32x32, 16x16, 8x8 a 4x4 jsou použity ve zbytkových datech v každém CU.

Kódování entropie

Bitový tok kodeku HEVC v sobě obsahuje koeficienty kvantování, informace o predikci, jako například pohybové vektory, informace o rozkladu a další informace obsažené v hlavičce. Všechny tyto elementy jsou kódovány užitím Context Adaptive Binary Arithmetic Coding (CABAC).

Profily

Verze 1 standartu HEVC definuje tři profily Main, Main 10 a Main Still Picture [5].

Main: Profil Main umožňuje při bitové hloubce 8-bitů na snímek vzorkování 4:2:0. Jedná se o klasické parametry, které využívá velké množství standartních zařízení.

Main 10: Profil Main 10 umožňuje bitovou hloubku 8–10 bitů při vzorkování 4:2:0. Vyšší bitová hloubka dovoluje větší množství barev. Pro 8-bitovou verzi získáme 256 odstínů základních barev (červená, zelená a modrá), celkem tedy 16,78 miliónů barev. Zatímco 10-bitová verze umožňuje 1024 odstínů základních barev, čemuž odpovídá 1,07 miliardy různých možností. Zvýšením počtu barev se z velké části povedlo zamezit problému s nepřesnou barevnou prezentací. Ta se projevuje viditelnými přechody mezi barvami, například pruhy na modré obloze.

Main Still Picture: Main Still Picture je rozšíření profilu MAIN. Umožňuje kódovat statický obrázek v bitové hloubce 8-bitů a vzorkováním 4:2:0. HEVC snižuje průměrný bit rate o 15,8% v porovnání s H.264, o 22,6% v porovnání s JPEG 2000 a o 43% v porovnání s JPEG[8].

Verze 2 přidala dalších 21 profilů:

Monochrome, Monochrome 12, Monochrome 16, Main 12, Main 4:2:2 10, Main 4:2:2 12, Main 4:4:4, Main 4:4:4 10, Main 4:4:4 12, Monochrome 12 Intra, Monochrome 16 Intra, Main 12 Intra, Main 4:2:2 10 Intra, Main 4:2:2 12 Intra, Main 4:4:4 Intra, Main 4:4:4 10 Intra, Main 4:4:4 12 Intra, Main 4:4:4 16 Intra, Main 4:4:4 Still Picture, Main 4:4:4 16 Still Picture, High Throughput 4:4:4 16 Intra, Scalable Main, Scalable Main 10, a Multiview Main.

Zatím nejnovější Verze 3 zavedla například profil 3D Main pro lepší podporu zobrazování 3D videí.

Obdobně jako u H.264 i zde jsou zavedeny úrovně (celkem 13, 6 základních a 7 rozšiřujících). Nejvyšší úroveň 6.2 zobrazuje video v rozlišení 8192x4320 pixelů (8K), při 120 snímcích za sekundu.

1.3 VP8

Kodek VP8 od společnosti Google je nástupcem předešlého kodeku VP7. Oproti své konkurenci, kodekům H.264 a H.265, má tu výhodu, že byl uvolněn pod licencí CC (Creative Commons), tedy za jeho komerční používání se nemusí platit poplatky. Původní verze VP8 byla vyvíjena malým týmem ve společnosti On2 Technologies.

Kodek VP8 byl navrhován tak, aby byl schopen pokrýt většinu video formátů používaných na internetu. Používá tedy standartní bitovou hloubku 8-bitů, při vzorkování 4:2:0. Maximální rozlišení je 16383x16383 pixelů.

VP8 používá bloky o velikosti 4x4 pixelů. Ty jsou transformovány nejčastěji pomocí Diskrétní Cosinové Transformace (DCT), nicméně v závislosti na módu predikce a při zmenšování 16x16 makrobloků na 4x4, se používá i Walsh-Hadamardova transformace [9].

Predikce

Kodek VP8 používá na predikci tři typy snímků. Jsou to tzv. „poslední snímky“, „zlaté snímky“ a „alternativní referenční snímky.“ Oproti ostatním kodekům je zde výhoda menší náročnosti na výpočetní techniku. Velikost bufferu je limitována třemi referenčními snímky, což ve výsledku snižuje nároky na paměť.

Stejně jako u předchozích kodeků, i zde mluvíme o inter a intra predikci. Intra predikce přináší nový mód TM_PRED (TrueMotion prediction). Ten je určen pro co nejefektivnější a nejrychlejší predikci. U inter predikce se přidal mód SPLITMV. Tento mód umožňuje kódování libovolných bluků uvnitř makrobloku, což je výhodné zvláště v případě, kdy máme uvnitř makrobloku několik rozdílných pohybových charakteristik.

Adaptabilní cyklické filtrování

Cyklické filtrování je proces, při kterém se odstraňují blokové artefakty, vzniklé převážně Diskrétní Cosinovou Transformací. VP8 přináší několik inovací. Například možnost volby nepoužít žádný cyklický filtr nebo nastavit jeho sílu. Dá se tedy určit, že pro intra kódovaný blok, se použije silnější filtr, než pro inter kódovaný Zlatý snímek s pohybovým vektorem (0,0). Z této možnosti nastavení kodéru vyplývá větší rychlost oproti konkurenci.[10]

Použití

Pro svoji rychlost a jednoduché nároky se stal kodek VP8 velmi oblíbeným nástrojem při zpracovávání internetových videí. Projekt WebM, který v sobě mají imple-

mentovány všechny hlavní internetové prohlížeče, zprostředkovává zpracování videa pomocí VP8 pro zatím poslední značkovací jazyk HTML5. Stejně tak podporují VP8 i Flash Player od společnosti Adobe nebo mobilní operační systém Android.

1.4 VP9

Otevřený kodek VP8 od společnosti Google, je nástupcem kodeku VP8, popsaného výše. Obdobně jako H.265, je určen pro pracování s videi ve vysokém rozlišení. Hlavním cílem vývoje bylo snížení bit ratu o 50% v porovnání s VP8, při zachování stejné kvality a zvýšení efektivity komprese v porovnání s standartem HEVC. V roce 2013 byl vydán profil 0, o dva měsíce později se VP9 implementoval do prohlížeče Google Chrome a na konci roku se VP9 decodér dostal do FFmpegu. V září roku 2014 Google oznámil práci na novém kodeku VP10.

Při práci s videi ve vysokém rozlišení bylo nutné vylepšit rozdělování blokových oblastí. Nově se tak zavedly superbloky, které mají velikost 64x64 pixelů. Dále se dají dělit na menší superbloky 32x32 pixelů a makrobloky 16x16 pixelů.

Profily

V aktuální verzi VP9 se nacházejí tyto čtyři profily:

Profil 0 nastavuje bitovou hloubku 8-bitů a vzorkování 4:2:0.

Profil 1 přidává podporu 4:2:2 a 4:4:4 vzorkování.

Profil 2 zvětšuje bitovou houbku na 10-bitů až 12-bitů při 4:2:0 vzorkování.

Profil 3 nechává stejnou bitovou hloubku jako Profil 2, navíc ale podporuje 4:2:2 a 4:4:4 vzorkování.

2 POUŽITÝ SOFTWARE

2.1 FFmpeg

FFmpeg je software, který sdružuje knihovny a programy pro práci s multimedialními daty. První verze vyšla v roce 2000, aktuální verze 2.8.2 je z listopadu 2015. Jeho hlavní výhody jsou dostupnost, dá se stáhnout zdarma ze stránek <http://ffmpeg.org/> Kompatibilita, i když je vyvíjen pod Linuxem, tak byl zkompilován pod velkou většinu operačních systémů a dále třeba velká podpora různých formátů. Projekt FFmpeg cílí zejména na vývojáře, neboť neobsahuje jakékoliv UI (user interface) a je obsluhován přes příkazovou řádku. FFmpeg obsahuje tyto nástroje:

- *ffmpeg* – program pro konverzi video formátů, ovládání přes příkazový řádek
- *ffserver* – multimedialní HTTP a RTSP streamovací server
- *ffplay* – multimedialní přehrávač
- *ffprobe* – analyzátor multimedialních streamů

2.2 Java

Java je objektově orientovaný programovací jazyk vyvinutý firmou Sun Microsystems roku 1995. Jedná se o jeden z nejpoužívanějších programovacích jazyků na světě. Praktické využití nachází ve všech možných oblastech, ať už jde o deskopové počítače, mobilní telefony, čipové karty nebo auta.

Mezi jeho základní vlastnosti patří [11]:

- *jednoduchost* – syntaxe je zjednodušenou verzí syntaxe jazyka C++
- *interpretace* – nezávislost na architektuře zařízení, stačí, když má zařízení implementováno tzv. Java Virtual Machine
- *správa paměti* – garbage collector dokáže automaticky vyhledávat již nepoužívaná místa v paměti a uvolňovat je pro ostatní procesy

2.2.1 JavaFX

JavaFX je softwarová platforma postavená na základech programovacího jazyka Javy od společnosti Sun Microsystems. Popud pro její vznik byla snaha o vytvoření konkurenta prostředí, jako je Adobe Flash a Microsoft Silverlight. První verze byla představena roku 2008 (JavaFX 1.0). JavaFX nabízí vývojářům snadný a efektivní nástroj k vytvoření tzv. Rich Internet applications. Jde o aplikace, zejména pro internetové stránky, které s uživatelem komunikují, tedy nabízí interakci.

Jako vhodné vývojové prostředí se dá použít NetBeans IDE nebo Eclipse. Pro jednoduchou tvorbu uživatelského rozhraní je k dispozici nástroj JavaFX Scene Builder, který stylem přetáhnutí prvku z palety do našeho prostředí umožňuje vytvářet dobře vypadající aplikace. Informace se ukládají do FXML souboru.

2.3 MySQL

MySQL je nejpopulárnější open-source databázový systém, vytvořený švédskou firmou MySQLAB a nyní vlastněný společností Oracle Corporation. [12] MySQL je multiplatformní databáze (může běžet na více počítačových platformách, např. Microsoft Windows a Mac OS X). Komunikace probíhá pomocí dotazovacího jazyka SQL (Structured Query Language). SQL je nejčastější standardizovaný jazyk používaný k databázovému přístupu. V závislosti na prostředí, můžeme zadávat SQL příkazy přímo (např. pomocí terminálu), vložit je do kódu používající jiný jazyk anebo vytvořit API, která skryje SQL syntaxi.

Ukázka SQL příkazu:

```
SELECT * FROM srnec.vysledky;
```

Logický model MySQL databáze je složen z tabulek (jedné nebo více), řádků a sloupců. Pro propojení několika tabulek slouží sloupec s příznakem primárního klíče. Hodnoty v tomto sloupci musí být unikátní a nesmí obsahovat nuly. Tímto způsobem se mohou propojit řádky se stejnou hodnotou klíče.

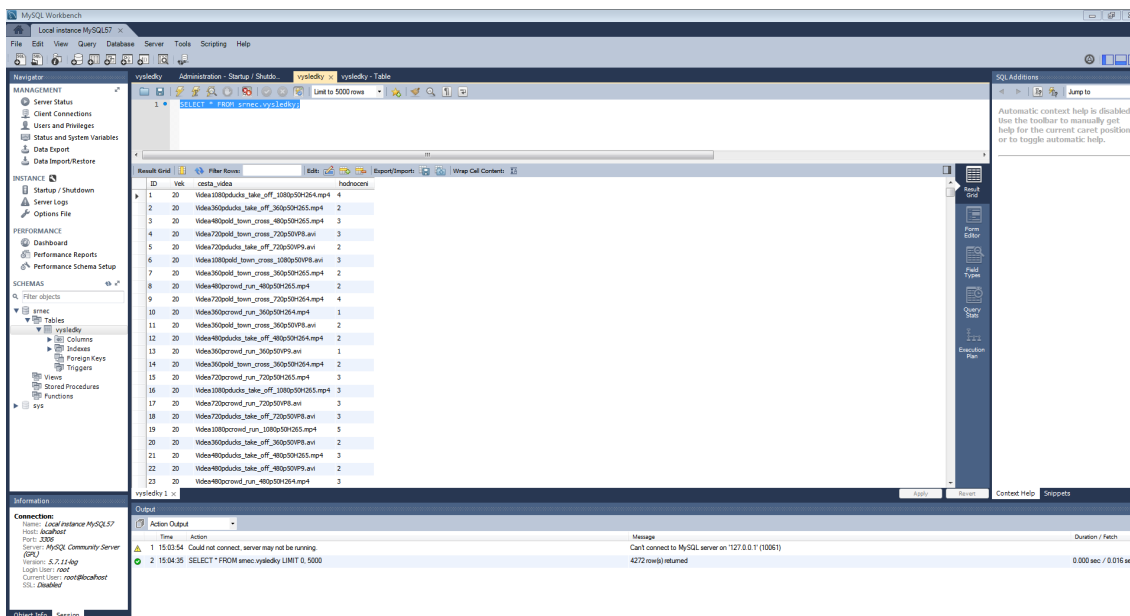
Jednou z hlavních předností MySQL serveru je jeho zaměření na rychlost zpracování požadavků a to i za cenu zjednodušení některých součástí, například zálohování. Od začátku byl vyvíjený pro práci s velkým množstvím dat. Autentizace probíhá přes uživatelské jméno a heslo. Při každém dotazu si server ověřuje, zda-li má uživatel potřebné oprávnění. Pro zabezpečení připojení se používá protokol SSL.

2.3.1 MySQL Workbench

MySQL Workbench poskytuje uživateli grafický nástroj pro práci s MySQL serverem a databázemi.[13] Zahrnuje pět základních funkcí:

- **SQL Development** – Vytváření a správa jednotlivých spojení k databázovému serveru. Provádění SQL příkazů v databázi pomocí vestavěného SQL Editoru.
- **Data Modeling** – Umožňuje zobrazovat graficky námi sledované databáze a editovat je pomocí Tabulkového editoru.
- **Server Administration** – Ovládání serveru, kde má uživatel práva administrátora.

- **Data Migration** – Přenášení dat mezi různými SQL databázemi. Podpora přenosu dat ze starších verzí MySQL na novější.
- **MySQL Enterprise Support** – Podpora podnikových MySQL produktů.



Obr. 2.1: Prostředí MySQL Workbench

3 METODY SUBJEKTIVNÍHO HODNOCENÍ

3.1 Charakteristika scény

Výběr správné scény, na které bude probíhat test, je velice důležitý. Konkrétně množství prostorových a časových vjemových informací jsou kritické parametry scény. Tyto parametry hrají klíčovou roli při určování rozpoznatelného množství video komprese, tudíž při zhoršování stupně kvality vůči originálu.

Množství sekvencí je zvoleno dle potřeb testování. Za účelem zaručení minimální spolehlivosti, je třeba vybrat alespoň čtyři různé scény. Maximální množství se volí s přihlédnutím ke snaze neunudit pozorovatele.

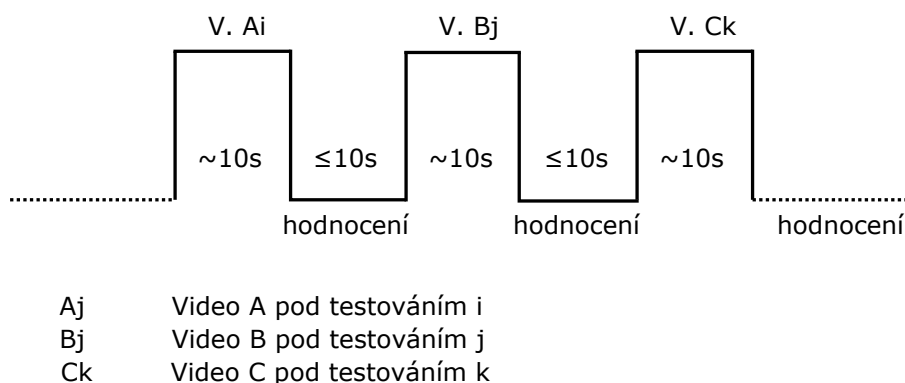
3.2 Testovací metody

Norma ITU-T definuje celkem čtyři metody pro subjektivní hodnocení video sekvencí. Jsou to Absolute category rating (ACR), Absolute category rating with hidden reference (ACR-HR), Degradation category rating (DCR) a Pair comparison method (PC). V následujících kapitolách se budu věnovat každé metodě zvlášť.

3.2.1 Absolute category rating (ACR)

Metoda absolute category rating (hodnocení nezávislých stimulů) používá k hodnocení uzavřenou stupnici. Testované sekvence jsou přehrávány postupně za sebou a hodnoceny nezávisle. Po ukončení videa v sekvenci je ponechán čas na jeho zhodnocení.

Konkrétní časové rozpětí je popsáno na obrázku 3.1. Při použití stejně dlouhého času na hodnocení u všech sekvencí by doporučená doba neměla překročit 10 s.



Obr. 3.1: Možné nastavení ACR metody

Pro hodnocení se používá tato pětibodová stupnice:

| | |
|---|--------------|
| 5 | Výborná |
| 4 | Dobrá |
| 3 | Uspokojivá |
| 2 | Špatná |
| 1 | Velmi špatná |

Dále je možné zvolit devítibodovou stupnici, pokud chceme získat přesnější výsledky. Devítibodová stupnice se používá v případě podobnosti sekvencí, i když jsou zpracovány rozdílně. Umožňuje podrobnější odlišení pozorovaných rozdílů.

Ukázka devítibodové stupnice:

| | |
|---|--------------|
| 9 | Výborná |
| 8 | |
| 7 | Dobrá |
| 6 | |
| 5 | Uspokojivá |
| 4 | |
| 3 | Špatná |
| 2 | |
| 1 | Velmi špatná |

Pokud chceme hodnotit i jiné parametry, než celkovou kvalitu, můžeme položit i otázky týkající se barev, kontrastu, plynulosti či problikávání.

Výsledné číslo se získá opakováním stejné testovací podmínky v různých místech testování.

3.2.2 Absolute category rating with hidden reference (ACR–HR)

Absolute category rating with hidden reference method (metoda hodnocení nezávislých stimulů se závislostí na skryté referenci) je stejně jako metoda ACR, vztažena k uzavřené stupnici, na níž se hodnotí nezávisle každá sekvence zvlášť. Rozdíl oproti metodě ACR je v přítomnosti původního nekódovaného videa, tzv. referenci. Tato reference je náhodně vložena do testovací sekvence a o jejím umístění ví pouze autor testu. Každé testované video musí mít vždy svoji referenci. Po samotném testování se vypočítá skóre rozdílu kvality (differential quality score (DMOS)). DMOS udává číselný rozdíl mezi sekvencí a její skrytou referencí.

Samotné hodnocení probíhá, stejně jako u ACR, po každém videu v sekvenci. Správné nastavené časů je popsáno v obrázku 3.1. Čas na hlasování by opět neměl přesáhnout deset sekund.

Hodnocení se provádí na pětibodové stupnici, stejně jako u ACR.

| | |
|---|--------------|
| 5 | Výborná |
| 4 | Dobrá |
| 3 | Uspokojivá |
| 2 | Špatná |
| 1 | Velmi špatná |

Skóre rozdílu (SA) každého pozorovatele se počítá z kódované sekvence (KS). Příslušná skrytá reference (REF) je použita při výpočtu SA pomocí následujícího vzorce:

$$SA(KS) = V(KS) - V(REF) + 5 \quad (3.1)$$

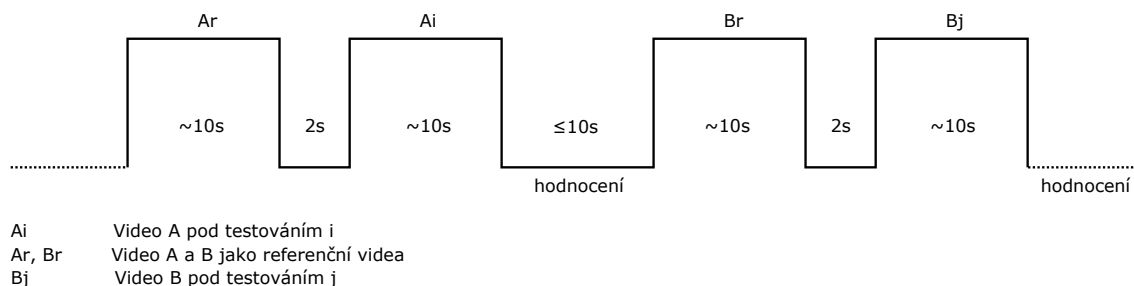
kde V je skóre pozorovatele získané ACR stupnicí. Při používání tohoto vzorce, hodnota $SA = 5$ odpovídá Výborné kvalitě, naopak $SA = 1$ značí Velmi špatnou kvalitu. Hodnota větší než 5 (tedy zpracované video bylo ohodnoceno jako kvalitnější než referenční), se považuje za platnou. Obdobně jako u ACR, i zde se může použít devítibodová stupnice, pokud jsou si testované sekvence podobné a potřebujeme detailnější měřítko.

ACR-HR metoda by měla používat referenční videa, které odborník v dané oblasti určí na stupnici výše jako Výborná, nanejvýš Dobrá.

Metoda ACR-HR se nehodí pro analýzu zhoršení objevujících se v první a poslední sekundě videosekvence. Pozorovatelova neznalost referenčního videa může způsobit, že jinak zřejmé zhoršení kvality může být přehlédnuto (například pauza předcházející konci sekvence je buď považována jako chyba zařízení nebo chyba sekvence).

3.2.3 Degradation category rating (DCR)

Degradation category rating (hodnocení degradace) oproti předchozím metodám prezentuje testované sekvence ve dvojicích. Vždy první z páru je původní, referenční video, kdežto druhé video je od referenčního upraveno námi danou testovací podmínkou. Způsob, kdy není možné videa přehrávat současně vedle sebe, je popsán na obrázku 3.2.



Obr. 3.2: Možné nastavení DCR metody

V tomto případě jsou účastníci požádáni, ať ohodnotí zhoršení druhého videa, vzhledem k referenci. Měla by být použita tato pětibodová stupnice:

| | |
|---|-----------------------------|
| 5 | Neznatelné |
| 4 | Znatelné, ale ne nepříjemné |
| 3 | Lehce nepříjemné |
| 2 | Nepříjemné |
| 1 | Velmi nepříjemné |

Výsledné číslo se získá opakováním stejné testovací podmínky v různých místech testování.

Pro menší obrazové formáty je doporučeno zobrazovat obě sekvence vedle sebe na jednom monitoru.

Výhody souběžného testování jsou takové:

- Snížení celkové doby testování.
- Jednodušší rozpoznání rozdílů mezi referencí a upravenou verzí.
- Při snížení celkové doby testování roste pozornost účastníků testu.

Při souběžném testování je velice důležité dodržovat synchronizaci; tedy obě videa musí začínat a končit na stejném snímku. Toto nevyklučuje testování sekvencí o různých bit ratech, jen musí být zajištěno vhodné převzorkování.

Sekvence musí být zobrazeny ve dvou oknech vedle sebe na pozadí 50% šedé barvy. Pro zmenšení potřebného pohybu očí při přeskokování mezi okny, by vzdálenost pozorovatele měla být 8H, kde H je výška videa. Minimální velikost použitého monitoru musí být 14 palců.

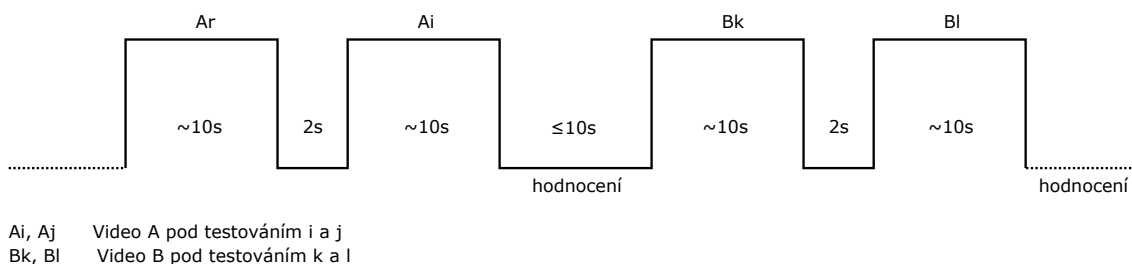
Pozorovatel vždy ví, kde se nachází referenční video (standardně vlevo).

3.2.4 Pair comparison method (PC)

Název pair comparison method (metoda porovnávání dvojic) napovídá, že testované sekvence se přehrávají ve dvojicích. Ta stejná sekvence se přehraje nejprve za první

testovací podmínky, následně za druhé.

Testovací podmínky (A, B, C, atd.) se zpravidla mísí ve všech $n(n-1)$ možných kombinacích AB, BA, CA, atd. . . Tudíž všechny dvojice se zobrazí v obou možných variantách (AB, BA). Po přehrání každého páru se určuje, který ze dvojice byl lepší. Na hodnocení je doporučeno použít čas maximálně 10 sekund. Schéma hodnocení je zobrazeno na obrázku 3.3



Obr. 3.3: Možné nastavení PC metody

Pro videa v nízkém formátu se doporučuje přehrávat obě videa současně na jednom monitoru. Více v DCR.

V této metodě nemusíme opakovat stejnou testovací podmínku, jelikož je již zajištěna opakováním dané dvojice, jen v jiném pořadí.

3.3 Porovnání metod

Klíčovým faktorem při výběru správné metody je, zdali chceme porovnávat testovací videa s referenčním, jako je to u metody DCR, nebo použijeme metody bez reference (ACR, ACR–HR (skrytá reference), PC). Tato druhá kategorie metod netestuje transparentnost a věrnost.

DCR metoda by měla být použita v případě testování věrnosti kódovaného videa k videu původnímu. Tento důležitý faktor znamená časté použití metody u systému s vysokou kvalitou. DCR je základní metodou popsanou v normě ITU-R BT.500-9 [4], při posouzení televizního obrazu, které často bývá považováno za nejvyšší stupeň kvality ve videotelefonii a videokonferencích. Slovní komentáře na DCR hodnotící stupnici (Neznatelné, Znatelné, . . .) mají smysl, pokud hodnotíme zhoršování kvality, stejně tak, při zkoumání věrnosti.

ACR metoda je rychlá a jednoduchá. Její výhoda je, že testované podněty jsou zobrazovány podobně, jak se s nimi testovaný člověk může potkat například doma (např. zhoršení kvality při sledování internetového videa).

ACR–HR pokrývá všechny výhody ACR s ohledem na rychlost a jednoduchost prezentace. Hlavní výhodou ACR–HR oproti ACR je porovnání se skrytou referencí. Pozorovatel netuší, že jedno z hodnocených videí v sekvenci je referenční. Tento fakt

snižuje dopad zkreslení (např. pozorovateli se buď líbí, nebo nelíbí referenční video), kvalitu referenčního videa (např. rozdíly v použité kameře) a v úrovni použitého monitoru (profesionální proti běžnému). ACR–HR se hodí zejména pro velké experimenty, kde je zaručeno, že referenční video je expertem ohodnoceno alespoň jako „dobré.“ Nicméně ACR–HR by mohla být necitlivá k některým zhoršením, které jsou snadno detekovatelné metodami přímého porovnávání, jako například DCR. Ukázkovým příkladem je plynulý pokles v ziskovosti barev (např. útlum barevnosti).

Hlavní výhodou PC metody je její vysoká rozlišující síla, jenž se hodí v momentě, kdy se testovací sekvence velice podobají.

Pokud máme v plánu testovat velké množství videí, PC metoda není nejvhodnější kvůli její časové náročnosti. V tomto případě se doporučuje nejprve použít ACR nebo DCR test na menším množství osob a následně použít PC metodu na sekvencích s velmi podobným hodnocením.

3.4 Referenční podmínky

Výsledky testu subjektivního hodnocení často nezávisí jen na testovaných sekvencích, ale také na celkovém rozptylu kvality testovaných videí či zkušenosti a očekávání zúčastněných. Pro zachycení některých těchto efektů se může použít falešná testovací podmínka, nebo falešné referenční video.

Pro správné vyhodnocení musí být stejné podmínky pro všechny. Jednak velikost a typ monitoru by měl vyhovovat testovanému účelu. Pokud jsou sekvence testovány skrze PC, je nutné popsat specifikace soustavy jako například použitá grafická karta, rozlišení monitoru atd.

Preferovaným formátem zobrazení je použití celé obrazovky (full screen). Pokud tohle z nějakého důvodu nelze zajistit, sekvence se zobrazí v okně, kde barva pozadí bude 50% šedá.

3.5 Zpracování a přehrávání videa

Máme dvě metody pro získání testovacího videa ze zdrojového:

- Přenášením videa v reálném čase kódovacím zařízením a jeho okamžité zobrazení, zatímco účastníci pozorují a odpovídají.
- Offline zpracování videa potřebným kóděrem, jeho nahrání a uschování pro vyhodnocování. Tímto vznikají nahrávky, které se testují vybranou metodou.

CRT, LCD, Plasma, Dataprojektor, nebo monitor jsou všechno přípustné zobrazovací zařízení. Jejich nastavení by vždy mělo korespondovat s účelem testování.

3.6 Účastníci testu

Počet všech účastníků testu by se měl pohybovat v rozpětí od 4 do 40. Čtyři je absolutní minimum ze statistického hlediska, zatímco větší množství jak 40 nebývá zpravidla zapotřebí.

Vybraný počet pro specifický test by měl odpovídat vypovídající hodnotě účastníků a rovnoměrnému rozložení potřebných věkových kategorií.

Obecně, experimentu by se mělo účastnit alespoň 15 lidí. Neměli by mít přímé zkušenosti s hodnocením obrazové kvality a neměli by být zkušenými hodnotiteli.

Nicméně při raných fázích vývoje a při prvních pilotních testech před testováním velké skupiny lidí je dobré, si vše vyzkoušet na malé skupince expertů (4-8). Ti mohou poskytnout orientační výsledky.

3.6.1 Pokyny pro účastníky a školení

Před samotným začátkem experimentu je dobré seznámit účastníky s důvodem testování, včetně popisu testovaných podmínek. Dále je třeba seznámit testované subjekty s určenou stupnicí, na které budou videa hodnotit. Před testováním je možné seznámit účinkující s ukázkovými sekvencemi, které reprezentují rozsah testování a pomohou utvořit lepší představu o tom jak hodnotit. Tyto videa by však měla být odlišná od testovaných. Nic nesmí naznačovat tomu, že nehorší kvalita u ukázkových sekvencí se shoduje s nehorší kvalitou u sekvencí testovaných.

Otázky týkající se důvodu a způsobu testování se musí zodpovědět před začátkem a s opatrností neprozradit nic, co by mohlo způsobit zkreslení výsledků.

Ukázka úvodu pro ACR a ACR–HR metodu

Dobré ráno, díky že jste přišli. V tomto experimentu uvidíte na obrazovce před sebou krátké video sekvence. Po každém videu prosím ohodnoťte jeho kvalitu na pětibodové stupnici zakroužkováním odpovídající hodnoty.

| | |
|---|--------------|
| 5 | Výborná |
| 4 | Dobrá |
| 3 | Uspokojivá |
| 2 | Špatná |
| 1 | Velmi špatná |

Před udělením hodnocení, si prosím prohlédněte video až do konce.

4 PRAKTICKÁ ČÁST

4.1 Popis použitých zařízení

Pro účely našeho testování byla rezervována učebna SE5.113, obsahující dostatečně množství výpočetní techniky. Všechny modely byly totožné se stejným softwarovým a hardwarovým vybavením. Videá se zobrazovala na 22" monitorech Eizo o rozlišení 1920x1080 pixelů (FullHD). Počítače běžely na operačním systému Windows 7.

4.2 Realizace MySQL serveru

Pro potřeby našeho testování byl vytvořen MySQL server, umístěný v prostorách školy. Na něm byla vytvořena databáze s přiděleným uživatelským jménem a heslem. Ostatní nastavení databáze bylo ponecháno na autorovi práce. Vzdálený přístup byl zajištěn pomocí webového nástroje phpMyAdmin, který poskytoval všechny potřebné funkce.

Pro naše testování jsem vytvořil tabulku *vysledky* a do ní čtyři sloupce. První se jmenuje *ID* s funkcí *AUTO_INCREMENT* zajišťující, že tomuto sloupci není potřeba posílat nějakou hodnotu. Druhý sloupec s názvem *Vek* nese informace o věku, zadanou na úvodní obrazovce programu. Třetí sloupec *cesta_video* slouží k jedinečné identifikaci právě testovaného videa. A konečně v posledním sloupci *hodnoceni* se nachází udělená známka.

| ID | Vek | cesta_video | hodnoceni |
|----|-----|--|-----------|
| 1 | 20 | Videa1080pducks_take_off_1080p50H264.mp4 | 4 |

Tab. 4.1: Ukázka jednoho řádku v databázi

4.2.1 Použité příkazy

Pro práci s databází jsme používali několik příkazů. Jejich odesílání probíhalo přes náš JavaFX program pomocí konektoru JDBC dostupného na: <https://www.mysql.com/products/connector/>. Jako příklad uvedu kód odeslaný po stisku hodnocení "1".

```
String cm1 = "INSERT INTO vysledky (Vek, cesta_video, hodnoceni)"
+ "VALUES ('" + rvek + "', '" + getInfo0Videu() + "', " + "1)";
```

V tomto kódu si ukládám do proměnné `cm1` typu `String` příkaz, který se odešle na server pomocí metody `.executeUpdate(cm1)`. První řádek určuje do jaké tabulky a do jakých sloupců se bude zapisovat. Druhý řádek s klíčovým slovem `VALUES` obsahuje proměnou `rvek`, do které vložil uživatel svůj věk na úvodní obrazovce. Metoda `getInfo0Videu()` vrátí název testovaného videa a číslo "1" jako výsledek.

Další příkaz slouží k výpisu určených záznamů pro vytvoření výsledků:

```
SELECT * FROM 'vysledky' where cesta_video like '%crowd%'
AND cesta_video like '%360%' AND cesta_video like '%h264%'
```

V překladu, příkaz načte všechny řádky z tabulky *vysledky*, které obsahují ve sloupci *cesta_video* klíčová slova `crowd`, `360` a `h264`. Ze zobrazeného řádku si převezmeme pouze hodnoty ze sloupce *hodnoceni* a můžeme je použít pro další operace. Tímto způsobem se dají zobrazit všechny výsledky dle našich kritérií.

4.3 Výběr metody

Pro testování námi vybraných tří video sekvencí jsme se rozhodli použít metodu Absolute category rating (ACR) 3.2.1. Důvody pro její výběr byly její vhodnost pro větší množství testovaných videí a větší počet zúčastněných, snadná implementace a možnost použít pětibodovou hodnotící stupnici. Vzhledem k tomu, že účastníci testování nebyli dříve seznámeni s problematikou kódování videí, zvolení devíti bodové stupnice by dávalo zbytečně veliký rozsah.

4.4 Testované videosekvence

Pro potřeby testování byly vybrány čtyři videosekvence, které se stáhli z webové adresy <https://media.xiph.org/video/derf/>. Vhodné parametry byly určeny jako formát 16:9 s rozlišením 1920x1080 pixelů. Konkrétně se jedná o tyto videa:

- `crowd_run` – 50 FPS, 500 snímků, vzorkování 4:2:0
- `old_town_cross` – 50 FPS, 500 snímků, vzorkování 4:2:0
- `ducks_take_off` – 50 FPS, 500 snímků, vzorkování 4:2:0

4.4.1 Proces kódování

Před samotným procesem kódování, bylo nejprve nutné stanovit parametry. Prvním základním parametrem je rozlišení. Rozhodli jsme se pro tyto standartní rozlišení:

- 1080p: 1920x1080
- 720p: 1280x720
- 480p: 854x480
- 360p: 640x360

Od původního plánu testovat i v rozlišení 3840x2160 (4K) jsem byl nucen odstoupit, z důvodů nedostatečného výkonu mého počítače, který nezvládal přehrávání videa v takto vysokém rozlišení.

Dalším krokem bylo nastavení bit ratu a počtu snímků za sekundu. Potřebné informace jsou dostupné na webu YouTube v rámci jejich doporučení (Recommended upload encoding settings). Konkrétní hodnoty jsou obsaženy v tabulce 4.2 a dostupné na adrese: <https://support.google.com/youtube/answer/1722171?hl=en-GB>.

| rozlišení | bitrate (25, 25, 30 fps) | bitrate (48, 50, 60 fps) |
|-----------|--------------------------|--------------------------|
| 1080p | 8 Mbps | 12 Mbps |
| 720p | 5 Mbps | 7.5 Mbps |
| 480p | 2.5 Mbps | 4 Mbps |
| 360p | 1 Mbps | 1.5 Mbps |

Tab. 4.2: Doporučené hodnoty bit ratu pro daná rozlišení

| kodek | profil |
|-------|-----------|
| H.264 | High |
| H.265 | Main |
| VP8 | Profile 0 |
| VP9 | Profile 0 |

Tab. 4.3: Použité profily kodeků

Pro kódování videa byla použita platforma FFmpeg 2.1 ve verzi 20151103-git-6df2c94. Pro kodeky H.264 a H.265 byl zvolen kontejner .mkv a pro kodeky VP8 a VP9 byl zvolen .avi. Obecný zápis pro kódování videa vypadá takto:

```
ffmpeg -i zdroj.yuv parametry_kódování výstup.mp4
```


Použité příkazy jsou uvedeny v tabulce.

| | |
|-------------------|--|
| -c:v libx264 | nastavení pro kódování do kodeku H.264 |
| -c:v libx265 | nastavení pro kódování do kodeku H.265 |
| -c:v libvpx | nastavení pro kódování do kodeku VP8 |
| -c:v libvpx-vp9 | nastavení pro kódování do kodeku VP9 |
| -b:v 5000k | nastavení bit ratu |
| -bufsize 5000k | párový příkaz pro -b:v |
| -vf scale=640:360 | nastavení rozlišení |

Tab. 4.4: Použité příkazy

Výsledný příkaz vypadá například takto:

```
ffmpeg -i station2_1080p25.yuv -c:v libvpx-vp9 -b:v 5000k  
-bufsize 5000k -vf scale=1280:720 station2_720p25VP9.avi
```

Tímto způsobem jsem kodovali celkem 48 videí (4 rozlišení, 4 kodeky, 3 videa).

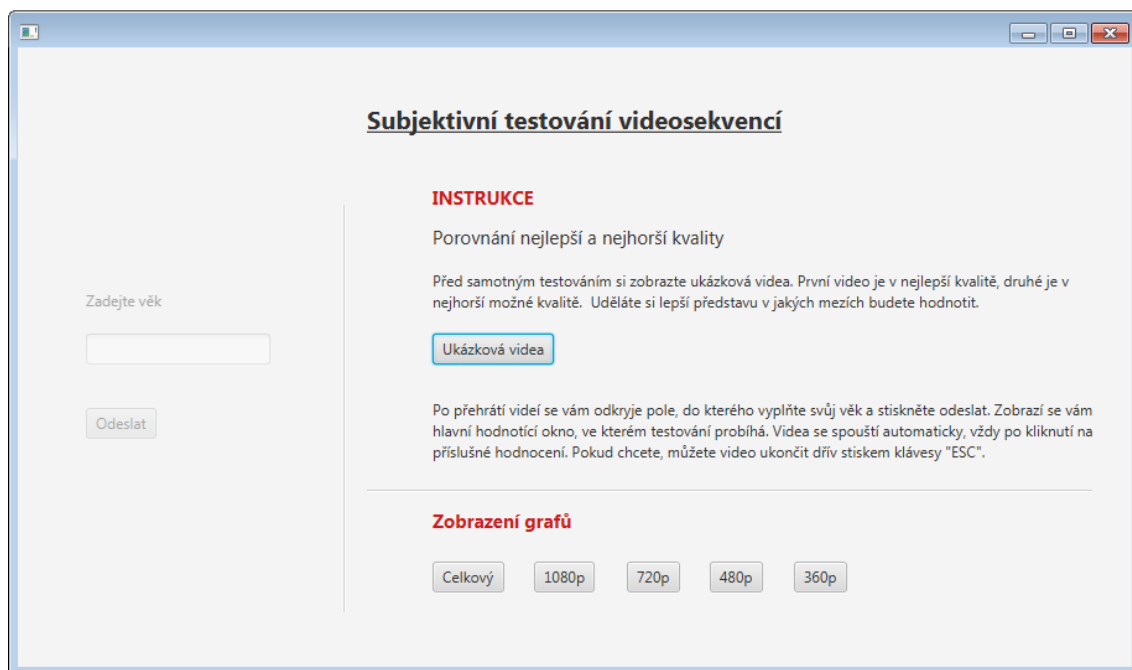
4.5 Testovací aplikace

Pro vytvoření testovací aplikace bylo použito vývojové prostředí Eclipse a programovací jazyk JavaFX. Na školním serveru zároveň běžel MySQL server, na který se odesílaly výsledky.

4.5.1 Ovládání a funkce aplikace

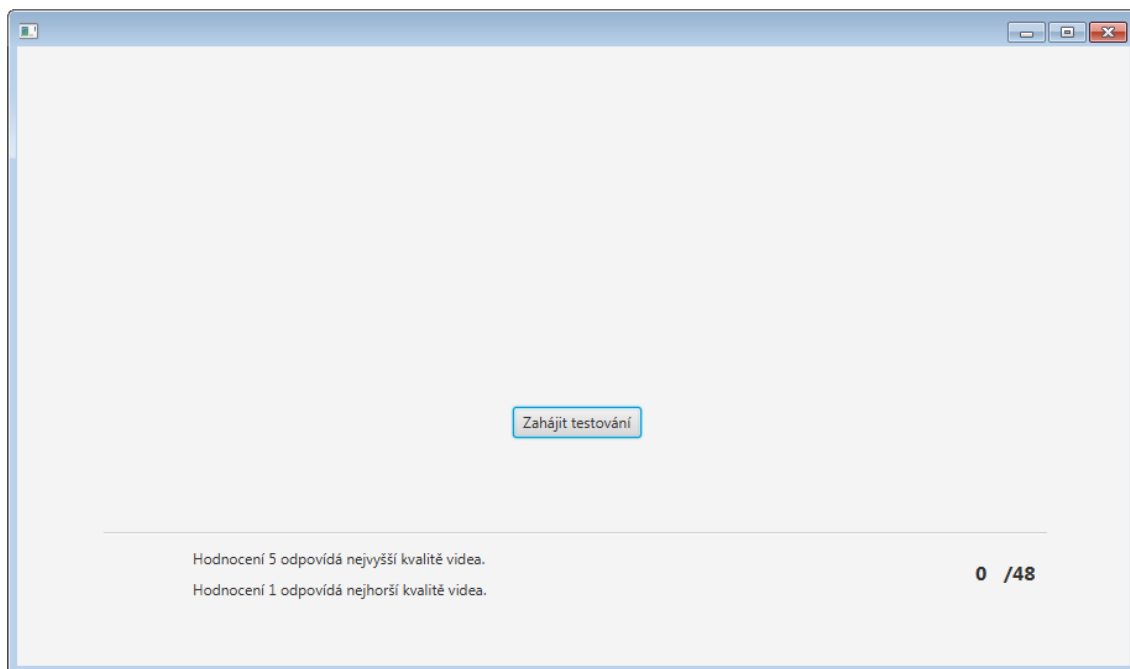
Aplikace po prvním spuštění obsahuje instrukce: *Před samotným testováním si zobrazte ukázková videa. První video je v nejlepší kvalitě, druhé je v nejhorší možné kvalitě. Uděláte si lepší představu v jakých mezích budete hodnotit. Po přehrání videí se vám odkryje pole, do kterého vyplňte svůj věk a stiskněte odeslat. Zobrazí se vám hlavní hodnotící okno, ve kterém testování probíhá. Video se spouští automaticky, vždy po kliknutí na příslušné hodnocení. Pokud chcete, můžete video ukončit dřív stiskem klávesy "ESC"*

Dále je zobrazeno tlačítko *Ukázková videa*, které spustí video crowd_run.mp4 v rozlišení 1080p a kodeku H.264. Po skončení se spustí stejné video v rozlišení 360p. Ve spodní části jsou tlačítka na zobrazení grafů. Data pro grafy se načítají po kliknutí na tlačítko ze všech doposud získaných dat. Tlačítko *Celkový* zobrazí výsledky všech rozlišení dohromady. Další tlačítka zobrazí graf vždy pro vybrané rozlišení.



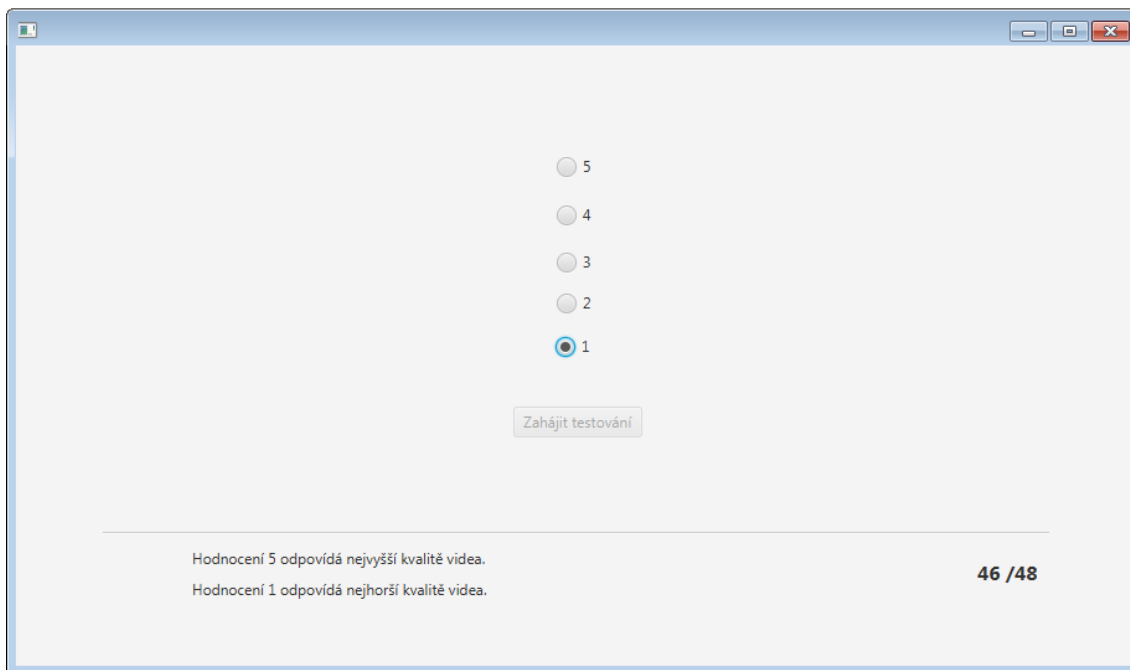
Obr. 4.1: Úvodní obrazovka

Po přehrání Ukázkových videí se odkryje panel pro zadání věku a tlačítko odeslat. Po jeho stisknutí je ustanoveno spojení s MySQL serverem a okno se přepne na jiné, určené k hodnocení.



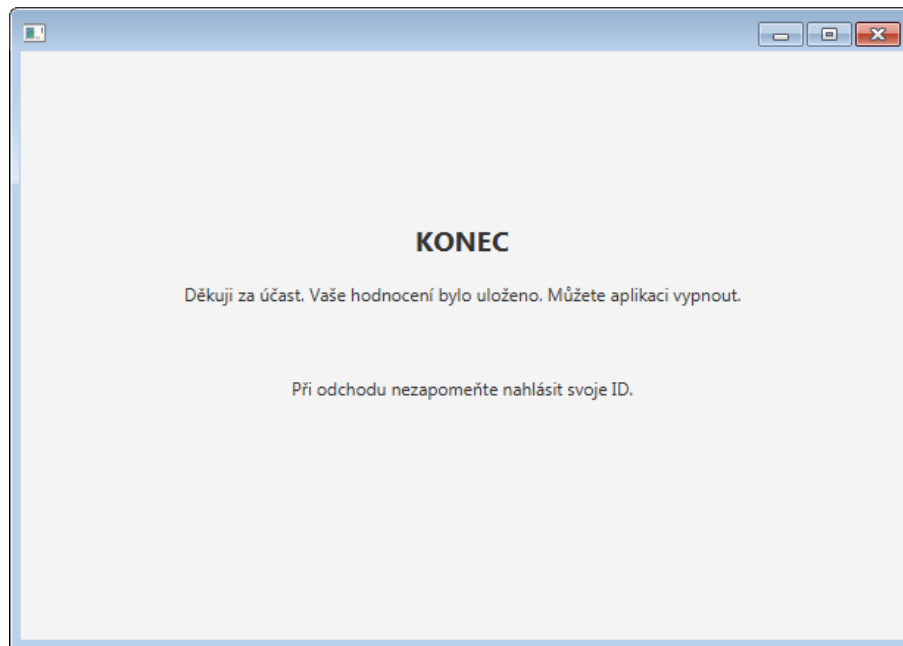
Obr. 4.2: Začátek testování

Testovací obrazovka obsahuje počítadlo pro lepší představu, kolik videí zbývá do konce, instrukce k bodovému hodnocení a tlačítko *Zahájit testování*. Po jeho stisku se spustí jedno vybrané video ze všech. V kódu programu je ošetřeno, aby se žádné video nezobrazilo dvakrát a aby každý z účastníků měl soubor poskládaný zcela náhodně.



Obr. 4.3: Průběh testování

Po přehrání prvního videa se už zobrazí hodnotící pětibodová stupnice. V momentě kliknutí na zvolené hodnocení se spouští další video a výsledek se posílá na MySQL server, kde se vytvoří nový řádek. Ohodnocení všech videí zobrazí poslední obrazovku s poděkováním a rozloučením.



Obr. 4.4: Video station2



Obr. 4.5: Video crowd_run



Obr. 4.6: Video ducks_take_off



Obr. 4.7: Video old_town_cross

4.6 Průběh testování

Testování se zúčastnilo celkem 89 studentů Fakulty elektrotechniky a komunikačních technologií VUT v Brně. Věkový rozptyl byl od 18 do 28 let a průměrný věk byl 20 let. Testování probíhalo tři dny 22.4., 28.4 a 29.4. v prostorách multimediální učebny SE5.113. Zájemci se hlásili vždy na jeden, jim vyhovující termín. Pro otestování jedné skupiny byla vyhrazena půlhodina času, ve skutečnosti však mnozí odcházeli již po patnácti minutách. Před a ani po testování nebyla účastníkům vysvětlena problematika kodeků pro získání co nejvíce nezávislých výsledků.

Během testování se nevyskytla žádná komplikace. Jak vytvořený program, tak počítače a server fungovali dle očekávání a bez problémů.

5 VÝSLEDKY TESTOVÁNÍ

Pro zpracování celkem 4272 záznamů byly použity tři základní statistické funkce, medián, modus a aritmetický průměr. Záznamy byly exportovány do programu Microsoft Excel a odtud vypočítány všechny uvedené hodnoty. Grafy použité v této práci jsou vytaženy z testovacího programu.

5.1 Rozlišení 1080p



Obr. 5.1: Výsledky hodnocení u jednotlivých videí pro rozlišení 1920x1080

| kodeky | H.264 | VP8 | H.265 | VP9 |
|--------|--------------|------------|--------------|------------|
| medián | 4 | 4 | 5 | 5 |
| modus | 5 | 4 | 5 | 5 |
| průměr | 4,240 | 3,940 | 4,487 | 4,498 |

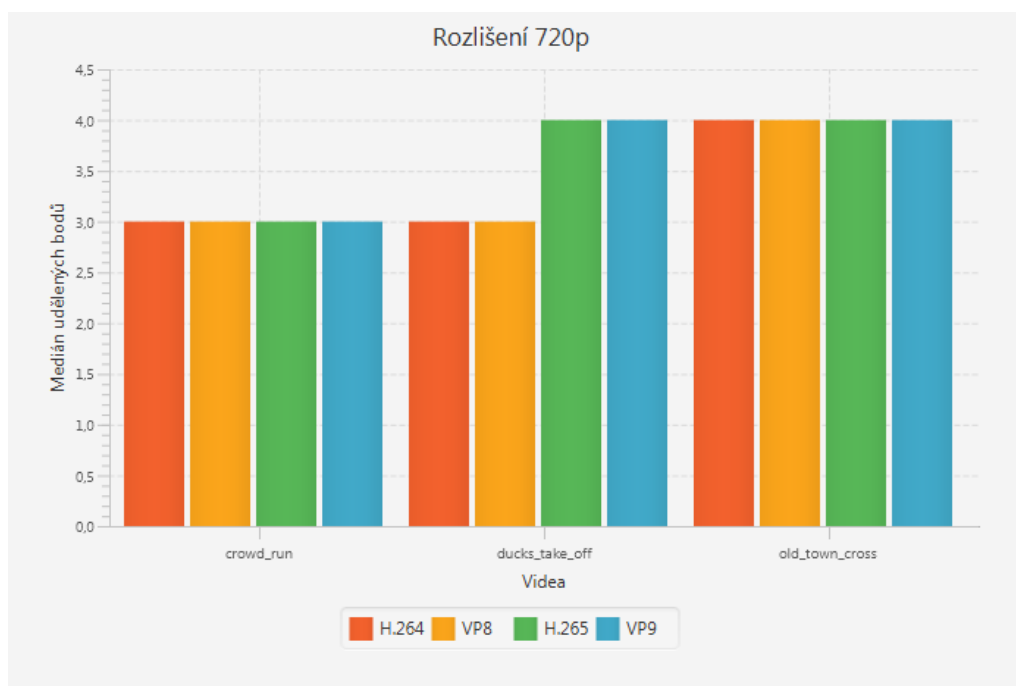
Tab. 5.1: Výsledky pro rozlišení 1920x1080

Z výsledků je patrné, že u nejvyššího testovaného rozlišení nejlépe dopadly největší kodeky H.265 a VP9. Z průměru vidíme, jaký minimální rozdíl mezi nimi je. U tohoto rozlišení může nastat problém v případě nedostatečného výpočetního výkonu zařízení, které se projevilo při testování provedeném v semestrální práci.

V tomto případě počítače zvládaly dekódovat všechny kodeky stejně dobře a nebyl tedy některý znevýhodněn.

Na třetím místě skončil, ostatně jako u všech ostatních rozlišení, kodek H.264 a na posledním místě kodek VP8. Zajímavé je, že výsledky jsou velice podobné pro každé video. Rozdílnost videí v tomto případě neměla vliv.

5.2 Rozlišení 720p



Obr. 5.2: Výsledky hodnocení u jednotlivých videí pro rozlišení 1280x720

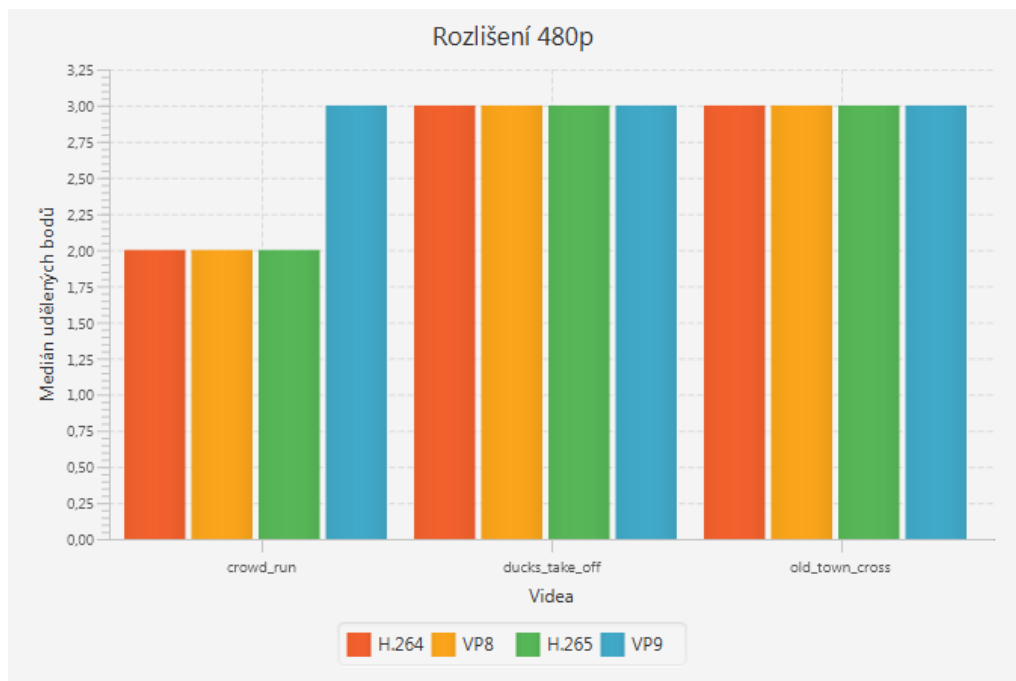
| kodeky | H.264 | VP8 | H.265 | VP9 |
|--------|--------------|------------|--------------|------------|
| medián | 3 | 3 | 4 | 4 |
| modus | 3 | 3 | 4 | 4 |
| průměr | 3,318 | 3,172 | 3,581 | 3,573 |

Tab. 5.2: Výsledky pro rozlišení 1280x720

Stejně jako u rozlišení 1080p, výsledky moderních kodeků H.265 a VP9 jsou velice vyrovnané. Jenom u tohoto rozlišení se však povedlo kodeku H.265 být lepší než VP9. Zajímavé že videa old_town_cross a crowd_run mají pro všechny kodeky vyrovnané výsledky. Obě videa jsou dynamický záběr. Na videu crowd_run je veliké množství pohybujících se objektů, pozorovatel si všímá detailů. Na videu old_town_cross je

záběr z letadla. Na scéně není tolik pohybujících se objektů a pokud ano, jsou v porovnání se scénou malé.

5.3 Rozlišení 480p



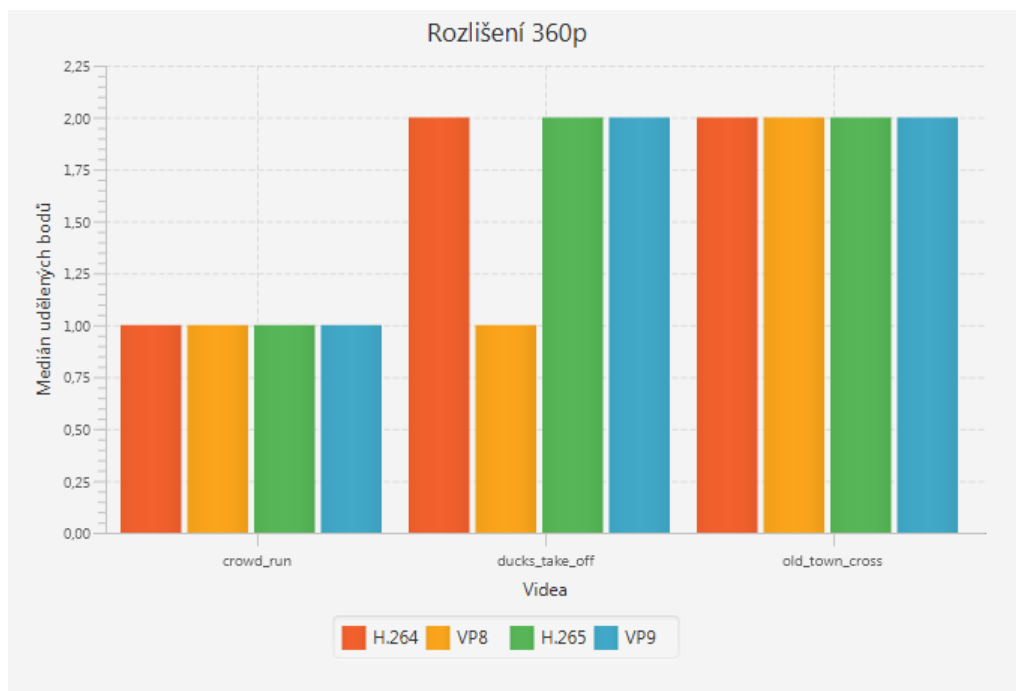
Obr. 5.3: Výsledky hodnocení u jednotlivých videí pro rozlišení 854x480

| kodeky | H.264 | VP8 | H.265 | VP9 |
|--------|--------------|------------|--------------|------------|
| medián | 3 | 3 | 3 | 3 |
| modus | 3 | 3 | 3 | 3 |
| průměr | 2,727 | 2,678 | 3,000 | 3,116 |

Tab. 5.3: Výsledky pro rozlišení 854x480

U rozlišení 480p jsou zajímavé výsledky u kodeků H.264 a VP8. Jsou si velice podobné a kodek VP8 nezaostává o několik desetin jako tomu je u ostatních rozlišení. Další zajímavostí je výraznější vítězství kodeku VP9 a to jak celkově, tak u videa crowd_run.

5.4 Rozlišení 360p



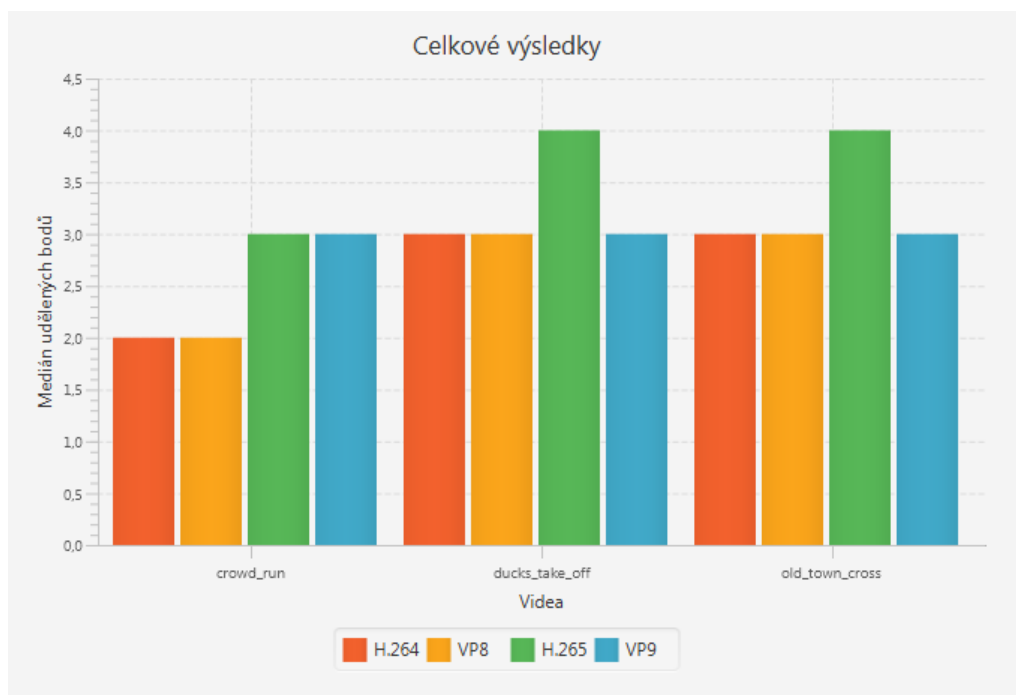
Obr. 5.4: Výsledky hodnocení u jednotlivých videí pro rozlišení 640x360

| kodeky | H.264 | VP8 | H.265 | VP9 |
|--------|--------------|------------|--------------|------------|
| medián | 1 | 1 | 1 | 2 |
| modus | 1 | 1 | 1 | 1 |
| průměr | 1,610 | 1,476 | 1,693 | 1,794 |

Tab. 5.4: Výsledky pro rozlišení 640x360

U nejnižšího testovaného rozlišení vede kodek VP9. Jako jediný má medián 2. Zajímavostí jsou dobré výsledky kodeku H.264, zejména u videa ducks_take_off. Jedná se o video se statickým záběrem na vodní hladinu, ze které odlétá hejno kachen. Je tedy velice dobře vidět, jak si který kodek dokáže poradit s pohybem a rychlým překleslováním obrazu.

5.5 Celkové výsledky



Obr. 5.5: Celkové výsledky hodnocení u jednotlivých videí

| kodeky | H.264 | VP8 | H.265 | VP9 |
|--------|--------------|------------|--------------|------------|
| medián | 3 | 3 | 3 | 3 |
| modus | 3 | 3 | 4 | 4 |
| průměr | 2,974 | 2,816 | 3,190 | 3,245 |

Tab. 5.5: Celkové výsledky pro všechna rozlišení

Z celkového hodnocení pro všechna rozlišení dohromady lze určit, že jako nejlepší kodek byl vyhodnocen VP9. Pro videa s vysokým rozlišením je výhoda moderních kodeků zcela jasná. Pro nižší rozlišení se rozdíly mírně snižují, zejména u kodeku H.264. Z tohoto výsledku může plynout například vhodnost tohoto kodeku pro mobilní zařízení, které nemají výkon pro dekódování nových kodeků a nemají velikou obrazovku s vysokým rozlišením.

Dá se říci, že rozdíly mezi kodeky nejsou nijak veliké, ovšem jsou při soustředěném pozorování viditelné. Pokud tedy naše multimediální zařízení má dostatečný výkon, doporučil bych používání kodeku VP9. Je ale překvapivé, že obě testované verze VP se pohybují na opačných koncích výsledného hodnocení. Kodeky H.264 a H.265 dosáhly podobnějšího hodnocení.

5.6 Porovnání výsledků

V mé práci jsem se pokusil podle vybrané testovací metody určit, který kodek je nejlepší. Samozřejmě nejsem jediný kdo se o něco takového pokusil, a proto v této kapitole udělám krátký souhrn podobných měření.

Práce *Comparison of compression efficiency between HEVC/H.265 and VP9 based on subjective assessments* [14] testuje čtyři videa v UltraHD rozlišení, kódované kodeky H.264, H.265 a VP9. Výsledky obsahují i porovnání PSNR (odstup signál–šum), mě však zajímaly výsledky subjektivního hodnocení, kterého se zúčastnilo 24 osob. I když byla použita jiná metoda, stále se hodnotila vizuální kvalita při různém bitratu na sto bodové stupnici. Jako nejlepší byl vyhodnocen kodek H.265. Jako druhý byl vyhodnocen H.264 s jen velice malým náskokem nad VP9.

V další práci *Subjective quality assessment of H.265 versus H.264 Video Coding for High-Definition Video Systems* [15] hodnotí účastníci metodou ACR–HR kodeky H.264 a H.265 ve třech rozlišeních 1080p, 1080i a 720p. I zde, dle očekávání měl kodek H.265 lepší výsledky, než jeho předchůdce.

Ve třetí práci *Quality Evaluation of HEVC and VP9 Video Compression in Real-Time Applications* [16] se hodnotila kvalita osmi videí, v rozlišeních 1080p a 720p. Z výsledků vyplynul jako vítěz kodek H.265 následovaný VP9 a H.264.

Právě dominance kodeku H.265 se při mém testování nepotvrdila, nicméně rozdíl mezi jeho konkurentem VP9 byl velice malý. Jisté je, že moderní nástupci podávají lepší výsledky, jak po stránce komprese, stejně tak jako výsledné kvality.

6 ZÁVĚR

V této semestrální práci byly popsány moderní používané video kodeky H.264, H.265, VP8 a VP9. Jaké jsou mezi nimi rozdíly, jaké používají profily nebo jak funguje predikce. V další byly podrobně popsány metody subjektivního hodnocení video sekvencí dle normy ITU–T P.910. Jde o metody ACR, ACR–HR, DCR a PC.

Pro potřeby testování byly vybrány tři videa ve FULL HD rozlišení a postupně převedeny, pomocí programu FFmpeg, na výše zmíněné kodeky. V praktické části byla vybrána metoda ACR pro zpracování a implementování do počítačové aplikace. Ta byla vytvořena v jazyce JavaFX a umožňuje jednoduché a správné přehrávání požadovaných sekvencí, jejich ohodnocení, odeslání výsledků na databázový server a zobrazení výsledků v podobě grafů.

Výsledné testování probíhalo v jedné místnosti, kdy každý účastník seděl u svého počítače a subjektivně hodnotil pomocí testovací aplikace. Celkem se testování zúčastnilo 89 studentů.

Ze získaných výsledků vyšel vítězně kodek VP9 následovaným kodekem H.265, H.264 a nejhůře dopadl kodek VP8. Novější kodeky H.265 a VP9 nabízejí větší datovou úsporu než jejich předchůdci a kvalitnější vykreslení obrazu, ovšem za cenu vyšších nároků na výpočetní kapacitu.

LITERATURA

- [1] *Cisco předpovídá do roku 2019 pětinasobný nárůst objemu dat přenášených po internetu ve střední a východní Evropě a zdvojnásobení průměrné rychlosti připojení* [online]. Praha, 2015 [cit.2015-12-05]. Dostupné z URL: <<http://www.cisco.com/web/CZ/about/news/2015/20150608.html>>.
- [2] RICHARDSON, Iain E a Iain E RICHARDSON. *The H.264 Advanced Video Compression Standard*. nd ed. Chichester, West Sussex, UK: Wiley, 2010, xxx, 316 p.
- [3] ITU-T H.264 *Advanced video coding for generic audiovisual services* Ženeva:International Telecommunication Union, 2014
- [4] ITU-R BT.500-9 *Methodology for the subjective assessment of the quality of television pictures* Ženeva:International Telecommunication Union, 2012
- [5] ITU-T H.265 *SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services – Coding of moving video* Ženeva:International Telecommunication Union, 2013
- [6] TAN, T. K., M. MRAK, V. BARONCINI a N. N. RAMZAN. *HEVC verification test report*. Valencia, 2014.
- [7] RICHARDSON, Iain. *Summary: Introduction to HEVC*.Aberdeen, 2013. Dostupné také z: <<http://vcodex.com/h265.html>>.
- [8] NGUYEN, T. a D. D. MARPE. *Performance Comparison of HM 6.0 with Existing Still Image Compression Schemes Using a Test Set of Popular Still Images*. Ženeva, 2012.
- [9] BANKOSKI, J., J. KOLESZAR, L. QUILLIO, J. SALONEN, P. WILKINS a Y. XU. *VP8 Data Format and Decoding Guide*. USA, 2011. RFC 6386. Dostupné také z: <<http://www.rfc-editor.org/info/rfc6386>>.
- [10] BANKOSK, Jim, Paul WILKINS a Yaowu XU. *TECHNICAL OVERVIEW OF VP8, AN OPEN SOURCE VIDEO CODEC FOR THE WEB*. USA, 2011. Dostupné také z: <<http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/37073.pdf>>.
- [11] GOSLING, James a Henry MCGILTON. Design Goals of the Java TM Programming Language. *The Java Language Environment* [online]. 1996 [cit. 2015-12-13]. Dostupné z: <<http://www.oracle.com/technetwork/java/intro-141325.html>>.

- [12] *1.3.1 What is MySQL?* [online]. Oracle Corporation, 2016 [cit. 2016-05-20]. Dostupné z URL: <<http://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>>.
- [13] *Chapter 28 MySQL Workbench* [online]. Oracle Corporation, 2016 [cit. 2016-05-20]. Dostupné z URL: <<http://dev.mysql.com/doc/refman/5.7/en/workbench.html>>.
- [14] ŘEŘÁBEK, Martin; EBRAHIMI, Touradj. *Comparison of compression efficiency between HEVC/H. 265 and VP9 based on subjective assessments*. International Society for Optics and Photonics, 2014. p. 92170U-92170U-13.
- [15] DUMIĆ, E.; GRGIĆ, D.; ŠAKIĆ, F. and K.. *Subjective quality assessment of H.265 versus H.264 Video Coding for High-Definition Video Systems* Telecommunications (ConTEL), 2015 13th International Conference on, Graz, 2015
- [16] ŘEŘÁBEK, Martin; EBRAHIMI, Touradj; HANHART, Philippe; KORSHUNOV, Pavel; *Quality Evaluation of HEVC and VP9 Video Compression in Real-Time Applications*, Quality of Multimedia Experience (QoMEX), 2015 Seventh International Workshop on, Pylos-Nestoras, 2015

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

| | |
|---------|---|
| DVB-T | pozemní digitální televizní vysílání |
| ITU-T | International Telecommunication Union |
| ISO/IEC | (International Organisation for Standardisation / International Electrotechnical Commission |
| DCT | Diskrétní Kosinová Transformace |
| HDTV | formát televizního vysílání s vysokým rozlišením |
| FXML | programovací jazyk pro tvorbu uživatelského prostředí aplikace |
| CRT | cathode ray tube – typ monitoru, dnes již zřídka používaný |
| LCD | displej z tekutých krystalů |
| SQL | Structured Query Language |
| API | Application Programming Interface |
| PSNR | Peak signal-to-noise ratio |

SEZNAM PŘÍLOH

| | |
|--|-----------|
| A Příloha | 50 |
| A.1 Datová komprese použitých kodeků | 50 |

A PŘÍLOHA

Přílohou elektronické verze této práce je složka Java projektu obsahující zdrojový kód aplikace TestApp, bez použitých videí. Program lze spustit ve složce dist. Dále obsahuje Excelový dokument se všemi shromážděnými daty.

A.1 Datová komprese použitých kodeků

Původní velikosti videosekvencí:

- crowd run - 1,44 GB
- old town cross - 1,44 GB
- ducks take off - 1,44 GB

| Kodek | Video | 640x360 | 854x480 | 1280x720 | 1920x1080 |
|--------------|----------------|----------------|----------------|-----------------|------------------|
| H.264 | crowd run | 1883 kB | 4978 kB | 9246 kB | 14760 kB |
| H.264 | ducks take off | 1538 kB | 3825 kB | 7678 kB | 12282 kB |
| H.264 | old town cross | 1891 kB | 4998 kB | 9229 kB | 14750 kB |
| H.265 | crowd run | 1852 kB | 4875 kB | 9050 kB | 14487 kB |
| H.265 | old town cross | 1797 kB | 4840 kB | 9086 kB | 14558 kB |
| H.265 | ducks take off | 1336 kB | 3590 kB | 7305 kB | 11967 kB |
| VP8 | crowd run | 1852 kB | 4912 kB | 9212 kB | 10326 kB |
| VP8 | ducks take off | 1474 kB | 3828 kB | 7644 kB | 12582 kB |
| VP8 | old town cross | 1868 kB | 4889 kB | 9160 kB | 5415 kB |
| VP9 | crowd run s | 1990 kB | 5028 kB | 9520 kB | 15686 kB |
| VP9 | ducks take off | 694 kB | 1727 kB | 4398 kB | 7691 kB |
| VP9 | old town cross | 1050 kB | 2398 kB | 6028 kB | 10061 kB |

Tab. A.1: Komprese dat.