



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VLIV ŠUMU NA KOMPRESI VIDEOA

EFFECT OF NOISE ON VIDEO COMPRESSION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MATYÁŠ ANTON

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. DAVID BAŘINA

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Anton Matyáš**

Obor: Informační technologie

Téma: **Vliv šumu na kompresi videa**

Effect of Noise on Video Compression

Kategorie: Zpracování obrazu

Pokyny:

1. Seznamte se se současnými metodami komprese videa.
2. Významné metody podrobně popište. Popište také různé druhy šumů a rušení vyskytujících se ve videu a metod, které tento šum a rušení redukuje.
3. Otestujte vliv různých typů šumu na kompresi videa (kompresní poměr, kvalita).
4. Před vlastní kompresí zařadte filtr redukující šum. Otestujte různé druhy filtrů.
5. Ze srovnání vyvodte závěry, navrhněte další pokračování práce.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

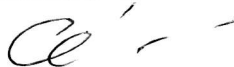
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Bařina David, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Bcžetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Videa jsou jakožto obrazový signál v různých fázích zpracování náchylná na vznik zkreslení ve formě šumu. Tato práce zkoumá, jakým způsobem může přítomnost šumu ovlivnit výsledky komprese obrazových dat a jaký dopad má naopak ztrátová komprese na přítomný šum. Pro tento účel byly zvoleny čtyři v současnosti používané formáty H.264, H.265, Motion JPEG 2000 a DIRAC zastupující rozdílné přístupy ke kompresi videa, především pak kontrast mezi diskretní kosinovou a diskretní vlnkovou transformací.

Abstract

Videos are in fact image signals and as such, they are susceptible to distortions in the form of noise. This thesis' goal was to determine how the presence of noise can affect the results of image data compression as well as what impact might the lossy compression have on the noise present. Four contemporary compression formats were chosen, namely H.264, H.265, Motion JPEG 2000 and DIRAC, representing different approaches to the video compression, especially the contrast between a discrete cosine transform and a discrete wavelet transform.

Klíčová slova

Komprese videa, Redukce šumu, Diskretní kosinová transformace, Diskretní vlnková transformace, H.264, H.265, Motion JPEG 2000, DIRAC

Keywords

Video compression, Noise reduction, Discrete cosine transform, Discrete wavelet transform, H.264, H.265, Motion JPEG 2000, DIRAC

Citace

ANTON, Matyáš. *Vliv šumu na kompresi videa*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Bařina David.

Vliv šumu na kompresi videa

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Davida Bařiny. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Matyáš Anton
16. května 2016

Poděkování

Chtěl bych poděkovat vedoucímu práce, Ing. Davidu Bařinovi, za trpělivost a za cenné rady, které mi při tvorbě této práce poskytl.

© Matyáš Anton, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	2
2	Kompresie videa a šumy	3
2.1	Video, barevné modely a podvzorkování	3
2.2	Kódování snímků	4
2.3	Diskrétní kosinová transformace (DCT)	5
2.4	Diskrétní vlnková transformace (DWT)	5
2.5	Využití prostorové a časové redundance a entropické kodéry	6
2.6	Formáty pro uchovávání videa	7
2.7	Druhy šumů	9
2.8	Metody redukce šumu	9
3	Zkoumání vlivu šumu na kompresi videa	11
3.1	FFmpeg	11
3.2	Generování šumu	11
3.3	Hodnocení kvality SSIM	12
3.4	Implementace programů pro kódování, filtrování a porovnání videí	12
3.5	Testování vlivu šumu na kompresi videa	13
3.6	Srovnání komprese bez přidaného šumu	13
3.7	Srovnání efektu šumu s nízkou granularitou na různé formáty	14
3.8	Srovnání efektu šumu s vysokou granularitou na různé formáty	16
3.9	Efekt šumu na jednotlivé formáty	17
3.10	Srovnání komprese po zapojení filtrů pro redukci šumu	20
4	Závěr	23
	Literatura	24
	Seznam příloh	25

Kapitola 1

Úvod

Video je v principu obrazový signál. Díky tomu může v různých fázích jeho zpracování od snímání přes zachytávání až po digitalizaci a další úpravy dojít ke zkreslení obsažené informace přimícháním šumu. Současně s sebou objemnost těchto obrazových dat nese potřebu komprimovat je pro snazší manipulaci.

Cílem této práce je zhodnotit, jak se přítomnost šumu a různé kompresní metody vzájemně ovlivňují a který formát je pro kompresi takovýchto dat vhodnější. Pro experimentování byly vybrány formáty H.264 a H.265, jejichž jádro staví na diskrétní kosinové transformaci, a Motion JPEG 2000 a DIRAC vycházející z diskrétní vlnkové transformace. Ve všech čtyřech případech se jedná o v současnosti používané formáty reprezentující rozdílné přístupy ke kompresi videa.

Se zmíněnými formáty byla provedena série experimentů, jejímž výsledkem je nejen stanovení nejefektivnějšího formátu pro uchovávání videa zkresleného šumem, ale také odhalení anomálie, kdy snižování datového toku a z něj plynoucí ztráta informace vedou k paradoxnímu zvyšování kvality obrazu.

Práce je rozčleněna do dvou hlavních celků. Kapitola 2 se věnuje vysvětlení teoretických principů, na nichž práce staví. To zahrnuje primárně základní termíny a postupy využívané při kompresi a dekompresi videa, ale také rozbor jednotlivých zkoumaných formátů. V závěru jsou nastíněny původ a povaha šumů a také způsoby, jakými se standardně redukují.

Druhá významná část, nacházející se v kapitole 3, se soustřeďuje na praktické aspekty práce. Je zde popsána implementace dvojice programů využitých k provádění experimentů a dále detaily experimentů samotných.

V poslední, 4. kapitole, se nachází shrnutí cílů, zjištění a výsledků práce.

Kapitola 2

Kompresce videa a šumy

Tato kapitola se věnuje vysvětlení teoretických principů, na nichž je práce vystavěna. V první části jsou popsány terminologie a principy související se zpracováním a uchováváním videa, jako jsou kódování snímků, diskrétní transformace a využití redundance. Informace v této části pochází především z knihy Handbook of Data Compression [11]. Následně jsou v sekci 2.6 s využitím zavedené terminologie rozebrány čtyři konkrétní formáty, kterými se tato práce dále zabývá. Poslední dvě sekce kapitoly, 2.7 a 2.8, nastiňují druhy šumů, se kterými je možné se při zpracování obrazu setkat, a metody, jak s těmito šumy bojovat.

2.1 Video, barevné modely a podvzorkování

Video je možné popsat jako funkci tří diskrétních veličin (x, y, t) , kde x a y reprezentují souřadnice a t pozici v čase. Jejím výstupem je pak informace o barevné hodnotě odpovídajícího bodu, pixelu. Z toho vyplývá potřeba tuto hodnotu nějak reprezentovat. Způsob, jakými jsou data uložena, se označuje jako formát.

Jeden z důležitých faktorů je způsob, jakým je zaznamenána barevná informace. Ve zpracování obrazu hojně využívanou variantou je model RGB. Ten nese informaci ve formě tří složek, červené (red), zelené (green) a modré (blue), přičemž pro každou z nich může být vyhrazeno až 8 bitů. Model funguje na aditivním principu, tedy všechny složky v nule odpovídají černé, všechny složky maximální znamenají bílou. [11]

V případě reprezentace videa je však častěji využíván model YCbCr. Barva je v tomto případě reprezentována intenzitou jasu (složka Y) a dvěma chromatickými složkami C_b, C_r . Pro převod z formátu RGB je možné pro konkrétní pixely vycházet ze vztahu 2.1.

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} +0,299 & +0,587 & +0,114 \\ -0,16875 & -0,33126 & +0,5 \\ +0,5 & -0,41869 & -0,08131 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.1)$$

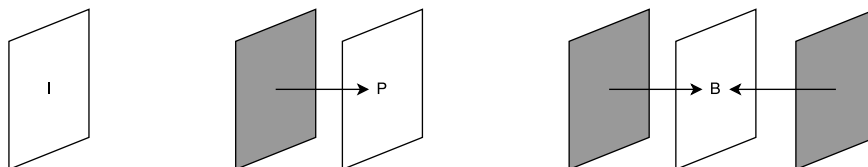
Za předpokladu 8 bitů na jednu barevnou složku by na pixel připadalo 24 bitů. Sama o sobě tato hodnota není příliš velká, ovšem po vynásobení rozsahy na osách x, y, t mohou být výsledné paměťové nároky značné. Z toho důvodu se přistupuje k rozličným kompresním technikám. Jednou z těch jednodušších je podvzorkování barvonosných složek. Staví na tom, že lidské oko je nejcitlivější na jasovou složku obrazu, a drobné odchylky v barvonosných obvykle nezachytí. K označení způsobu podvzorkování se používá trojice čísel $a:b:c$, kde a představuje šířku bloku (standardně se udává 4), b počet vzorků chromatických složek na prvním řádku a c jejich počet na druhém. Je tak možné setkat se například s variantami

YCbCr 4:4:4 (bez podvzorkování), 4:2:2 (na každém řádku dva vzorky, tzn. horizontální podvzorkování na polovinu) nebo 4:2:0 (horizontálně i vertikálně podvzorkováno na polovinu). Zvláště poslední zmíněná reprezentace je v oblasti kódování videa silně zastoupena. [11]

Dvojměrná matice, kterou spolu jednotlivé pixely v dimenzích x a y utvářejí, se nazývá snímek, a představuje základní jednotku videa. Jeho rozměry popisuje hodnota zvaná rozlišení, která svou velikostí popisuje úroveň detailů, jež může snímek nést.

2.2 Kódování snímků

Při kompresi videa je možné využívat několik druhů snímků. V základě je možné rozdělit je na snímky klíčové (intra frame) a rozdílové (inter frame). Klíčové snímky, v terminologii standardu MPEG označované jako I snímky, obsahují veškeré obrazové informace potřebné pro své dekódování. Rozdílové naproti tomu nesou jen rozdíl oproti svým sousedům – tzv. referenčním snímkům. V případě, že se rozdíly určují jen oproti jeho předchůdci, označuje se snímek jako P (predictive), pokud se však využívají i informace ze snímku, který po něm následuje, značí se B (bidirectional). [11]



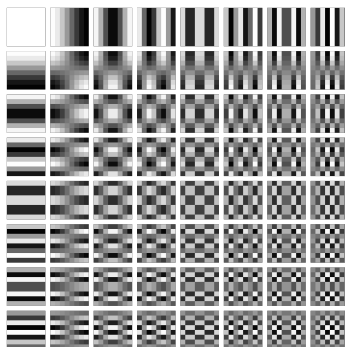
Obrázek 2.1: Znázornění závislosti jednotlivých typů snímků na snímcích sousedních.

Rozdílové snímky se díky menším hodnotám lépe komprimují a umožňují tak snížit velikost datového toku videa. Na druhou stranu přináší nutnost udržovat pro dekódování v paměti i sousední snímky, ze kterých vycházejí. V případě, že jsou využity jen snímky typu P, tedy takové, které se odkazují jen na předchůdce, je situace poměrně jednoduchá, jelikož stačí potřebné, již dekódované snímky uchovat pro další použití. Při zakomponování snímků typu B je ovšem situace jiná, jelikož vyžadují také informace ze snímků budoucích, které při sekvenčním postupu dekodér teoreticky ještě nemá. Řešením je změnit pořadí snímků tak, aby se vždy dekodovaly až po všech svých závislostech. Pro korektní rekonstrukci ovšem změnu pořadí musí být schopen detekovat i dekodér, proto bývá zavedena dvojice časových razítek (time stamp). První z nich, DTS (decoding time stamp), udává, v jakém pořadí mají být snímky dekódovány. Druhé, PTS (presentation time stamp) pak říká, kdy mají být zobrazeny. [11]

Použití rozdílových snímků však s sebou nese ještě jednu potenciální komplikaci. V případě, že je video tvořeno klíčovými snímky a dojde k poškození nebo ztrátě dat, projeví se změna jen lokálně na krátký okamžik – délku trvání snímku. Pokud by ale bylo video tvořeno jednolitou posloupností rozdílových snímků a někdy v průběhu došlo ke ztrátě, chyba by se při dekódování propagovala a v extrémním případě by mohlo dojít dokonce i k ovlivnění veškerých snímků. Mimo jiné z tohoto důvodu se obvykle rozdílové snímky v určitých intervalech prokládají snímky klíčovými a vznikají tak vzájemně nezávislé, I snímky ohraničené skupiny. Klíčový snímek ukončující jednu skupinu pak současně otevírá další. V terminologii standardu MPEG se tyto skupiny označují jako GOP (Group of Pictures). [11]

2.3 Diskrétní kosinová transformace (DCT)

Ve standardu MPEG je možné snímky jako takové dále členit na menší části, takzvané makrobloky. Ty jsou pro jasovou složku obrazu standardně tvořeny maticemi 16×16 pixelů, dále dělenými na bloky o rozměrech 8×8 . Velikost makrobloků pro chromatické složky se odvíjí od podvzorkování, např. pro 4:2:0 to bude 8×8 .



Obrázek 2.2: Bázové funkce 2D DCT pro blok 8×8 využívané ve formátu JPEG.¹

Při kompresi je poté na jednotlivé bloky 8×8 aplikována diskretní kosinová transformace (DCT). Výsledkem je 64 koeficientů uspořádaných v matici. Koeficient na souřadnicích $(0, 0)$ reprezentuje stejnosměrnou složku signálu a označuje se DC (inspirováno terminologií z elektrotechniky, kde DC znamená direct current, tedy stejnosměrný proud). Zbylých 63 koeficientů se značí AC (alternating current, střídavý proud) a udávají váhy, s jakými je přítomna odpovídající dvourozměrná kosinusoida. Tato část je plně reverzibilní. [11]

Koeficienty jsou dále kvantovány podle kvantizační tabulky. Ta udává, jakými hodnotami mají být jednotlivé koeficienty poděleny, než budou opět zaokrouhleny na nejbližší celé číslo. Tím dochází ke ztrátě přesnosti, a tato část tudíž není plně reverzibilní. Míra ztrátovosti roste s velikostí hodnot v tabulce. Jejich rozložení vychází z předpokladu, že nejvíce obrazových informací nesou koeficienty na nízkých frekvencích (poblíž souřadnic $0, 0$), a naopak hodnoty na vyšších frekvencích tolik důležité nejsou, tudíž je snaha je redukovat více, ideálně až na nulu. V tabulce se proto kolem nuly koncentrují hodnoty nižší, ve vyšších frekvencích hodnoty vyšší. Při dekódování se koeficienty obsahem tabulky násobí. [11]

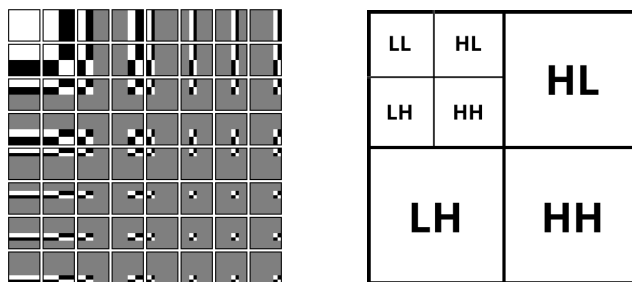
Kombinace kvantování a nezávislého zpracovávání bloků obrazu může při nízkém datovém toku (a tedy vysoké ztrátovosti) vést ke vzniku tzv. blokového efektu, neboli artefaktů způsobených nedokonalým vizuálním navazováním sousedních bloků znázorněných na obrázku 2.4. Většina kodeků se však snaží tomuto efektu nějakou formou předcházet. [11]

2.4 Diskrétní vlnková transformace (DWT)

Alternativou k diskretní kosinové transformaci je použití DWT neboli diskretní vlnkové transformace. Pro názornější vysvětlení jejího principu bude použit příklad z formátu JPEG 2000.

V prvním kroku může být snímek pro každou barevnou složku rozsekán na stejně velké, samostatně zpracovávané dlaždice. Na ty je dále aplikována diskretní vlnková transformace. Prováděna je pro každý rozměr zvlášť, nejprve po řádcích a následně po sloupcích. [1]

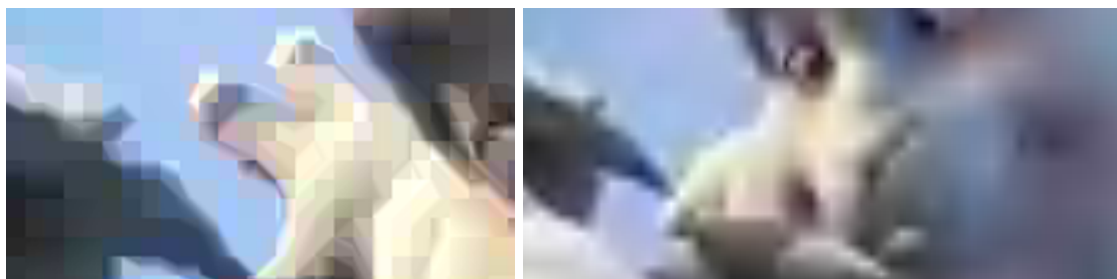
¹Zdroj: https://en.wikipedia.org/wiki/Discrete_cosine_transform/media/File:DCT-8x8.png



Obrázek 2.3: Nalevo ilustrace 2D bázových funkcí při použití Haarovy vlnky³, napravo znázornění podpásem při rozkladu úrovně 2.

Standard JPEG 2000 specifikuje pro transformaci dva druhy vlnek – $9/7$, využívající hodnoty s desetinnou čárkou (pro ztrátovou kompresi), a celočíselnou $5/3$ (možno využít pro ztrátovou i bezztrátovou kompresi). Výsledkem jsou čtyři frekvenční podpásma koeficientů (LL, LH, HL a HH). Pásmo LL, obsahující nejnižší frekvence tvořící primární část obrazové informace, je v obou směrech podvzorkováno v poměru 2:1 na nižší rozlišení a zbylá tři pásma slouží jako nositelé doplňkových informací pro rekonstrukci původního obrazu. Následně je možné nad LL provést další diskretní vlnkovou transformaci, opět jej rozdělit na sadu čtyř podpásem a tímto způsobem pokračovat teoreticky až do standardem povolené hloubky 32. Ve většině případů však úrovně rozkladu větší než pět nemají z hlediska komprese výraznější přínos. [9]

Vzniklé koeficienty jsou dále kvantovány. V rámci každého pásma se vydělí velikostí kvantizačního kroku (může být pro jednotlivá pásma individuální) a výsledky se zaokrouhlí. V případě bezztrátové komprese je kvantizační krok 1, tedy kvantování de facto neprobíhá.



Obrázek 2.4: Ukázka blokového efektu při nízkém datovém toku u formátu H.264 (nalevo) a zkreslení při nízkém datovém toku u formátu Motion JPEG 2000 (napravo).

2.5 Využití prostorové a časové redundance a entropické kódey

Při kódování intra snímků se dále pro snížení objemu využívá jejich prostorové redundance pomocí techniky predikce makrobloků. Variant existuje více, v principu se však makroblok vyplní podle již známého okolí (sloupec před, řádek nad nebo kombinace obou) a kóduje se jen chyba, tedy jak se tato predikce lišila oproti skutečnosti.

Kódování rozdílových snímků se poněkud liší. Využívá se zde technika kompenzace po-

³Zdroj: <http://www.blitzcode.net/compression.shtml>

hybu (motion compensation), vycházející z časové redundance snímků. V režimu rozdílových snímků je obsah odvozen z rozdílu vůči předchozímu, případně následujícímu snímku (chyby predikce). Změny mezi snímky jsou však ve videu často způsobeny pohybem objektů. Z toho plyne, že se shodné části obrazu mohou vyskytovat v různých snímcích na různých pozicích a v odlišných makroblocích. Proto jsou snímku přiřazeny jeden (v případě P snímků) nebo dva (v případě B snímků) pohybové vektory, které tento posun popisují. Hodnoty pohybových vektorů mohou být celočíselné (s přesností na pixel) nebo reálné (poloviny, případně čtvrtiny pixelu). Desetinná přesnost s sebou nese nutnost interpolace a z toho plynoucí vyšší výpočetní i paměťovou náročnost.

Algoritmy pro určování pohybových vektorů jsou netriviální a platí, že nemusí vždy určit související oblasti správně. Z toho plyne mimo jiné to, že se pro různé implementace mohou výsledné pohybové vektory značně lišit. V dnešní době se obvykle využívá jak prostorové (intra snímky, predikce makrobloků), tak časové redundance (inter snímky, kompenzace pohybu). Kodeky, které tyto dva přístupy kombinují, se označují jako hybridní.

Posledním krokem činnosti kodeku je získané koeficienty, případně jejich predikce efektivně zakódovat. K tomu se využívají entropické kodéry. Entropie je veličina udávající střední množství informace obsažené v jednom symbolu dat, respektive množství dat potřebné k vyjádření jednoho znaku, a právě z entropie dat tyto kodéry vychází. Na základě četnosti výskytu opakujících se dat přiřazují znakům nebo celým sekvencím zástupné kódy. Jednou z variant je použití kódů proměnné délky, používaných např. Huffmanovým kódováním. Myšlenka je taková, že častěji se vyskytující znaky dostanou přiřazeny krátké kódy, a s dalšími, méně častými znaky se délka přiřazených kódů zvyšuje. To může vést k situaci, kdy se znaku přiřadí kód delší než znak samotný, ve výsledku však díky rozdělení na základě četnosti stejně dochází k úspoře. Alternativou jsou např. aritmetické kodéry, vycházející z rozdělení pravděpodobnosti vstupních symbolů.

Konkrétních kódovacích algoritmů existuje celá řada a jejich volba se odvíjí primárně od specifikace formátu, který je využívá. Jednotným rysem však je, že je tato forma komprese bezztrátová.

2.6 Formáty pro uchovávání videa

Následující sekce se věnuje čtyřem v současnosti používaným formátům videa, jejichž zkoumáním se tato práce zabývá. Tyto formáty reprezentují různé přístupy k uchovávání obrazových dat, ať už po stránce použitých transformací, kódování, nebo strukturování snímků. Zmíněny jsou také významné implementace kodeků (software sloužící pro kódování, resp. dekodování dat určitého formátu) s nimi pracujících.

H.264 / MPEG-4 AVC je formát spadající do rozsáhlého multimediálního standardu MPEG-4, konkrétně jeho části 10 (Advanced Video Coding), vznikl jako plánovaná náhrada svých v té době již kompresně neefektivních předchůdců H.261 až H.263. Jeho formální specifikace ITU-T H.264, resp. ISO/IEC 14496-10, vyšla v roce 2003 a od té doby doznala řady rozšíření, naposledy v roce 2014. [7]

V současné době je tento formát značně rozšířen. Nachází uplatnění na Blu-ray discích, v digitálním televizním vysílání, ale také jako jeden z hlavních formátů pro přenos internetového videa. Díky jeho rozšíření existuje nespočet softwarových i hardwarových implementací, proprietárních i otevřených, často integrovaných již přímo do operačního systému. K významným otevřeným variantám patří například x264 (jen kodér) nebo libavcodec z balíku FFMPEG (pouze dekodér).

Podporuje YCbCr formáty 4:0:0, 4:2:0, 4:2:2 a 4:4:4 a rozlišení obrazu až do 4K (4096 × 2160 pixelů). Při kódování využívá I, P a B snímky, časových razítek a snímky jsou členěny na řezy složené z makrobloků. Kromě samotných snímků však mohou být typy I, P, B nově přiřazeny také konkrétním makroblokům. Makroblok se dělí na bloky, které mohou nabývat rozměrů 4 × 4 až 16 × 16. Pohybové vektory podporují přesnost na čtvrt vzorku a pro predikci je možné používat až 16 referenčních snímků. Pro B snímky je navíc využívána vážená predikce pracující s vahou, ke kterému referenčnímu snímku má kódovaný blok blíž. Pro získání koeficientů je využita diskretní kosinová transformace, před kvantizací jsou však ještě hodnoty v tabulce násobeny koeficientem q . Jakožto entropický kodér může být použit buď CAVLC, využívající kódy s proměnlivou délkou, nebo na aritmetickém kódování založený CABAC, který je efektivnější, avšak také náročnější na prostředky. [11]

H.265 / HEVC byl vytvořen organizacemi ITU-T a ISO/IEC jako nástupce dnes rozšířeného H.264. Hlavní motivací byly zvyšující se požadavky na rozlišení videa související s vývojem technologií. Přestože totiž H.264 teoreticky podporuje i videa v rozlišení 4K, potřebný datový tok byl shledán neúnosným, a tak vyvstala nutnost přijít s efektivnějším kompresním algoritmem. V současné době je stále spíše ve fázi vývoje a nasazení do praxe ho teprve čeká, avšak už nyní deklaruje až o 50 % lepší kompresní poměr než H.264 a jeho podpora byla předběžně potvrzena pro Blu-ray disky v rozlišení 4K. [8]

Původní specifikace ITU-T H.265, resp. ISO/IEC 23008-2, pochází z roku 2013 a od té doby vyšly dvě revize, poslední v roce 2015 [6]. Vzhledem ke stále probíhajícímu vývoji standardu se v různě raných fázích nacházejí také jeho implementace. K pokročilejším v současnosti patří například x265 nebo libde265.

V současné době podporuje YCbCr formáty 4:0:0, 4:2:0, 4:2:2 a 4:4:4 a rozlišení až do 8K (7680 × 4320 pixelů). Bloky a makrobloky jsou zde nahrazeny CTB (coding tree block) a CTU (coding tree unit), které na rozdíl od klasických 16 × 16 mohou nabývat také rozměrů 32 × 32 a dokonce až 64 × 64 pixelů, což umožňuje dosahovat efektivnějších kompresních výsledků. Pohybové vektory opět umožňují přesnost na čtvrtiny vzorku, maximální počet odlišných referenčních snímků je tentokrát 8. Stejně jako u H.264 i zde je pro B snímky využívána váhová predikce. Jádro komprese znovu stojí na diskretní kosinové transformaci a kvantování koeficientů. Jako entropický kodér je použita vylepšená verze CABAC, která dosahuje vyšší komprese za menších hardwarových nároků. CAVLC na rozdíl od H.264 podporován není. [12]

Motion JPEG 2000 je formát postavený na základu JPEG 2000, který je určen pro kódování statických obrazových dat. Definován je v různých částech standardu ITO/IEC 15444, především částech 1 (jádro algoritmu) a 3 (rozšíření pro videa). Má tři referenční implementace, JJ2000, JasPer a OpenJPEG, přičemž za aktivní je v současnosti považována jen poslední zmíněná [5]. Experimentální podpora je implementována například také v balíku FFMPEG.

Využívá dělení na dlaždice, zpracování diskretní vlnkovou transformací, kvantování, viz sekce 2.4. Pro kódování videa se zde, na rozdíl od běžných formátů, nepoužívá žádná forma časové predikce, všechny snímky jsou typu I, a tedy kódovány nezávisle.

Důvodem je, že si formát neklade za cíl konkurovat běžným formátům, jaké představuje např. rodina MPEG, nýbrž vytvořit škálovatelný formát odolný vůči chybám vhodný pro síťový přenos, navíc umožňující rychlejší náhodný přístup ke snímkům, a to i za cenu vyššího datového toku. Pro závěrečné kódování koeficientů formát využívá aritmetický kodér EBCOT. [4][11]

DIRAC je formát vyvinutý pod záštitou společnosti BBC, vytvořený jakožto konkurent v té době již rozšířeného standardu H.264. Jeho specifikace pochází z roku 2008, formálně

standardizovaná je však jen jeho omezená podmnožina pojmenovaná VC-2 (nebo také Dirac Pro), konkrétně v SMPTE 2042-1-2009, podporující pouze I snímky. Formát má dvě referenční implementace, dirac-research (dříve jen Dirac) a Schrödinger.

Jedná se o hybridní kodér, na rozdíl od ostatních však místo diskrétní kosinové transformace využívá transformaci vlnkovou. K vlnkám 5/3 a 9/7 používaným v JPEG 2000 přidává ještě nové, například 13/7. Jinak ovšem funguje na stejných principech jako ostatní hybridní kodéry – používá I, P, B snímky a kompenzaci pohybu s využitím pohybových vektorů. K podporovaným podvzorkováním patří 4:4:4, 4:2:2 a 4:2:0. Pro kompresi koeficientů z DWT je využíván aritmetický kodér. [15]

2.7 Druhy šumů

Jedním z problémů při zachytávání, přenosu a zpracovávání obrazových dat je výskyt šumu. Šumem se v této oblasti rozumí nežádoucí, z povahy obvykle náhodné zkreslení jasové nebo barevné informace oproti originálnímu vzoru, často způsobené elektrickým rušením [3]. Následující řádky se věnují některým běžným typům těchto šumů a možnostem, jak je redukovat.

Gaussovský šum v obraze vzniká obvykle již během jeho zachytávání, ať už přímo na senzoru (například vlivem nedostatečného osvětlení nebo vysoké teploty), nebo při následném přenosu informace důsledkem šumu v elektrických obvodech. Mnohdy je navíc podpořen vestavěným zesilovačem v použitém zařízení. Jeho označení vyplývá ze stochastické povahy, nabývá totiž hodnot odpovídajících normálnímu rozložení. Jestliže má toto rozložení střed v nule, jedná se o tzv. bílý šum. Projevuje se aditivně (šum se k původním hodnotám přičítá), standardně ovlivňuje veškeré pixely obrazu, na jejich hodnoty však nejčastěji působí nezávisle. Díky tomu, že normální rozložení dobře kopíruje přirozené náhodné jevy, používá se Gaussovský šum také při aproximaci jiných, složitějších typů šumu, například filmového zrna. [3][2]

Impulzní šum na rozdíl od Gaussovského nepostihuje změnou veškeré pixely obrazu, ale jen jejich náhodnou podmnožinu. Zvláštním případem je tzv. šum "sůl a pepř", jehož případný vliv je vysoký a zjednodušeně se tak dají výsledné pixely podle efektu rozdělit na tři skupiny – nezměněné, černé (nulové) a bílé (maximální). K podobným jevům dochází nejčastěji důsledkem chyb na digitální rovině, například při převodu z analogové hodnoty, poškozením přenášených bitů apod. Pro redukci tohoto typu šumů je možné využít mediánový filtr. [3]

Filmové zrno je šum daný podstatou zachytávání obrazu na fotografický film. Film je tvořen velkým počtem zrn, která po dopadu světla mohou, ale také nemusí pohltit část fotonů, čímž se změní jejich vzhled. Počet takto změněných zrn je dán binomickým rozložením, obvykle je však možné modelovat tento šum zjednodušeně Gaussovským rozložením. [2]

2.8 Metody redukce šumu

Jelikož šum v obraze obvykle představuje vyšší frekvence než původní obrazová informace, zvláště pak pokud je jeho intenzita prostorově nezávislá, je jednou z možností, jak jej redukovat, provedení konvoluce obrazu s maskou reprezentující nízkopropustný filtr. Výsledkem je vzájemné přiblížení hodnot sousedních pixelů, což však v důsledku vede k rozostření obrazu. [3]

Při filtrování snímku konvolučním filtrem se konkrétní výsledky odvíjí především od rozměrů a koeficientů konvoluční masky, udávající kolik pixelů a jakou váhou ovlivní výslednou hodnotu. Nejjednodušším typem je maska, kde jsou všechny koeficienty stejné a tedy mají všechny zpracovávané pixely v okolí stejnou váhu a výsledkem je tak čisté zprůměrování jejich hodnot. Z toho kromě rozmazání detailů pramení také šíření lokálních extrémů dále po obraze. Druhému jmenovanému se dá do jisté míry předcházet použitím váženého průměru, jaký používá např. **Gaussův filtr**. V principu jde opět o konvoluci, hodnoty koeficientů v masce však vychází z Gaussovy funkce a nejvyšší váhy se tak soustřeďují v jejím středu a vliv výkyvů v okolí se snižuje. Přestože je technika poměrně efektivním nástrojem pro odstraňování Gaussovského šumu, stále zde dochází k rozmazání hran, a tudíž nežádoucímu zkreslení obrazu. [10]

Dosud uvedené filtry jsou tzv. lineární, jejich algoritmy vycházejí čistě z prostorové závislosti mezi pixely, což vede k nežádoucím účinkům v podobě ztráty informací o hranách nebo šíření lokálních extrémů. Alternativní řešení poskytují nelineární filtry.

Typickým příkladem tohoto druhu filtrů je mediánový filtr. Pro každý pixel vychází z hodnot okolí, které seřadí podle intenzit a vybere z nich hodnotu odpovídající mediánu, kterou poté pixelu přiřadí. Tento postup je velice efektivní zvláště proti šumům typu sůl a pepř popsaným výše. [3]

Dalším, pokročilejším nástrojem pro redukci šumu je **bilaterální filtr**. Ten, jak dává název tušit, kombinuje přístupy lineárních i nelineárních filtrů. Základem je jako u lineárních postupů konvoluce s maskou vycházející nejčastěji z Gaussovy funkce. Zde však před jejím provedením přichází druhý krok – pro každý pixel vyhodnocení barevné podobnosti s pixely sousedními a následná individuální modifikace původní masky. Pixely s podobnou barvou si svoje váhy zachovávají takřka nezměněné, naopak váha odlišných pixelů je snižována. To má za následek vyhlazení relativně souvislých ploch (a tedy odstranění šumu na nich) a současně zachování vysokých frekvencí tam, kde jsou třeba, tedy na hranách objektů. [13]

Kapitola 3

Zkoumání vlivu šumu na kompresi videa

Tato sekce se věnuje praktickým aspektům práce. V první části je popsána implementace dvou pomocných programů pro vkládání šumu do videa, jeho kompresi a následné porovnávání vůči originálu. Kromě toho jsou zde popsány také knihovny a algoritmy třetích stran pro jejich implementaci využité. Druhá část, začínající sekcí 3.5, pak rozebírá provádění jednotlivých experimentů, jejich výchozí předpoklady a zjištěné výsledky.

3.1 FFmpeg

Pro práci s multimediálními soubory byl využit balík FFmpeg ve verzi 2.8.2. Jedná se o multimediální framework, tedy soubor nástrojů a knihoven pro přehrávání, kódování, dekódování a další práci s multimediálními daty. Kromě několika konzolových aplikací poskytuje také knihovny s rozhraním pro jazyk C, z nichž stojí za zmínku především libavcodec zaštiťující kódování a dekódování.

Pro kompresi do jednotlivých formátů byly využity kodeky x264 (pro MPEG-4 AVC), x265 (HEVC), OpenJPEG (Motion JPEG 2000) a Schroedinger (DIRAC). Ve všech případech jde o udržované, otevřené implementace formátů, které je možné k frameworku volitelně doinstalovat.

3.2 Generování šumu

Klíčovým prvkem pro provedení experimentu je schopnost programu generovat do obrazu šum. Ten se přidává aditivně do jasové složky. Intenzita pro konkrétní pixel leží v intervalu $\langle -A, A \rangle$, kde A je uživatelem zvolená amplituda, přičemž jasová složka pixelu může nabývat hodnot od 0 (černá) do 255 (bílá). Konkrétní intenzita je dána náhodně dle normálního rozložení se středem v nule. Aby se docílilo efektu granularity, jsou při aplikaci šumu na pixel menší měrou ovlivněny také okolní pixely do vzdálenosti 4σ . Míra, s jakou jsou ovlivněny, je dána vzorcem pro dvourozměrnou Gaussovu funkci 3.1, kde σ je odchylka zadaná jako parametr uživatelem, x a y souřadnice počítané od aktuálně zpracovávaného pixelu (ten představuje bod $0, 0$) a a intenzita šumu určená výše. Z toho plyne, že se vliv na okolní pixely se vzdáleností od středu snižuje.

$$f(x, y) = a \exp\left(-\left(\frac{x^2}{2\sigma^2} + \frac{y^2}{2\sigma^2}\right)\right) \quad (3.1)$$

Z důvodu vysoké výpočetní složitosti tohoto algoritmu se exponenciální část vzorce pro jednotlivé souřadnice, jejíž výsledky jsou neměnné, předpočítá na začátku běhu programu do tabulky a dále se jen čtou hodnoty na základě aktuálních indexů.

3.3 Hodnocení kvality SSIM

Jakožto metoda pro hodnocení kvality výsledných videí byl zvolen SSIM (Structural SIMilarity) index. Tento algoritmus navržený Zhou Wangem a Alanem Bovikem se vyznačuje faktem, že nehodnotí kvalitu obrazu na základě absolutních numerických rozdílů tak jako jeho alternativy MSE nebo PSNR, ale zaměřuje se na to, jak je obraz vnímán člověkem, tedy na jeho věrnost z hlediska struktury. Typicky se zaměřuje na jasovou složku, na kterou je lidské oko nejcitlivější, a předpokládá potom vstup ve stupních šedi. [14]

V projektu byla využita volně šiřitelná implementace v jazyce C++ využívající grafickou knihovnu OpenCV, jejímž autorem je Rabah Mehdi¹. Výsledkem jsou reálné hodnoty v intervalu od 0 do 1, kde 1 značí naprostou shodu se vzorem a 0 absolutní rozdílnost.

3.4 Implementace programů pro kódování, filtrování a porovnání videí

Pro zašumění, filtraci, kompresi a srovnávání videí byly vytvořeny dva programy postavené na frameworku FFmpeg, viz 3.1. Jedná se o konzolové aplikace ovládané skrze vstupní parametry, vytvořené prostřednictvím jazyka C++.

V prvním kroku je spuštěn program `recode`. Ten jako vstupní parametry přijímá název vstupního a výstupního souboru, typ operace nad snímky s až dvěma parametry (amplitudu a směrodatnou odchylku pro generování šumu, velikost jádra a případně barevnou mez pro redukční filtry), název požadovaného výstupního kodeku, hodnotu určující kvalitu výstupu specifickou pro konkrétní kodek, jejíž podrobnosti je možné najít v tabulce 3.1.

Kodek	Parametr	Popis
x264, x265	CRF	hodnota v intervalu 0 až 51 (0 nejlepší, 51 nejhorší)
OpenJPEG	compression_level	hodnota v intervalu 0 až 100 (0 nejlepší, 100 nejhorší)
Schroedinger	bitrate	hodnota v KB/s, obvykle řády stovek až desetitisíců

Tabulka 3.1: Parametry určující kvalitu výstupu kodeků.

Vstupní soubor je otevřen, nalezne se videostopa a na základě ní je vyhledán odpovídající dekodér, to vše skrze knihovní funkce. Následně je podle zvolených parametrů nastaven výstupní kodér. Poté může začít dekodovací smyčka. V ní je opakovaně volána funkce, která má na výstupu dekodované pakety. V případě, že paket nese snímek požadovaného video proudu, předá se ke zpracování, v opačném případě je obsah paketu dealokován a načte se další. Snímek je nejprve převeden do barevného formátu YCbCr 4:2:0, společného pro všechny čtyři výstupní kodeky, a následně filtrován podle zvolených parametrů. V případě přidávání šumu je předán funkci `addGaussNoise`, která aplikuje postup popsany

¹<http://mehdi.rabah.free.fr/SSIM/>

v sekci 3.2. Pokud má být šum naopak redukován, využijí se funkce `GaussianBlur`, případně `bilateralFilter` poskytované knihovnou `OpenCV`. Takto modifikovaný je snímek předán kodéru a posléze uložen do výstupního kontejneru. Kodér však nezaručuje, že bude každý předaný snímek ihned zpracován. Důvodem je interní odkládání do vyrovnávací paměti pro účely tvorby rozdílových snímků. Z tohoto důvodu je třeba po skončení dekodovací smyčky spustit smyčku další, která opakovaně volá kódovací funkci s nulovým ukazatelem, dokud od ní nedostane informaci o dokončení kódování. Poté je výstupní kontejner uzavřen a veškeré zdroje dealokovány, čímž činnost `recode` končí. Výsledkem je modifikovaný, komprimovaný video soubor.

Nyní přichází na řadu druhý program, `compare`. Jeho parametrů je výrazně méně, postačuje zadat jen vzorový a nový soubor pro porovnání. Po spuštění programu je stejně jako v předchozím případě provedeno načtení informací ze souboru a určení potřebného dekodéru, tentokrát pro oba soubory. Následuje opět dekodovací smyčka, ovšem uvnitř ní je v tomto případě vložena druhá, aby byly snímky dostupné současně. Před porovnáním jsou snímky převedeny do formátu `BGR24` a jejich obsah vložen do struktur `IplImage` využívaných implementací algoritmu `SSIM`. Takto jsou předány porovnávací funkci. Výsledné informace jsou vypisovány na standardní výstup ve formátu číslo snímku, velikost, typ, `SSIM` index, přičemž jednotlivé položky jsou odděleny tabulátory a mezi záznamy o snímcích se nachází znak nového řádku.

3.5 Testování vlivu šumu na kompresi videa

V následující části práce jsou popsány experimenty s využitím dvou výše zmíněných programů, jejichž účelem bylo zjistit, jaký je vzájemný vztah mezi přítomností šumu v obraze a výsledky jeho komprese při použití čtyř rozdílných kompresních formátů. Testy byly prováděny na 10 vzorcích, reprezentujících různé druhy obrazových dat, a výsledné hodnoty následně zprůměrovány. Zastoupen byl živý obraz, čerpaný z otevřených filmů `Vaalkama`² a `Tears of Steel`³, dále počítačová 3D animace pocházející z otevřených filmů `Big Buck Bunny`⁴ a `Sintel`⁵, a nakonec dvojrozměrná animace z bezplatného filmu `Sita Sings the Blues`⁶. Vzorky byly v rozlišení 848×480 pixelů a každý o délce zhruba 1,5 minuty.

3.6 Srovnání komprese bez přidaného šumu

V první fázi byly vzorky zkomprimovány s různými úrovněmi kvality (od nejvyšší, takřka bezztrátové, až po nejnižší dovolenou faktorem konstantní kvality) bez přítomnosti šumu, aby byla k dispozici reference pro srovnání s výsledky komprese po přidání šumu. Předpokladem bylo, že nejefektivnějších kompresních poměrů by měl jakožto formát nové generace dosahovat `H.265`. Naopak nejhorší výsledky se daly očekávat od `Motion JPEG2000`, jehož návrh cílí spíše na spolehlivost, a z tohoto důvodu využívá jen `I` snímky. Jelikož `DIRAC` vznikl v době rozkvětu `H.264` a s cílem stát se jeho konkurentem, měly by být výsledky pro tyto dva formáty relativně podobné.

Výsledky referenčního testu jsou vyneseny na grafu 3.1. Na horizontální ose je hodnota `BPP`, neboli bitů na pixel, znázorňující použitý datový tok. Vertikální osa pak repre-

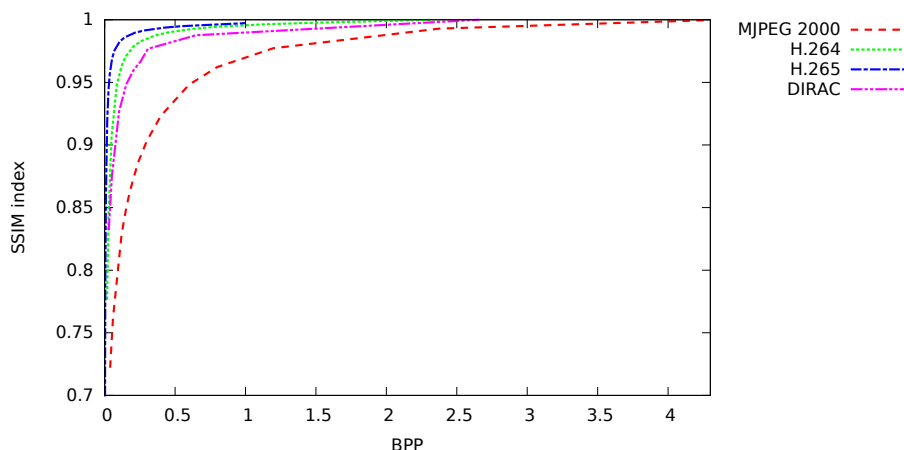
²<http://www.valkaama.com/>

³<https://mango.blender.org/>

⁴<https://peach.blender.org/>

⁵<https://durian.blender.org/>

⁶<http://sitasingingstheblues.com/>



Obrázek 3.1: Srovnání závislosti kvality na počtu bitů pro čtveřici formátů H.264, H.265, Motion JPEG 2000 a DIRAC.

zntuje hodnotu SSIM indexu (blíže vysvětlenou v sekci 3.3), kde 1 zastupuje nezkraslený originál.

Experiment dle očekávání ukazuje, že nejefektivnější komprese dosahuje kodek H.265. Na začátku sice kvalita klesla hlouběji než u ostatních (až na 0,72), avšak při hodnotě 0,002 BPP, což je tok, kterého jeho předchůdce H.264 ani vlnkové alternativy nebyly schopny dosáhnout. Maximální kvality pak dosahuje již na 1,3 BPP, tedy výrazně dříve nežli ostatní formáty. Jak je ovšem vidět na grafu, tato maximální kvalita nedosahuje SSIM hodnoty 1. To je způsobeno funkcí deblockingu zabudovanou v kodeku, která pomáhá redukovat blokové kompresní artefakty a dosahovat tak lepších kompresních poměrů. Negativem je, že i při maximální kvalitě může způsobovat tyto nepatrné odchylky od zdroje. I pro Motion JPEG 2000 se očekávání splnila a plné kvality dosáhl až při 4 BPP. Odchylkou oproti očekávání byly formáty H.264 a DIRAC, u kterých se, přestože by měly být efektivitou srovnatelné, ukázalo, že DIRAC má kompresní poměr horší. Ve všech případech kvalita nejprudčeji klesala při nízkých datových tocích, kde dochází ke vzniku artefaktů, ve vyšších tocích dochází ke změnám spíše pozvolně s tím, jak ubývají v obraze detaily.

3.7 Srovnání efektu šumu s nízkou granularitou na různé formáty

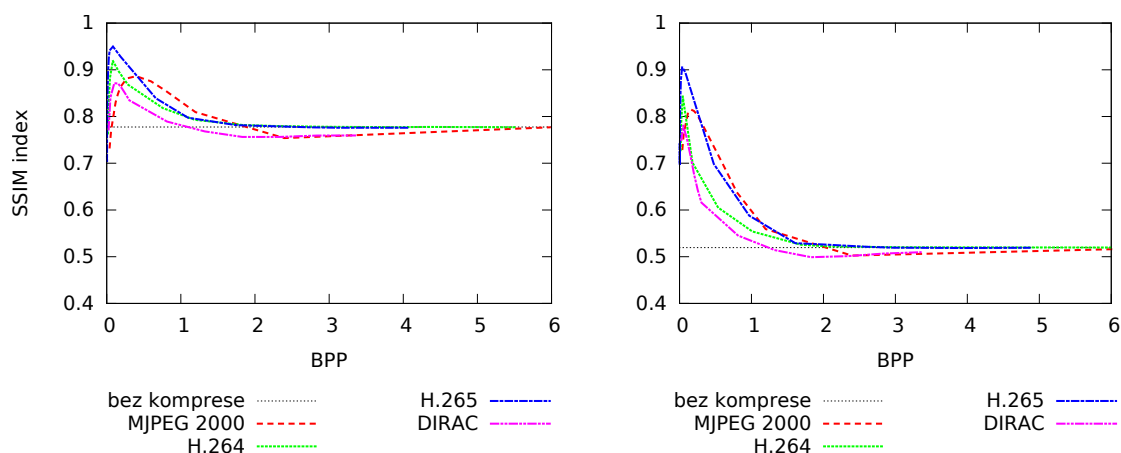
Druhý experiment měl za cíl zjistit, jak se výsledky formátů pro různé datové toky změní po přimíchání šumu o nízké granularitě, zvláště pak s ohledem na formáty používanou transformaci. Granularitou se rozumí velikost okolí, které je aplikací šumu na konkrétní pixel ovlivněno. V tomto případě byla zvolena vzdálenost 1, což znamená, že efekt byl jen lokální, a testovaly se dvě amplitudy šumu, 16 a 32.

Velké množství lokálních výchylek v obraze by mělo vést ke zvýšení entropie a z něj plynoucímu růstu nároků na datový tok potřebný k jeho reprezentaci, a to tím většimu, čím větší bude amplituda šumu. Jelikož jsou změny v obraze spíše bodového, prostorově nezávislého charakteru, představují vysoké frekvence, tedy ty, které se při kompresi nejvíce zkraslují. Snižování datového toku by tedy mělo vést k ubývání šumu a paradoxně i mírnému zlepšení kvality. Při věrné reprezentaci šumu lze očekávat, že opět nejlepšího poměru



Obrázek 3.2: Obraz s různými amplitudami šumu (zleva 0, 16 a 32) o dvou různých granularitách (horní řádek 1, dolní 5).

dosáhne H.265 a nejvyšší nároky na datový tok opět vyvstanou u Motion JPEG 2000, který na rozdíl od ostatních kóduje všechny snímky jako klíčové.



Obrázek 3.3: Srovnání závislosti kvality na počtu bitů na pixel při amplitudě šumu 16 (nalevo) a 32 (napravo) pro čtveřici formátů H.264, H.265, Motion JPEG 2000 a DIRAC.

Na levém grafu v 3.3 je možné vidět srovnání formátů při aplikaci slabšího šumu o amplitudě 16. Na vertikální ose je hodnota SSIM hodnotící věrnost komprimovaného videa se šumem vůči původnímu, nezašuměnému obrazu. Nekomprimovanému videu obsahujícímu šum pak v tomto případě odpovídá SSIM 0,78.

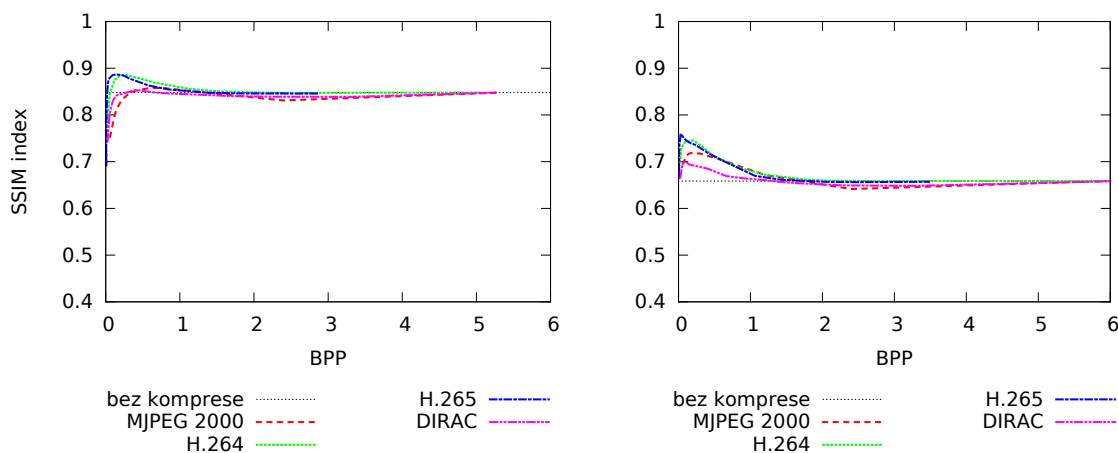
Křivky ukazují, že entropie se přidáním šumu skutečně zvýšila – u všech čtyř formátů datový tok výrazně stoupl, u H.264 a H.265 založených na diskretní kosinové transformaci dokonce více než dvojnásobně, pořadí efektivit však dle očekávání zůstalo oproti předchozímu experimentu nezměněno. S ubývajícím datovým tokem začíná docházet k ořezávání vysokých frekvencí. U vlnkových formátů sice dojde na čas k mírnému poklesu kvality, poté však stejně jako u těch na bázi DCT začne SSIM index díky redukování šumu prudce stoupat. Dochází tak k anomálii, kdy snižování datového toku vede ke zlepšování kvality. Tento jev kumuluje kolem 0,09 BPP (pro Motion JPEG 2000 vlivem pomalejšího průběhu poz-

ději), poté již následuje prudký propad vlivem kompresních artefaktů. K nejzásadnějšímu zlepšení došlo v případě H.265, který z původních 0,78 dosáhl až na kvalitu 0,95. Druhé nejefektivnější redukce dosáhl H.264 s kvalitou 0,92. Efekt vlnkových formátů byl horší, 0,87 v případě formátu DIRAC a 0,89 u Motion JPEG 2000. Za povšimnutí stojí, že vlivem pomalejšího průběhu redukce způsobené absencí neklíčových snímků formát Motion JPEG 2000 na úseku 0,5 až 1,5 BPP své konkurenty kvalitou převyšuje.

Graf v 3.3 napravo vyobrazuje situaci po přidání šumu o vyšší amplitudě 32 odpovídajícímu výsledné kvalitě SSIM 0,52. Entropie a z ní plynoucí nároky na datový tok dle očekávání opět narostly, byť ne již tak zásadně jako oproti videu bez šumu. Tvar průběhů a vzájemné pořadí dle efektivity zůstaly stejné jako v případě slabšího šumu, redukce na nízkých frekvencích je však znatelnější. H.265 se z SSIM 0,52 dostal až na 0,91 a skončil tak s výrazným předstihem. Pro H.264 a vlnkové formáty už jsou výsledky vzájemně blíže, u prvního jmenovaného 0,84, pro DIRAC 0,78 a 0,81 u Motion JPEG 2000.

3.8 Srovnání efektu šumu s vysokou granularitou na různé formáty

Při zvýšení granularity dochází k tomu, že je každý pixel obrazu ovlivněn nejen lokálně, ale také šumem z okolních pixelů, v případě třetího experimentu až do vzdálenosti 5 (jako amplitudy šumu byly opět zvoleny hodnoty 16 a 32), a dochází tak ke shlukování efektu do větších útvarů. Entropie by s ohledem na přidávání informací znovu měla narůstat, ovšem vzhledem ke zvýšené granularitě a s ní související podobnosti sousedních pixelů již ne v takové míře jako u granularity nízké. Stejně tak frekvence obsazené šumem se zřejmě s ohledem na potenciálně protichůdné působení sousedních změn sníží. To by v důsledku mohlo znamenat, že snižování datového toku nepovede v tomto případě pouze k redukci šumu, ale také zkreslování zbytku obrazu.



Obrázek 3.4: Srovnání závislosti kvality na počtu bitů na pixel při amplitudě šumu 16 (nalevo) a 32 (napravo) pro čtveřici formátů H.264, H.265, Motion JPEG 2000 a DIRAC.

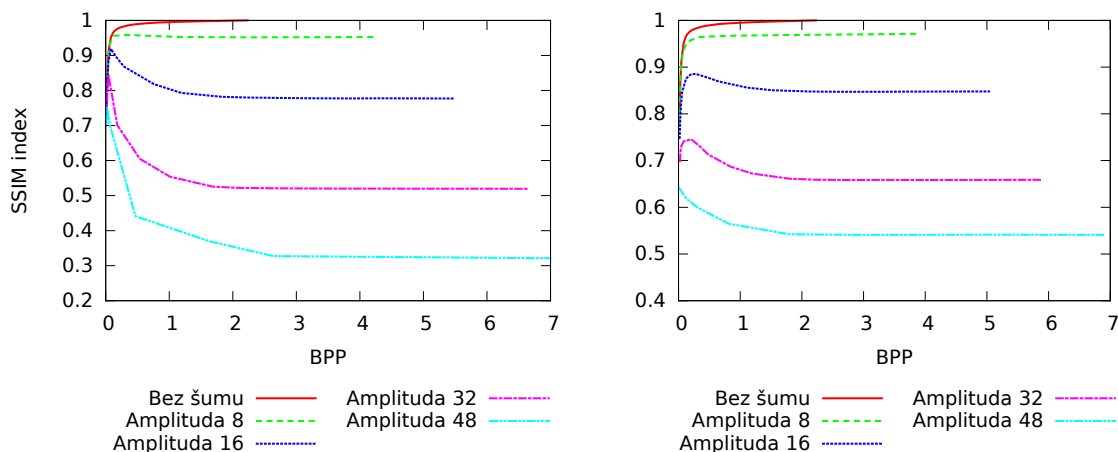
Výsledky experimentu je možné vidět na grafech 3.4. Ukázalo se, že šum s vyšší granularitou vede k menšímu zkreslení obrazu nežli šum s granularitou nízkou také z pohledu lidského oka. Tomu odpovídá kvalita 0,85 (při nízké granularitě byla 0,78). Entropie a s ní související nároky na datový tok oproti variantám s nízkou granularitou skutečně mírně

poklesla. Současně již však kompresí nedochází k natolik znatelnému zlepšení jako v případech popsaných v předchozí sekci. U kodeků na bázi DCT se stále nárůst kvality, byť v menší míře, vyskytuje, u DWT formátů ovšem na nízkých tocích kvalita rovnou klesá bez znatelného zlepšení. Ani umístění formátů není v tomto případě zcela jednoznačné. Křivky poměrů kvality jsou po většinu času velice blízko a v některých bodech se dokonce kříží. Celkově však po stránce kvality lépe vycházejí formáty na bázi diskretní kosinové transformace.

Pro vyšší amplitudu 32 (graf 3.4 napravo), reflektovanou kvalitou 0,66, již situace vypadá o něco přehledněji. Zvýšení amplitudy stačilo na to, aby i přes granulární efekt vedlo v obraze ke vzniku vysokých frekvencí, které mohou kodeky odstranit bez ztráty základní obrazové informace. Díky tomu se průběhy tvarem začínají podobat situaci s nízkou granularitou, přestože nárůst kvality s klesajícím datovým tokem stále není tak veliký. Také pořadí efektivit formátů začíná být jednoznačnější, avšak stále dochází k jistému prolínání.

3.9 Efekt šumu na jednotlivé formáty

Následující sekce se věnuje rozboru a srovnání dopadů výše probíraných jevů z hlediska individuálních kompresních formátů. Opět byly zvoleny šумы dvou granularit, nízké (vliv na sousední pixely do vzdálenosti 1) a vysoké (vliv do vzdálenosti 5). Kompresní vlastnosti jsou ilustrovány na amplitudách 8, 16, 32 a 48, odpovídajících kvalitám 0,95, 0,78, 0,52 a 0,32 pro nižší granularitu a 0,97, 0,85, 0,66 a 0,51 pro granularitu vyšší.



Obrázek 3.5: Závislost kvality na počtu bitů na pixel při použití formátu H.264 s různými intenzitami a granularitami šumu.

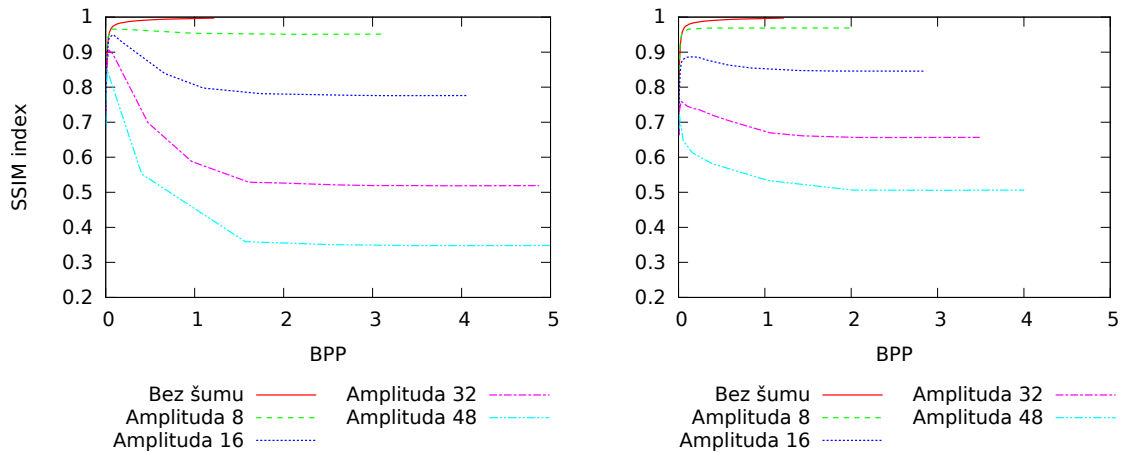
Na grafech 3.5 je vyobrazena závislost kvality na BPP při použití formátu H.264. Jak je vidět, s přibývajícím amplitudou šumu datový tok nutný pro věrnou reprezentaci kódovaného obrazu (odpovídající pravým koncům křivek) výrazně roste. Zatímco pro video bez šumu stačilo kodeku 2,23 BPP, při amplitudě 8 s nízkou granularitou již bylo třeba 4,27 BPP, tedy téměř dvojnásobek. Zvyšováním amplitudy datový tok dále rostl přes 5,51 BPP pro amplitudu 16 a 6,64 BPP pro 32 až na 7,4 BPP při amplitudě 48. Z hodnot vyplývá, že s přibývajícím šumem tempo nárůstu toku pozvolna klesá.

Také pro vysokou granularitu se nároky na datový tok zvyšují, byť o něco méně než při nízké. Objem pro amplitudu 8 tentokrát klesl na 3,92 BPP. S přibýváním intenzity šumu

se pak postupně vyšplhal na 5,03, 5,91 a 6,89 BPP. I zde je tedy možné sledovat trend zpomalování.

Druhým jevem, který se zde promítá, je redukce šumu při nízkých datových tocích. Pro nízkou granularitu a amplitudu šumu 8 sice redukce probíhá, je však natolik mírná, že má za následek pouze dočasné zastavení poklesu kvality viditelného u varianty bez šumu. To trvá až do hodnoty 0,32 BPP, kdy se kvalita začne prudce propadat. Po zdvojnásobení amplitudy na hodnotu 16 už dochází k jasně viditelné anomálii, kdy se kvalita s klesajícím tokem zvyšuje a kulminuje na hodnotě 0,92 při 0,086 BPP (značící zlepšení indexu kvality o 0,14), po níž opět dochází k propadu. Další zvýšení amplitudy přineslo také větší strmost růstu kvality. Při amplitudě 32 bylo maximum 0,84 při 0,046 BPP (rozdíl indexu 0,32) a po dalším zesílení šumu 0,75 při 0,023 BPP (rozdíl indexu 0,43). Se zvyšující se intenzitou šumu tedy roste také míra zlepšení kvality, které může ztratová komprese dosáhnout, a současně se maximum posouvá směrem k nižším datovým tokům. Za žádných okolností ovšem nedojde k situaci, kdy by formát při shodném datovém toku dosáhl vyšší kvality nežli pro šum s nižší amplitudou.

Vysoká granularita se projevila především na tempu průběhu vývoje kvality. Oproti šumu s nízkou granularitou zde v nízkých tocích dochází ke stoupání pozvolněji a také příkrý propad po dosažení maxima se zpomalil. Dalším efektem je celkové snížení kompresí dosahovaných zlepšení. V případě amplitudy 8 opět dochází spíše k brzdění propadu, zde již navíc ne zcela dokonalému. Propad kvality nastává na 0,32 BPP. V případě amplitud 16 a 32 se kvalita zlepšuje až na indexy 0,89 (rozdíl indexu 0,04) a 0,75 (rozdíl indexu 0,09), vrchol ovšem není ostrý. V případě amplitudy 48 je poloha maxima na tak nízkých tocích, že již není prostor pro růst ani propad, a tak se kvalita zastavuje na 0,66 (rozdíl indexu 0,15). To je dáno tím, že se poloha maxima stejně jako u nízké granularity posouvá s růstem amplitudy směrem k nižším tokům.



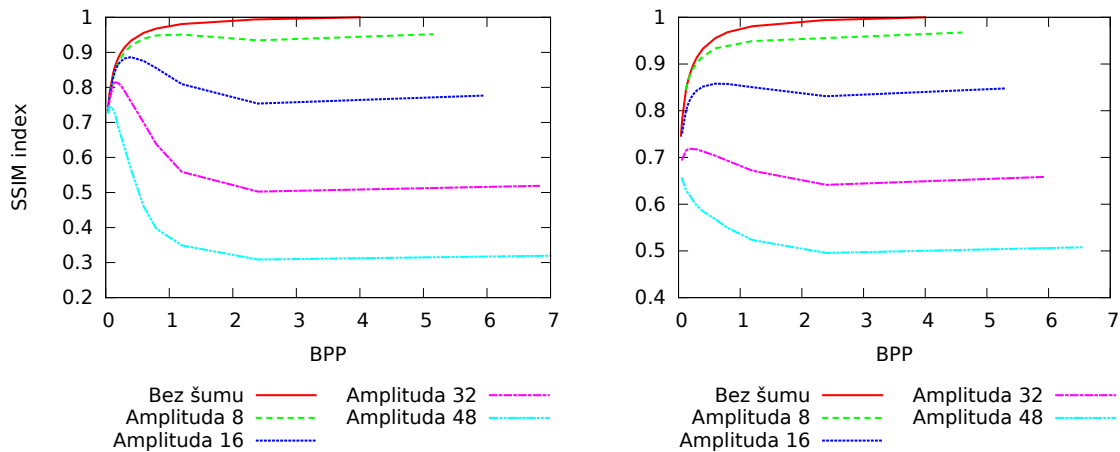
Obrázek 3.6: Závislost kvality na počtu bitů na pixel při použití formátu H.265 s různými intenzitami šumu.

V případě H.265 situace vypadá obdobně, primárním rozdílem jsou nižší datové toky potřebné pro reprezentaci dat. Zatímco pro uložení dat bez šumu stačilo kodeku 1,3 BPP, při amplitudě 8 byl třeba tok více než dvojnásobný, a to 3,1 BPP. Dále už nároky rostou pozvolněji, opět se zpomalující tendencí, konkrétně na 4,08 BPP, 4,86 BPP a nakonec 5,42 BPP. Stejně tak i vyšší granularita s sebou přinesla znovu přinesla pomalejší nárůst toků (od 1,99 BPP při amplitudě 8 přes 2,86 a 3,48 BPP až na 4,00 BPP při amplitudě

48).

Také průběh vývoje kvality tvarově odpovídá H.264. Na rozdíl od předchůdce však maxima kvality vzrostla a s nimi klesly datové toky, na kterých k nim dochází. I pro amplitudu 8 při nízké granularitě tentokrát dochází v maximu k mírnému zlepšení na kvalitu 0,97 (rozdíl indexu 0,02) v bodě 0,08 BPP. Pro další amplitudy pak nastává obvyklý posun do nižších toků a růst míry zlepšení na 0,95 pro amplitudu 16 (rozdíl indexu 0,17), 0,91 pro amplitudu 32 (rozdíl indexu 0,39) a 0,85 pro amplitudu 48 (rozdíl indexu 0,53).

Vyšší granularita opět vede k pozvolnějšimu stoupání, vzhledem k nižším tokům a menšímu prostoru však propady po maximu zůstaly strmé. Při amplitudě 8 dochází k nepatrnému zlepšení lámajícímu se na 0,33 BPP. Pro vyšší amplitudy jsou dosažené kvality 0,89 při 0,13 BPP (rozdíl indexu 0,04), 0,76 při 0,03 BPP (rozdíl indexu 0,1) a 0,73 při 0,007 BPP (rozdíl indexu 22). Tentokrát však v posledním jmenovaném bodě (0,007 BPP) dochází k situaci, kdy poměr datového toku a kvality videa s amplitudou šumu 48 dojde shody s kvalitativním poměrem při šumu o amplitudě 32.

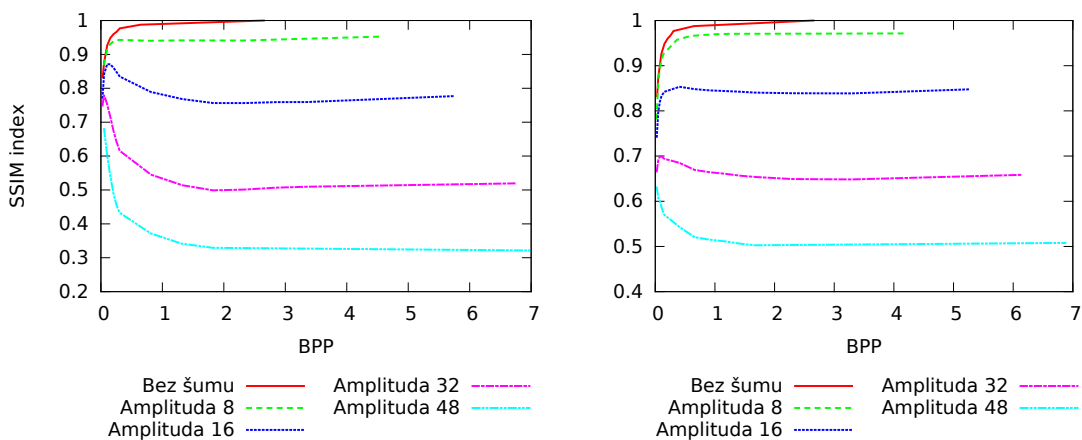


Obrázek 3.7: Závislost kvality na počtu bitů na pixel při použití formátu Motion JPEG 2000 s různými intenzitami šumu.

Vzhledem k absenci neklíčových snímků je vývoj kvality a datového toku u Motion JPEG 2000, vyneseno na grafech 3.7, výrazně pomalejší. Pro věrnou reprezentaci nezašuměného obrazu potřeboval 4 BPP, a tato hodnota se, stejně jako u formátů na DWT bázi, přidáním šumu dále zvýšila. Šum o amplitudě 8 a nízké granularitě si vyžádal 5,14 BPP. Postupným zvyšováním až k amplitudě 48 se tato hodnota změnila na 5,97 BPP, 6,82 BPP a nakonec 7,41 BPP. Opět je zde tedy trend postupně zpomalujícího růstu, prvotní skok oproti čistému videu však nebyl natolik markantní. Podobně probíhal také vývoj pro vyšší granularitu, kde se tok z 4,58 BPP dostal přes 5,30 a 5,94 BPP až na 6,57 BPP.

Pomalejší vývoj kvality se projevil také na maximech. Oproti předchozím formátům jsou i při nízké granularitě hladká, navíc se posunula do nepatrně vyšších toků a při snižování datového toku tentokrát před stoupáním dochází k mírnému poklesu vinou artefaktů. V ostatních směrech je však popis podobný. Při amplitudě 8 s nízkou granularitou dochází pouze k návratu na původní kvalitu 0,95 při 1,19 BPP, rozdíl indexů je tedy nulový. Pro vyšší amplitudy dosahují maxima hodnot 0,89 při 0,40 BPP (rozdíl indexů 0,11), 0,81 při 0,17 BPP (rozdíl indexů 0,29) a 0,75 při 0,06 BPP (rozdíl indexů 0,43). Posunuta jsou také maxima při vyšší granularitě. Pro amplitudu 8 dochází rovnou ke klesání kvality bez vý-

znamnějších změn. Pro amplitudy 16 a 32 se maxima nacházejí na 0,59 BPP s kvalitou 0,86 (rozdíl indexů 0,01) a 0,19 BPP s kvalitou 0,72 (rozdíl indexů 0,06). K největšímu zlepšení pak dochází při amplitudě 48 při 0,06 BPP s kvalitou 0,66 (rozdíl indexů 0,15).



Obrázek 3.8: Závislost kvality na počtu bitů na pixel při použití formátu DIRAC s různými intenzitami šumu.

V případě formátu DIRAC, reprezentovaného grafy 3.8, se průběh pro nízkou granularitu podobá H.264, jemuž má být dle svých tvůrců konkurentem. Přesto je však jeho kompresní poměr horší. Pro zakódování videa bez šumu potřeboval kodek 2,65 BPP. Se zařazováním jednotlivých amplitud se pak hodnota měnila od 4,58 BPP u amplitudy 8 přes 5,75 BPP a 6,77 BPP až na 7,54 BPP při amplitudě 48. Při práci se šumem vyšší granularity se průběh začíná podobat spíše formátu Motion JPEG 2000, s nímž DIRAC sdílí jádro ve formě diskretní vlnkové transformace. Při amplitudě 8 a vysoké granularitě potřeboval pro věrné uložení dat tok 4,17 BPP, pro následující amplitudy pak 5,28 BPP, 6,14 BPP a 6,88 BPP. Tím se svou efektivitou umístil na třetí místě právě mezi H.264 a Motion JPEG.

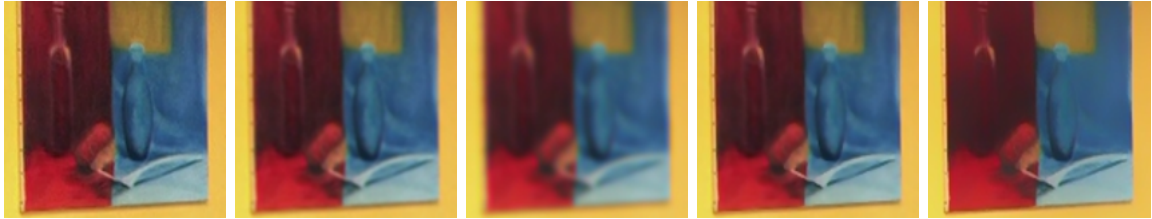
Také zde dochází kompresí k redukování šumu a podobně jako u JPEG 2000 předchází zvyšování kvality propad, v tomto případě je však zanedbatelný. Při amplitudě šumu 8 a nízké granularitě dochází pouze k pozastavení poklesu kvality trvajícím do 0,30 BPP. U vyšších amplitud už je možné pozorvat zlepšení ve formě 0,87 při 0,12 BPP (rozdíl indexu 0,09) pro amplitudu 16, 0,78 při 0,06 BPP (rozdíl indexu 0,26) pro amplitudu 32 a 0,68 při 0,05 BPP (rozdíl indexu 0,36) pro amplitudu 48. Pro vysoké granularity je míra redukce znatelně nižší. Pro amplitudu 8 opět dochází jen ke stagnování kvality do bodu 1,13 BPP a stejně tak u amplitudy 16 dochází jen k poklesu a opětovnému vyrovnání na 0,42 BPP. Stoupat začne kvalita až u amplitudy 32, kde dosáhne maxima 0,70 na 0,09 BPP (rozdíl indexů 0,04) a následně při amplitudě 48 dosáhne kvality 0,63 při 0,03 BPP (rozdíl indexů 0,12).

3.10 Srovnání komprese po zapojení filtrů pro redukcí šumu

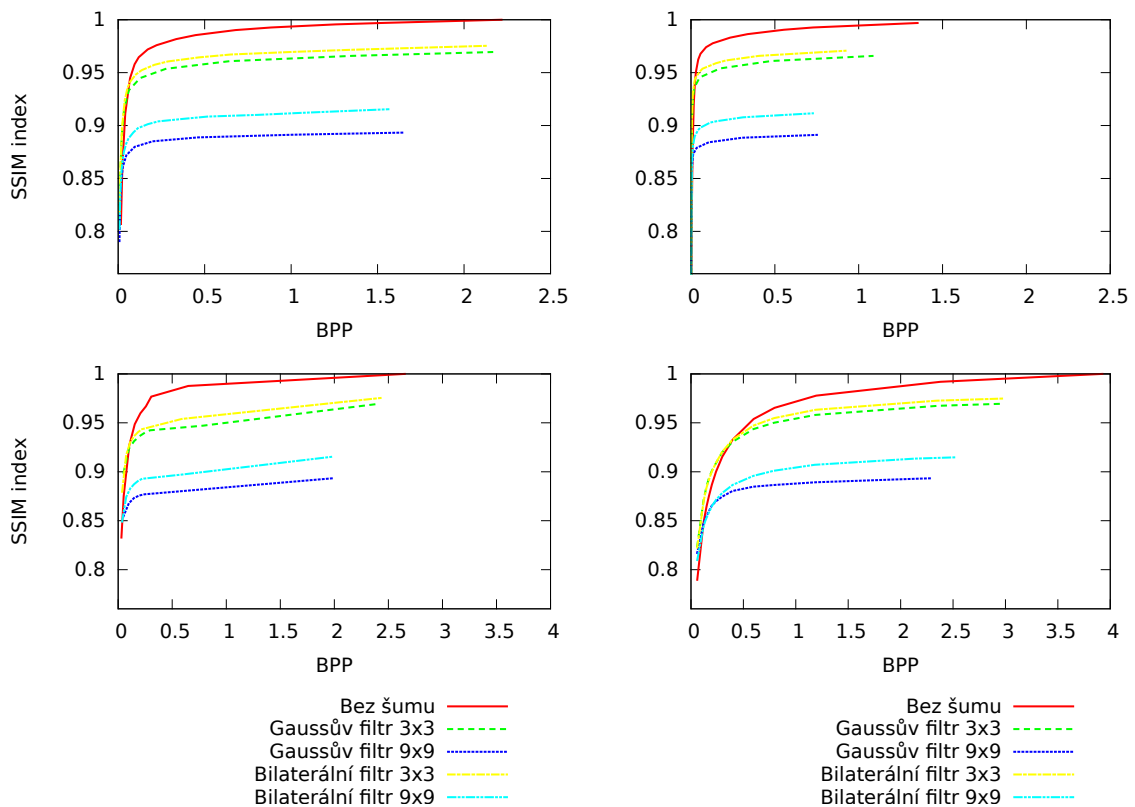
Posledním experimentem bylo zařadit před samotnou kompresí některý z filtrů běžně používaných pro redukování šumu v obraze. Záměrem bylo zjistit, zdali by částečné odstranění vysokých frekvencí – tedy za běžných okolností primárně informací o šumu zvyšujících entropii – nemohlo pomoci dosáhnout efektivnějšího využití datového toku za cenu zanedbatelné

ztráty celkové kvality, což by ve výsledku vedlo ke zlepšení kompresního poměru.

Aby experiment lépe reflektoval reálné situace, byla pro něj z testovací sady vyčleněna pouze videa obsahující hranou tvorbu, ve které se šum přirozeně vyskytuje vinou senzorů používaných k jejímu zachytávání. Pokus byl proveden s využitím dvou filtrů, Gaussova, způsobujícího rozmazání s omezeným šířením extrémů, a bilaterálního, který zachovává hrany, o dvou velikostech jádra (3×3 a 9×9).



Obrázek 3.9: Obraz po aplikaci různých variant filtrů pro redukci šumu. Zleva doprava nefiltrovaný obraz, Gaussův filtr 3×3 , Gaussův filtr 9×9 , bilaterální filtr 3×3 a bilaterální filtr 9×9 .



Obrázek 3.10: Závislost kvality na počtu bitů na pixel při použití filtrů pro odstranění šumu pro jednotlivé formáty (horní řádek zleva H.264 a H.265, spodní DIRAC a Motion JPEG 2000).

Na grafech 3.10 je vidět srovnání výsledků komprese při použití jednotlivých variant filtrů pro individuální formáty. Přestože datový tok potřebný pro reprezentaci dat skutečně

poklesl, spolu s ním se snížila také kvalita. Zásadnější pokles při stejné velikosti jádra znamenal Gaussův fltr, který nebere ohled na hrany a do větší míry tak rozrušuje strukturu obrazu, na kterou se lidské oko a hodnotící algoritmus SSIM zaměřuje. Předpokládané zlepšení kompresního poměru je skutečně možné pozorovat, v případě H.264, H.265 a DIRAC je však nepatrné a dochází k němu jen na nejnižších tocích. Dále platí, že zlepšení dříve nastává u menších konvolučních jader (pro H.264 pod 0,06 BPP, H.265 pod 0,03 BPP a DIRAC dokonce už pod 0,1 BPP), pro větší jádra je přínos mizivý a projevoval se začal velice pozdě (pro H.264 od 0,03 BPP, H.265 0,008 BPP a DIRAC vůbec). Odlišná je situace v případě Motion JPEG 2000. Jelikož má větší nároky na datový tok a kvalita u něj proto klesá pomaleji, dochází také k průniku kompresních poměrů později. Ke zlepšení kompresního poměru tak dochází pro malá jádra už od 0,4 BPP a v případě větších jader od 0,1 BPP.

Ze zjištěných informací vyplývá, že zařazení filtru pro redukci šumu skutečně může vést ke zlepšení kompresního poměru při zpracování videí obsahujících hranou tvorbu. Tato skutečnost však platí jen pro velice nízké datové toky a vhodně zvolené velikosti jádra.

Kapitola 4

Závěr

Cílem práce bylo seznámit se s metodami využívanými při kompresi videa, různými formáty, které je využívají, a zjistit, jak se vzájemně ovlivňují šum přítomný ve videu a datový tok, se kterým je toto video zkomprimováno, především pak jak se výsledky liší mezi diskretní kosinovou transformací a diskretní vlnkovou transformací. Pro tento účel byla vybrána čtveřice formátů H.264, H.265 (zastupujících DCT), DIRAC a Motion JPEG 2000 (reprezentujících DWT), s jejichž pomocí byla s různými datovými toky zkomprimována videa obsahující rozličné varianty Gaussovského šumu, standardně používaného jakožto aproximace šumů přirozených. Tyto výsledky byly dále srovnány a vyhodnoceny.

Potvrdilo se, že nároky na datový tok s rostoucí amplitudou šumu stoupají. V případě DCT i DWT dosáhly kodeky lepších výsledků při kompresi videa obsahujícího šum o vyšší granularitě. Jakožto nejefektivnější po stránce poměru datového toku a kvality se ve všech experimentech jevil formát H.265, následován svým předchůdcem H.264. Kodeky stojící na diskretní vlnkové transformaci se naopak ukázaly jako méně datově efektivní – DIRAC skončil na třetím místě a Motion JPEG na čtvrtém.

S klesajícím datovým tokem, a tedy ztrátou obrazových informací, se kvalita videa snižuje. V případě videí obsahujících nežádoucí šum však toto striktně neplatí. Na nízkých datových tocích dochází k anomálii, kdy kvalita naopak na čas stoupá. To je dáno způsobem komprese, kdy dochází k největšímu zkreslení na nejvyšších frekvencích, kde se obvykle vyskytuje mimo jiné právě šum. K nejvýraznější redukci dochází při nízké granularitě šumu, v případě formátů založených na DCT takřka pro všechny intenzity (příčemž více pro nižší), u DWT pak převážně pro vyšší amplitudy.

Sekundárním předmětem zkoumání bylo, jaký efekt bude mít v kontrastu k přítomnosti šumů aplikace filtru pro jejich redukci. Předpoklad, že by u hrané tvorby mohl vést ke snížení entropie dané senzorickým šumem z doby zachytávání za cenu zanedbatelné ztráty kvality, se sice potvrdil, lepších výsledků však kodéry dosáhly pouze pro velmi nízké datové toky a při použití malých konvolučních jader filtrů.

Ukázalo se tedy, že efektivnějšími formáty pro kompresi videa jsou ty založené na diskretní kosinové transformaci, především pak H.265. Dalším důležitým zjištěním bylo, že komprese lze využívat nejen jako nástroje pro úsporu datového objemu, ale také, zvláště v případě DCT, jako efektivní metodu redukce šumu v obraze. V budoucnu by bylo možné na výzkum navázat formou experimentů s dalšími druhy šumu a redukčních filtrů a zkoumat, jak se chování formátů změní.

Literatura

- [1] Acharya, T.; Tsai, P.-S.: *JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI architectures*. John Wiley & Sons, Inc., 2005, ISBN 0-471-48422-9.
- [2] Bovik, A.: *Handbook of Image and Video Processing*. Elsevier Academic Press, 2005, ISBN 0-12-119792-1.
- [3] Farooque, M. A.; Rohankar, J. S.: Survey on Various Noises and Techniques for Denoising the Color Image. *International Journal of Application or Innovation in Engineering & Management*, ročník 2, č. 11, listopad 2013: s. 1649–1668, ISSN 2319-4847.
- [4] Hunter, J.: Digital cinema reels from motion JPEG2000 advances [online]. *EE Times*. http://www.eetimes.com/document.asp?doc_id=1201549, 2003-06-01 [cit. 2016-01-22].
- [5] ISO/IEC WG1: JPEG 2000 Software. <http://jpeg.org/jpeg2000/software.html>, 2015 [cit. 2016-01-22].
- [6] ITU-T: *High efficiency video coding*. H.265, duben 2015.
- [7] ITU-T: *Advanced video coding for generic audiovisual services*. H.264, únor 2014.
- [8] Liu, T.-M.; et al.: Energy and area efficient hardware implementation of 4K Main-10 HEVC decoder in Ultra-HD Blu-ray player and TV systems. *2015 IEEE International Conference on Multimedia and Expo (ICME)*, červen 2015: s. 1–6, ISSN 1945-7871.
- [9] Mallat, S.: *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, třetí vydání, 2009, ISBN 978-0-12-374370-1.
- [10] Pratt, W. K.: *Digital Image Processing: PIKS Inside*. John Wiley & Sons, Inc., třetí vydání, 2001, ISBN 0-471-22132-5.
- [11] Salomon, D.; Motta, G.: *Handbook of Data Compression*. Springer-Verlag London, páté vydání, 2010, ISBN 978-1-84882-903-9.
- [12] Sullivan, G. J.; et al.: Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, ročník 22, č. 12, prosinec 2012: s. 1649–1668, ISSN 1051-8215.
- [13] Tomasi, C.; Manduchi, R.: Bilateral filtering for gray and color images. *Sixth International Conference on Computer Vision*, leden 1998: s. 839–846.

- [14] Wang, Z.; Bovik, A. C.; et al.: Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, ročník 13, č. 4, duben 2004: s. 600–612, ISSN 1057-7149.
- [15] Özenli, D.; Pazarci, M.: Performance Analysis of Dirac Video Codec in Different Motion Vector Accuracies and Wavelet Lifting Decompositions. *ELMAR, 2011 Proceedings*, září 2011, ISSN 1334-2630.