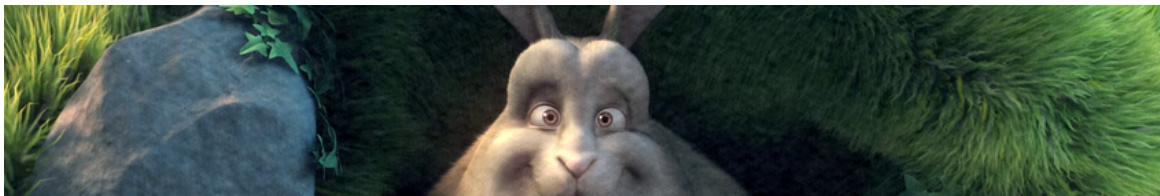


Vliv šumu na kompresi videa

Matyáš Anton*



Abstrakt

Videa jsou jakožto obrazový signál v různých fázích zpracování náchylná na vznik zkreslení ve formě šumu. Tento příspěvek zkoumá, jakým způsobem může přítomnost šumu ovlivnit výsledky komprese obrazových dat a jaký dopad má naopak ztrátová komprese na přítomný šum. Pro zkoumání byla zvolena čtveřice v současnosti používaných formátů H.264, H.265, Motion JPEG 2000 a DIRAC, reprezentujících různé postupy komprese, především pak pro porovnání efektu diskrétní kosinové a diskrétní vlnkové transformace.

Klíčová slova: Komprese videa — Redukce šumu — Diskrétní kosinová transformace — Diskrétní vlnková transformace

Přiložené materiály: N/A

*xanton03@stud.fit.vutbr.cz, Fakulta informačních technologií, Vysoké učení technické v Brně

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 přimícháním šumu. Současně objemnost těchto obrazových dat nese potřebu komprimovat je pro snazší manipulaci.

Cílem tohoto příspěvku je zhodnotit, jak se přítomnost šumu a kompresní formáty 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 [1], a Motion JPEG 2000 a DIRAC vycházející z diskrétní vlnkové transformace Handbook. Ve všech případech se jedná o soudobé formáty reprezentující různé přístupy ke kompresi videa.

Pro účely experimentování vznikla dvojice programů umožňující generování videa zkresleného šumem o různé amplitudě a granularitě, následnou kompresi jednotlivými kodeky a kvalitativní porovnání výsledku vůči originálu.

Na základě získaných výsledků práce hodnotí vhodnost formátů pro různé šumy a navrhuje další možné využití komprese z těchto zjištění plynoucích.

2. Základní principy

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 jejich dekódování. Rozdílové naproti tomu nesou jen rozdíl oproti svým sousedům – tzv. referenčním snímkům. 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.

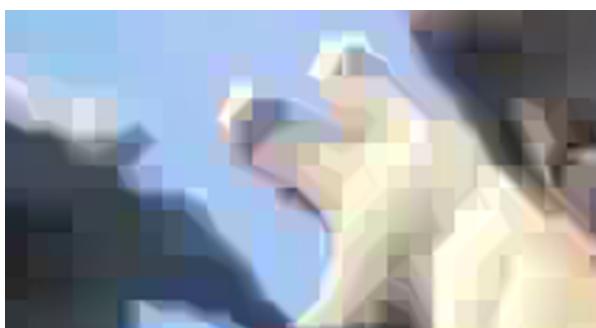
Použití rozdílových snímků však s sebou nese ještě jednu potenciální komplikaci. V případě, že je video tvořeno klíčovými snímky a dojde k poškození nebo ztrátě dat, projeví se změna jen lokálně na krátký

okamžík – 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). [2]

2.1 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 rozdílu 8×8 , velikost makrobloků pro chromatické složky se odvíjí od podzorkování, např. pro 4:2:0 to bude 8×8 . Souvislé úseky makrobloků se stejným jasem pak představují tzv. řezy (slices).

Při kompresi je poté na jednotlivé bloky 8×8 aplikována diskrétní 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.



Obrázek 1. Ukázka blokového efektu při nízkém datovém toku u formátu H.264.

Koeficienty jsou dále kvantovány podle kvantizační tabulky. Ta udává, jakými hodnotami mají být jednotlivé koeficienty poděleny a následně 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. 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. Většina kodeků se však snaží tomuto efektu nějakou formou předcházet. [2].

2.2 Diskrétní vlnková transformace (DWT)

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

Snímek může být pro každou barevnou složku rozsekán na stejně velké, samostatně zpracovávané dlaždice. Na ty je aplikována diskrétní vlnková transformace, čímž vznikne několik podpásem koeficientů. Standard specifikuje dva druhy vlnek – 9/7, využívající hodnoty s desetinnou čárkou (pro ztrátovou kompresi), a celočíselnou 5/3 (možno využít pro ztrátovou i bezztrátovou kompresi). Vzniklé koeficienty jsou dále kvantovány. V rámci každého pásma jsou hodnoty vyděleny velikostí kvantizačního kroku (může být pro jednotlivá pásma individuální) a výsledky se zaokrouhlí [2].

2.3 Využití prostorové a časové redundance

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, rádek nad nebo kombinace obou) a kóduje se jen chyba, tedy jak se tato predikce lišila oproti skutečnosti.

Naproti tomu při kódování rozdílových snímků se využívá technika kompenzace pohybu (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 makroblokách. 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 pixe-

lu). Desetinná přesnost s sebou nese nutnost interpolace a z toho plynoucí vyšší výpočetní i paměťovou náročnost.

Algoritmy pro určování pohybových vektorů jsou netriviální a obecně platí, že nemusí vždy určit související oblasti správně. Z toho plynne mimo jiné to, že se pro různé implementace mohou výsledné pohybové vektory značně lišit. V dnešní době se obvykle využívají jak prostorové (intra snímky, predikce makrobloků), tak časové redundance (inter snímky, kompenzace pohybu). Kodeky, které tyto dva přístupy kombinují, se označují jako hybridní [2].

2.4 Zkoumané formáty videa

H.264 je v současné době značně rozšířený formát. Nachází uplatnění na Blu-ray discích, v digitálním televizním vysílání, ale také jako jeden z hlavních formátů pro přenos internetového videa. Při kódování využívá I, P a B snímků, časových razítek a členění snímků na řezy složené z makrobloků. 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 získání koeficientů je využita diskrétní kosinová transformace, před kvantizací jsou však ještě hodnoty v tabulce násobeny koeficientem q [2].

H.265 byl vytvořen jako nástupce dnes rozšířeného H.264. Bloky a makrobloky jsou zde nahrazeny CTU (coding tree unit) a CTB (coding tree block), 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 znova stojí na diskrétní kosinové transformaci a kvantování koeficientů [3].

Motion JPEG 2000 je formát postavený na základu JPEG 2000, který je určen pro kódování statických obrazových dat. Využívá dělení na dlaždice, zpracování diskrétní vlnkovou transformací a kvantování. Pro kódování videa se zde, na rozdíl od běžných formátů, nepoužívá žádná forma časové predikce, všechny snímky jsou typu I, a tedy kódovány nezávisle [4].

DIRAC byl vyvinut jakožto konkurent v té době již rozšířeného standardu H.264. 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 [5]. Jinak ovšem funguje na stejných principech jako ostatní hybridní kodéry –

používá I, P, B snímky a kompenzaci pohybu s využitím pohybových vektorů.

3. Pomocné nástroje

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 knihovny s rozhraním pro jazyk C, z nichž stojí za zmínku především libavcodec zaštiťující kódování a dekódování.

Klíčovým prvkem pro provedení experimentu je schopnost programu generovat do obrazu šum. Ten se přidává aditivně. Intenzita pro konkrétní pixel leží v intervalu $(-A, A)$, kde A je uživatelem zvolená amplituda. Konkrétní hodnota 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 Gaussovou funkci.

Jakožto metoda pro hodnocení kvality výsledných videí byl zvolen algoritmus SSIM (Structural SIMilarity) index. Tento algoritmus navrženy Zhou Wangem a Alanem Bovikem se vyznačuje faktem, že nehodnotí kvalitu obrazu čistě z hlediska numerických rozdílů, ale zaměřuje se na to, jak je obraz vnímán člověkem. Také proto se zaměřuje na jasovou složku, na kterou je lidské oko nejcitlivější, a předpokládá vstup ve stupních šedi [6]. V projektu byla využita volně šířitelná implementace v jazyce C++ využívající grafickou knihovnu OpenCV. Jejím autorem je Rabah Mehdi¹. Výsledkem jsou reálné hodnoty v intervalu od 0 do 1, kde 1 značí naprostou shodu a 0 žádnou.

3.1 Implementace programů pro přidání šumu, kódování a porovnání videí

Pro zašumění, komprezi a srovnávání videí byly vytvořeny dva programy postavené na frameworku FFmpeg. 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, amplitudu šumu, směrodatnou odchylku, název požadovaného výstupního kodeku a hodnotu určující kvalitu výstupu, specifickou pro konkrétní kodék.

Vstupní soubor je otevřen, nalezne se videostopa a na základě ní je vyhledán odpovídající dekodér. Následně je podle zvolených parametrů nastaven výstupní

¹<http://mehdi.rabah.free.fr/SSIM/>

kodér. Poté může začít dekódovací smyčka. Snímek je nejprve převeden do barevného formátu YCbCr 4:2:0, společného pro všechny 4 výstupní kodeky, a následně zašuměn postupem popsaným dále. 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 dekódovací smyčky spustit smyčku další, která opakováně volá kódovací funkci s nulovým ukazatelem, dokud od ní nedostane informaci o dokončení kódování, čímž činnost `recode` končí. Výsledkem je komprimovaný video soubor s přidaným šumem.

Následně přichází na řadu druhý program, `compare`. Jeho parametru je 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 dekódovací 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 tabulátoru a mezi záznamy o snímcích se nachází znak nového řádku.

4. Vztah mezi šumem a kvalitou komprimovaného videa

Cílem experimentů bylo zjistit vzájemný vliv šumu a datového toku videa při kompresi do čtyř rozdílných formátů pro varianty šumu o dvou různých intenzitách (méně zkreslujícího s amplitudou 16 a více s amplitudou 32) s nízkou nebo vysokou granularitou (kdy je krom pixelu ovlivněno také okolí do vzdálenosti 1, resp. 5). Zkoumána je vždy kvalita SSIM zkomprimovaného videa se šumem vůči nezašuměnému originálu představujícímu 100 %.

Testy byly prováděny na 10 vzorcích, reprezentujících různé druhy obrazových dat, a výsledné hodnoty následně zprůměrovány. Zastoupen byl živý obraz, čerpaný z otevřeného filmu *Valkama*², dále počítačová 3D animace pocházející z otevřeného filmu *Big Buck Bunny*³ a nakonec dvojrozměrná animace z bezplatné-

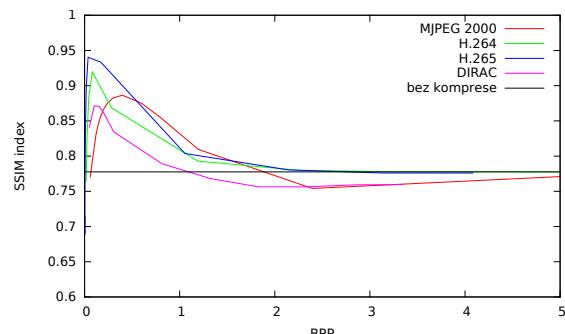
ho filmu *Sita Sings the Blues*⁴. Vzorky byly v rozlišení 848×480 pixelů a každý o délce zhruba 1,5 minuty.

4.1 Srovnání efektu šumu s nízkou granularitou na různé formáty



Obrázek 2. Obraz s různými amplitudami šumu o nízké granularitě, zleva doprava 0, 16 a 32.

První experiment měl za cíl zjistit, jakých výsledků dosáhnou formáty pro různé datové toky za přítomnosti šumu o nízké granularitě. Předpoklad byl, že s amplitudou šumu poroste také potřebný datový tok, a současně s nižším poskytnutým datovým tokem ubyde detailů a s nimi i šumu. Nejvyšší nároky na datový tok lze očekávat od formátu Motion JPEG 2000, který na rozdíl od ostatních kóduje všechny snímky jako klíčové.



Obrázek 3. Srovnání závislosti kvality na počtu bitů na pixel při amplitudě šumu 16 a nízké granularitě pro čtveřici formátů H.264, H.265, Motion JPEG 2000 a DIRAC.

Na grafu 3 je možné vidět srovnání formátů při aplikaci nižšího šumu. Na nejnižších datových tocích dochází těsně před propadem, daným vznikem kompresních artefaktů, k prudkému nárůstu kvality, což znamená redukci šumu v obraze. Nejznatelněji se tento jev projevuje pro formát H.265, následován H.264. Z toho vyplývá, že k efektivnější redukci na těchto tocích dochází při použití postupů založených na DCT. S rostoucím datovým tokem však zůstává stále více šumu zachováno. V případě formátů Motion JPEG 2000 a DIRAC dochází k redukci nižší. To je částečně zapříčiněno vznikem artefaktů, jejichž přítomnost je možné vidět také v tocích mezi 2 a 3 BPP, kde kvalita oproti původnímu obrazu se šumem klesá ještě hlouběji.

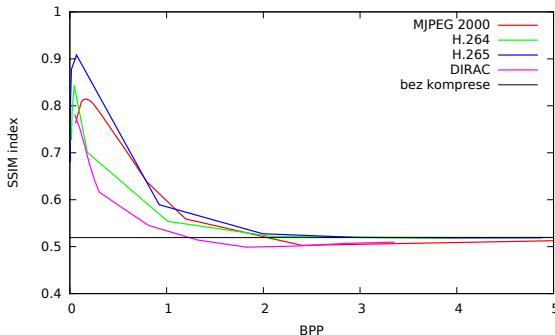
K věrné reprezentaci šumu, v tomto případě odpovídající kvalitě 0,78 %, se dříve přibližují DCT

²<http://www.valkama.com/>

³<https://peach.blender.org/>

⁴<http://sitasingingstheblues.com/>

formáty, v případě H.265 při hodnotě 2,3 BPP a u H.264 při 2,8 BPP. Horšího kompresního poměru dosáhly vlnkové formáty s 4,5 BPP pro Dirac a 5,9 BPP pro JPEG 2000.



Obrázek 4. Srovnání závislosti kvality na počtu bitů na pixel při amplitudě šumu 32 a nízké granularitě pro čtevěcí formátů H.264, H.265, Motion JPEG 2000 a DIRAC.

Graf 4 obsahuje srovnání formátů za přítomnosti vysokého šumu. Pro nízké datové toky opět nejlépe dopadl formát H.265. Tentokrát však se zdáleka ne tak velkým předstihem. Pro ostatní kodeky začínají být výsledky poměrně srovnatelné, byť u vlnkových opět mezi redukcí a věrnou reprezentací dochází k vyššímu poklesu kvality díky šumu.

Paměťové nároky pro věrnou reprezentaci s vyšší amplitudou šumu vedoucího ke kvalitě 0,52 % skutečně vzrostly. Pořadí efektivity kodeků zůstalo stejné, tentokrát však H.265 potřeboval 2,8 BPP, H.264 3,1 BPP. DIRAC si vyžádal tok 4,8 a JPEG 2000 dokonce 6,7 BPP.

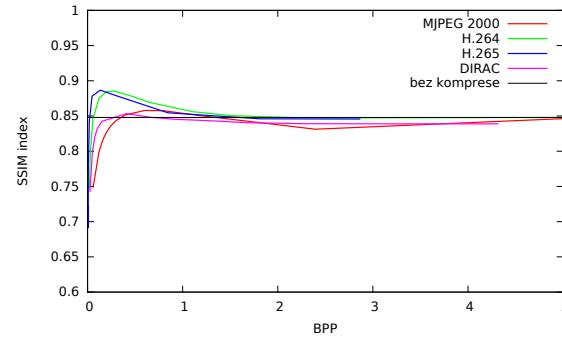
4.2 Srovnání efektu šumu s vysokou granularitou na různé formáty



Obrázek 5. Obraz s různými amplitudami šumu o nízké granularitě, zleva doprava 0, 16 a 32.

Při zvýšení granularity dochází k tomu, že je každý pixel obrazu ovlivněn nejen lokálně, ale taky šumem z okolních pixelů, v případě druhého experimentu do vzdálenosti 5, a dochází tak ke shlukování efektu do větších útvarů. Opět vycházel z předpokladu, že s rostoucí amplitudou porostou také nároky na tok.

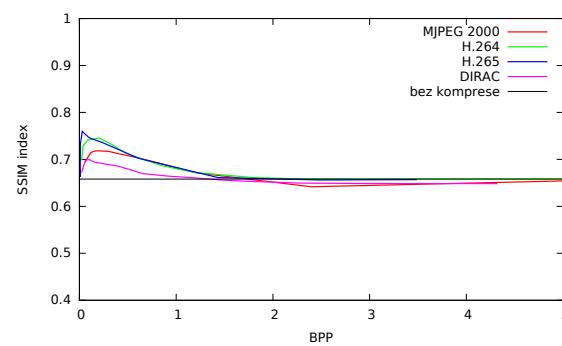
Jak je možné vidět porovnáním grafů 3 a 6, případně 4 a 7, šum s vyšší granularitou vede z pohledu lidského oka k menšímu zkreslení obrazu nežli s granularitou nízkou. Pro nízkou amplitudu dosahuje kvalita 0,85



Obrázek 6. Srovnání závislosti kvality na počtu bitů na pixel při amplitudě šumu 16 a vysoké granularitě pro čtevěcí formátů H.264, H.265, Motion JPEG 2000 a DIRAC.

%, pro vyšší 0,66 %. Současně již však jeho 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.

Věrné reprezentaci se opět nejrychleji přiblížil formát H.265, tentokrát již při 1,8 BPP. Opět byl blízce následován H.264 s hodnotou 2,1. Zbylé dva kodeky však vlivem kompresních artefaktů potřebovaly pro přiblížení podobné toky jako v případě nízké granularity, 4,5 BPP pro DIRAC a 5,3 BPP pro Motion JPEG.



Obrázek 7. Srovnání závislosti kvality na počtu bitů na pixel při amplitudě šumu 32 a vysoké granularitě pro čtevěcí formátů H.264, H.265, Motion JPEG 2000 a DIRAC.

Také pro vyšší amplitudy se vliv šumu i možnost jeho následné redukce zmenšily. Opět na nízkém toku nejvyšší kvality dosahují DCT kodeky, zde se však již začne projevovat mírná redukce také ze strany formátů založených na DWT.

Pro DCT formáty potřebný tok o něco vzrostl – H.265 2,5 BPP a H.264 3 BPP. V případě DWT se situace příliš nezměnila a dosáhl hodnot 4,7 pro DIRAC a 5,8 pro Motion JPEG 2000.

5. Závěry

Cílem této práce bylo zjistit, jak se vzájemně ovlivňují šum přítomný ve videu a datový tok, se kterým je toto video zkomprimováno, v případě různých formátů, především pak jak se výsledky liší mezi diskrétní kosi-novou transformací a diskrétní vlnkovou transformací.

Po stránce nároků na datový tok platí, že s rostoucí amplitudou šumu stoupají. V případě diskrétní kosi-nové transformace dosáhly kodeky lepších výsledků při vyšší granularitě šumu, naproti tomu na vlnkové formáty granularity vliv prakticky neměla. Jakožto nejfektivnější po stránce poměru toku a kvality se po všech experimentech jeví H.265, následován obvykle H.264. Kodeky stojící na diskrétní vlnkové transfor-maci se naopak ukázaly jako méně datově efektivní.

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

Ukázalo se tedy, že kompresi videa do formátů na bázi DCT je možné využívat také jako efektivní metodu redukce šumu v obraze. Dalším možným pos-tupem do budoucna by bylo otestovat efektivitu komprese a této redukce v kombinaci s některou standardně používanou metodou odstraňování šumu a výsledky porovnat.

Poděkování

Rád bych poděkoval Ing. Davidu Bařinovi za rady a vedení při tvorbě této práce.

Literatura

- [1] ITU-T. *High efficiency video coding*. H.265, duben 2015.
- [2] David Salomon and Giovanni Motta. *Handbook of Data Compression*. Springer-Verlag London, 2010.
- [3] Gary J. Sullivan and et al. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, prosinec 2012.
- [4] Jill Hunter. 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] Thomas Davies. Dirac: Wavelet trans-form. http://dirac.sourceforge.net/documentation/algorithm/algorithm/wlt_transform.xht, 2008-02-06 [cit. 2016-1-22].
- [6] Zhou Wang, Alan Conrad Bovik, and et al. Im-age quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, duben 2004.