VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SROVNÁNÍ MODERNÍCH FORMÁTŮ PRO ZTRÁTOVOU KOMPRESI OBRAZU

COMPARISON OF MODERN LOSSY IMAGE COMPRESSION FORMATS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **MAREK KALABZA**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. DAVID BAŘINA, Ph.D.**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2018/2019

Zadání bakalářské práce

Student: **Kalabza Marek**
Program: Informační technologie
Název: **Srovnání moderních formátů pro ztrátovou kompresi obrazu**
Comparison of Modern Lossy Image Compression Formats
Kategorie: Zpracování obrazu

Zadání:

1. Seznamte se s významnými formáty pro ztrátovou kompresi obrazu. Vybrané formáty popište. Zaměřte se na možnosti a parametry jednotlivých formátů.
2. Na vhodných datasetech tyto formáty vhodnými metrikami srovnejte. Zaměřte se závislost kvality na kompresním poměru, možnosti progresivního přenosu, kompresní artefakty, možnosti zabezpečení dat proti korupci. Dále se zabývejte výpočetními nároky běžně používaných implementací.
3. Diskutujte výsledky srovnání a vhodnosti použití jednotlivých formátů k různým účelům.

Literatura:

- William B. Pennebaker and Joan L. Mitchell, JPEG: Still Image Data Compression Standard. ISBN: 978-0442012724
- JPEG2000 Image Compression Fundamentals, Standards and Practice, by David Taubman and Michael Marcellin (ISBN-13: 978-0792375197)
- ITU-T T.81
- ITU-T T.800

Pro udělení zápočtu za první semestr je požadováno:

- Bod 1, dále rozpracovaný bod 2.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Bařina David, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 5. listopadu 2018

Abstrakt

V této práci se porovnávají moderní metody komprese digitálních obrazů na daných datasetech v oblasti kvality. První část práce popisuje princip ztrátové komprese a poté samotné komprese formátů JPEG, Lepton, JPEG 2000, JPEG XR, WebP, BPG, HEIF, H.265 intra frame, H.264 intra frame a AV1. V druhé části se práce zaměřuje na návrh testování, srovnání výsledků a vyhodnocení. Druhá část také popisuje použité metriky kvality (PSNR, PSNR-YUV, SSIM), zabývá se možnými artefakty při kompresi formátů a poukazuje na chování formátů po korupci dat. Na základě testování a měření bylo zjištěno, že mezi nejlepší kompresní formát z testovaných ztrátových formátu patří formát AV1 a to konkrétně na datasetu obrazů s vysokým rozlišením, datasetu skenovaných dokumentů a datasetu map.

Abstract

This thesis compares modern compression methods of digital images on given datasets in terms of quality. First part describes the principle of lossy compression and then compression of JPEG, Lepton, JPEG 2000, JPEG XR, WebP, BPG, HEIF, H.265 intra frame, H.264 intra frame and AV1. Second part focuses on testing, comparison of results and evaluation. It also describes quality metrics (PSNR, PSNR-YUV, SSIM), deals with possible artefacts created during compression and points out the behavior of formats after data corruption. Based on testing and measurement, it was found that the best compression format from the tested lossy image formats is the AV1 format, specifically on the high resolution dataset, the scanned document dataset, and the map dataset.

Klíčová slova

ztrátová komprese obrazu, JPEG, JPEG 2000, Lepton, JPEG XR, WebP, BPG, HEIF, H.265 intra frame, H.264 intra frame, AV1, Huffmanovo kódování, Aritmetické kódování, artefakty, PSNR, PSNR_{YUV}, SSIM, korupce dat

Keywords

lossy image compression, JPEG, JPEG 2000, Lepton, JPEG XR, WebP, BPG, HEIF, H.265 intra frame, H.264 intra frame, AV1, Huffman coding, Arithmetic coding, artifacts, PSNR, PSNR_{YUV}, SSIM, data corruption

Citace

KALABZA, Marek. *Srovnání moderních formátů pro ztrátovou kompresi obrazu.* Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. David Bařina, Ph.D.

Srovnání moderních formátů pro ztrátovou kompresi obrazu

Prohlášení

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

.....
Marek Kalabza
12. května 2019

Poděkování

Chtěl bych poděkovat vedoucímu bakalářské práce Ing. Davidovi Bařinovi, Ph.D. za ochotu a rady při konzultacích. Dále bych chtěl poděkovat své rodině za podporu po dobu studia.

Obsah

1 Úvod **2**

2 Ztrátová komprese obrazu **3**
2.1 JPEG 7
2.2 JPEG2000 9
2.3 JPEG XR 11
2.4 WebP 13
2.5 H.264 intra frame / AVC 15
2.6 H.265 intra frame / HEVC 16
2.7 AV1 intra 19

3 Srovnání formátů **22**
3.1 Datasety 22
3.2 Metody vyhodnocení kvality komprese 22
3.3 Srovnání vlastností ztrátových formátů 25
3.4 Testování 28

4 Závěr **44**

Literatura **45**

A Obsah přiloženého DVD **48**

Kapitola 1

Úvod

Bakalářská práce obsahuje popis a srovnání moderních i základních ztrátových formátů komprese obrazu. Téma mi bylo blízké, protože mě zajímají nové moderní ztrátové kompresní formáty obrazu.

V současnosti se všude využívá digitalizace a obzvláště digitální obrázky, které jsou ve velkém měřítku využívány a stále se s postupným rozvojem technologií rozšiřují. Digitální obrázky jsou taky všudypřítomné, ať je to samotný internet, kde na každé větší stránce jsou fotografie, nebo telefony, fotoaparáty, tablet, či počítače. Každé zařízení anebo služba, má své výhrady na kvalitu, velikost, či formát obrazu, jež zmíněná skupina podporuje.

Již od roku 1992 je nejpoužívanějším obrázkovým formátem stále ztrátový formát JPEG, rozhodl jsem se proto ukázat lepší varianty ztrátového formátu, jež mohou být vice přínosné pro dané služby, či pro vybraná zařízení. Pro porovnání jsem tedy vybral tyto konkrétní formáty JPEG, Lepton, JPEG 2000, JPEG XR, WebP, BPG, HEIF, H.265 intra frame, H.264 intra frame a AV1. Výše uvedené formáty budou testovány na základě vlastního programu v Pythonu.

Cílem mé práce je vybrat vhodné moderní i nejrozšířenější ztrátové formáty, srovnat jejich závislost kvality na kompresním poměru nad jednotlivými datasety a popsat vyskytující se artefakty.

Práce nejprve čtenáře obeznámí se ztrátovou kompresí obrazu, následně popíše kompresi u jednotlivých formátů. V další kapitole práce poukáže na testovací datasety, popisuje použité měřící metody kvality obrazu, srovnává vlastnosti formátů a poukáže na tvořící se artefakty. Dále se tato kapitola zabývá chováním formátů při korupci dat, následně je zmíněn proces samotného testování kvality obrazu vůči velikosti komprese a jsou poskytnuty výsledky testování v podobě grafů. V závěrečné kapitole je shrnuta výsledná práce a je navrženo doporučení pro nejlepší formáty k jednotlivým datasetům.

Kapitola 2

Ztrátová komprese obrazu

Ztrátová komprese obrazu je proces, při němž dochází ke ztrátě informací v obraze. Obrazy jsou většinou ukládané pomocí rastrové grafiky. Rastrová grafika ukládá informace v obraze po bodech (pixelech). Každý jednotlivý bod má přesnou svou polohu a barvu, která náleží barevnému modelu. Právě nejčastějším použitým barevným modelem je RGB model, jenž není příliš vhodný pro kompresní techniku. První proces komprese zpravidla nejprve provede transformaci barev do vhodného barevného modelu 2 a následně proběhne podvzorkování 2. Tyto prvotní procesy využívají vlastnosti sítnice lidského oka, které je méně citlivé na barevné složky obrazu, ale naopak je zase citlivější na intenzitu jasu. Ztráta určitých informací má výhodu a to tu, že dojde ke razantnímu snížení velikosti zkomprimovaných dat. Mnohdy má obraz nadbytečné informace, které nejsou zapotřebí, jako je například velké rozlišení obrazu, jež lidské oko neumí rozlišit. V této kapitole jsem čerpal z těchto zdrojů [5] a [7].

Obrázek 2.1: Obrázek vlevo je vznikem 15 úrovní komprese a napravo je důsledek 90 úrovní komprese.

Barevné modely

Barevné modely slouží pro zjednodušení záznamu barevné informace. Pro věrnou reprodukci barvy nějakého objektu je nutné zaznamenat v každém bodu tohoto objektu spektrální křivku. Soubor základních barev, pravidla jejich míchání a měněné barevné charakteristiky

jsou definovány pomocí barevných modelů. Mezi nejdůležitější barevné modely v oblasti ztrátové komprese patří RGB model a Y'C_bC_r model.

RGB

RGB je anglická zkratka, jež vychází ze tří anglických slov. Jedná se o 3 základní barvy, a těmi jsou červená, zelená a modrá, ty označujeme jako primární. Dnešní počítače využívají RGB model s 24bity na jeden pixel. Každá složka je tedy vyjádřena pomocí čísla od 0 do 255. Kombinací těchto složek se definuje barva.

YUV

RGB model není vhodný pro ztrátovou kompresi obrazu, jelikož jednotlivé složky se často mění mezi sousedními pixely. Změna hodnoty změní jak barvu, tak i jas daného bodu. Proto YUV model rozděluje složky na jasové a barvonosné. Při kompresi lze potom u složek vyjadřující barvu dosáhnout menší nepřesnosti a přitom zachovat jas. Model YUV je často označován jako Y'C_bC_r. Tyto dva modely však mají rozdílné výpočty složek. Jelikož YUV model u barvonosných složek U a V může nabývat kladných i záporných hodnot, kde hodnota 0 představuje černobílý obraz, tak Y'C_bC_r může nabývat pouze z hodnot 0 až 255. Převod z RGB na YUV model je popsán v 2.1.

Převod z RGB na YUV:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.1}$$

Y'C_bC_r

Je model, který vzešel z modelu YUV, aby zjednodušil kódování a kompresi. Model Y'C_bC_r se přitom nepoužívá pro samotné zobrazování obrazu. Převod ve formátu JFIF z 8bit RGB modelu na Y'C_bC_r je následující:

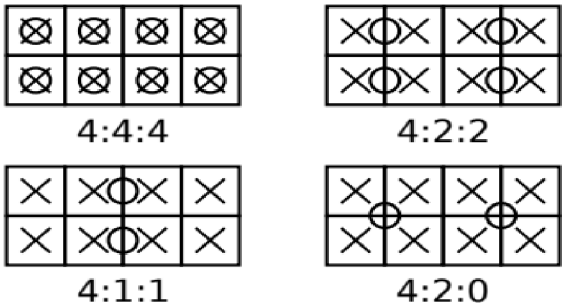
Převod z RGB na Y'C_bC_r:

$$\begin{bmatrix} Y' \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.2}$$

Podvzorkování barvonosných složek

Jak již bylo zmíněno, lidské oko je citlivější na změnu jasu Y více, než na změnu Cb a Cr barvonosných složek. Jelikož tyto barvonosné složky jsou proto méně důležité, můžeme je redukovat.

Tento proces se nazývá podvzorkování. Podvzorkováním dochází k prvnímu ztrátovému převodu, kdy barvonosné složky jsou redukovány na průměrnou hodnotu sousedních pixelů. Ztráta této informace je pro lidské oko nezpozorovatelná. Pixely, které sdílí stejnou barvonosnou složku tvoří blok, jenž má většinou velikost 4×2 pixely. Podvzorkování se udává ve formátu trojice čísel. Ve tvaru 4:X:Y, kde X je počet barvonosné složky v horním řádku a Y je počet barvonosných složek v dolním řádku. Mezi nejpoužívanější patří podvzorkování 4:4:4 (bez podvzorkování), 4:2:0 (barvonosné složky jsou horizontálně i vertikálně podvzorkované na polovinu), 4:2:2 (barvonosné složky jsou horizontálně podvzorkovány na polovinu) a 4:1:1 (barvonosné složky jsou horizontálně podvzorkovány na čtvrtinu).



Obrázek 2.2: Znázornění podvzorkování. Čtverec značí pixel, kolečka jsou barvonosné složky C_b, C_r a křížek znázorňuje jasovou složku Y. [5]

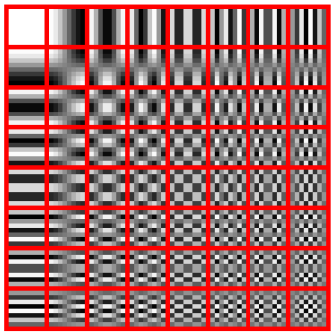
Rozdělení na datové bloky

Samotný obraz je nutné rozdělit na datové bloky, které budou postupně zpracovány ztrátovou kompresí. Toto blokové rozdělení je individuální u každého formátu. Rozdělení na datové bloky probíhá z důvodu výpočetní náročnosti a pro případné využití paralelního zpracování.

Diskrétní kosinová transformace

Diskrétní kosinová transformace (DCT) pomáhá oddělit obraz na části různého významu s ohledem na vizuální kvalitu obrazu. DCT tedy transformuje obraz nebo signál z prostorové domény na frekvenční doménu. Tuto transformaci využívá formát JPEG a Lepton. Transformace je vyjádřena rovnicí:

$$X_k = \left(\frac{1}{2}\right) x_0 + \sum_{i=0}^{N-1} x_n cos\left[\frac{\pi}{N} n \left(k + \frac{1}{2}\right)\right] \tag{2.3}$$



Obrázek 2.3: Bázová funkce DCT.

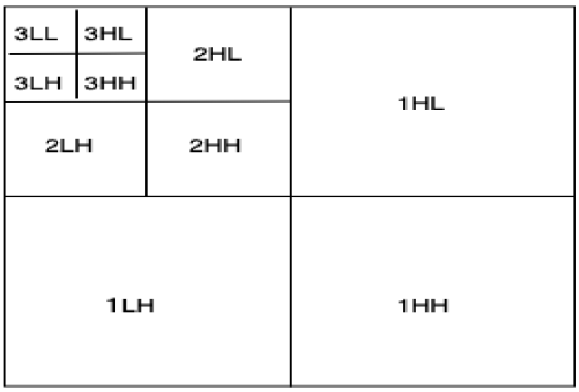
Diskrétní vlnková transformace

Existují dva výpočty diskrétní vlnkové transformace (DWT). Jednorozměrná DWT, která nám rozděluje signál pouze do dvou částí - detaily a aproximaci, zatímco dvourozměrná

vlnková transformace nám umožní rozdělit signál rovnou do čtyř částí, které se nazývají podpásma. Obě transformace využívají transformaci signálu přes dva filtry - dolní propust a horní propust. U dvourozměrné vlknové transformace se provádí opakované filtrování nad vzniklým podpásmem dolní propusti (LL) 2.4. Diskrétní vlnkovou transformaci využívá formát JPEG 2000. Postup výpočtu 1D DWT je definován jako konvoluce následovně:

$$c_{j+1} = (c_j \times h) \downarrow 2 \tag{2.4}$$

$$d_{j+1} = (c_j \times g) \downarrow 2 \tag{2.5}$$



Obrázek 2.4: Dělení na podpásma.

Kvantování

Jedná se ztrátový kompresní proces, jenž je prováděn po transformaci hodnot do frekvenční roviny. Kvantování využívá kvantizační tabulku, tu musí mít pro správnou funkčnost jak dekodér, tak i kodér. Kvantizační tabulka obsahuje koeficienty, které mají být při kódování poděleny s hodnotami zrovna zpracujícího bloku pixelů. Výsledné koeficienty jsou vzápětí zaokrouhleny na celé číslo. Kvantizační tabulka tedy nám určuje k jak velké ztrátě informací má dojít. Kvantizační tabulka může být upravena na základě zadání požadované kvality při kompresi.

RLE

RLE (run-length encoding) je bezeztrátová kompresní metoda, která je velmi jednoduchá. Jejím úkolem je využít posloupnost stejných symbolů a zkrátit je na kratší sekvence. Tyto zkrácené sekvence se vytváří na principu, že posloupnost stejných znaků se zapisuje jako počet opakujících se symbolů a následně se zapisuje daný symbol. Tato metoda existuje v mnoha upravených verzích a využívají je hlavně složité kompresní metody. Problém nastává při dekódování, kdy je nutné rozeznat symbol jako číslo, nebo jako hodnotu udávající posloupnost následujícího symbolu. Aby se předešlo tomuto problému, přidává se do abecedy speciální znak „escape“, který umožní uvozovat krátké sekvence. Pokud jsou použity všechny symboly, je nutné zvolit jednu z nejméně používaných a tu pak následně zapisovat jako samotnou „escape“ sekvenci o jednom znaku.

Entropické kódování

Entropické kódování převádí často vyskytující se symboly na krátké bitové řetězce. Čím větší je četnost daného symbolu, tím menší je bitový řetězec reprezentující daný symbol. Mezi entropické kodéry patří Huffmanovo kódování a také aritmetické kódování.

Huffmanovo kódování

Jedná se o bezeztrátový kompresní algoritmus, používaný hlavně v kompresi JPEG. Algoritmus vytváří binární strom na základě četnosti symbolů, kde listy stromu představují zakódovaný symbol. Při sestavování tohoto stromu je nutné určit četnost jednotlivých symbolů. Na základě této četnosti se seřadí tyto symboly sestupně podle pravděpodobnosti. Po seřazení vybírá a přiřazuje algoritmus 2 symbolům s nejnižší pravděpodobností hodnoty 0 nebo 1 (jako ohodnocení hrany). Následně se tyto uzly sloučí a algoritmus se opakuje. Následně vznikne strom s listy, který nakonec vytvoří výsledný kód pro jednotlivé symboly.

Aritmetické kódování

Aritmetické kódování je také bezeztrátový kompresní algoritmus. Oproti Huffmanovo kódování vytváří optimální kódy. Algoritmus přiděluje jeden kód pro samotný datový soubor v podobě jednoho čísla v intervalu $\langle 0,1)$. Algoritmus zužuje interval v poměru k pravděpodobnosti výskytu symbolu. Zužováním intervalu se zvětšuje kód. Při zužování pravděpodobnějších symbolů vznikne větší interval než u méně pravděpodobnějších, proto dochází ke kompresi.

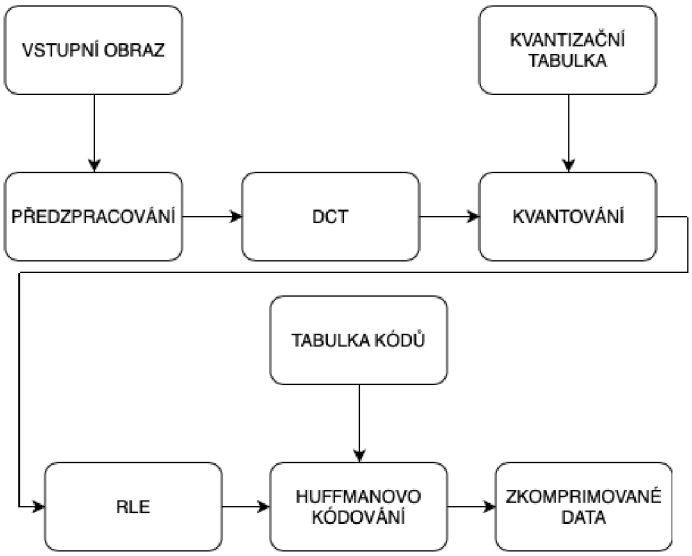
Profily a úrovně

Profily a úrovně se vyskytují snad ve všech standardech. Profily definují např. typ podvzorkování, bitovou hloubku obrazu, maximálně povolených počtů řezů, maximální počet povolených dlaždic a maximální bitový tok. Existují zpravidla hlavní profily (Main) a rozšiřující se profily (např. pro účely statických obrazů).
V dalších nadcházejících sekcích jsou popsané jednotlivé procesy komprese u vybraných ztrátových formátů.

2.1 JPEG

JPEG je standard ztrátové komprese obrazu a také se pod touto zkratkou označuje formát souboru. JPEG je zkratkou konsorcia Joint Photographic Expert Group, která právě stála za vznikem standardu JPEG. Jedná se o nejpoužívanější standard ztrátové komprese obrázků na webu, obzvláště fotografií, při kterých je komprese dostatečně vysoká. JPEG je převážně určený pro ztrátovou kompresi, ale umí také bezeztrátovou kompresi. JPEG není vhodný pro obrázky s kresbou, ikony a obecně grafiku, jelikož standard funguje na principu podobnosti barev v sousedních pixelech. Použitím vstupního obrazu s ostrými přechody barev nebo vektorové grafiky můžou následně vznikat různé typy rušivých artefaktů. JPEG umí ukládat EXIF informace a také umí spravovat barvy pomocí profilů ICC. V této sekci jsem čerpal z těchto zdrojů [11], [6] a [28].

Obrázek 2.5 znázorňuje proces komprese JPEG. Prvním procesem JPEG komprese je převod obrazu na správný barevný model. Nejprve se tedy převádí obraz z původního



Obrázek 2.5: Diagram JPEG komprese.

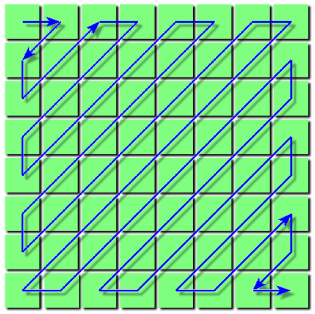
barevného modelu na $Y'C_bC_r$. Následně můžou být jednotlivé složky podvzorkovány. JPEG komprese nejčastěji používá model 4:2:0, ale také i 4:2:2 nebo 4:1:1.

Dalším procesem komprese je obraz rozdělen na bloky o velikosti 8×8 jasové složky a na bloky o velikosti 8×8 barevné složky. Existuje-li složka, která nemá rozměr v násobcích 8×8 pixelů, obraz se tak rozšíří na danou velikost pomocí duplikace sloupce nebo řádku, který je nejbližší.

Následujícím procesem komprese je transformace koeficientů. Nad jednotlivými zmíněnými bloky o velikosti 8×8 pixelů se provede transformace DCT (diskrétní kosinová transformace). Po provedení transformace DCT získáme obraz ve frekvenční oblasti, obrázek 2.3, v podobě matice se 64 koefienty (1 DC koeficient a 63 AC koeficienty).

Po provedení transformace DCT následuje krok kvantování koeficientů. V tomto kroku dochází právě ke ztrátě informací. Je to proces, kdy DCT koeficienty jsou děleny kvantizační tabulkou, která určuje k jak velké kompresi má dojít. Tuto tabulku musí mít kodér i dekodér.

Finálním procesem komprese je entropické kódování. Koeficienty jsou postupně čítány metodou zig-zag, zobrazující obrázek 2.6. Data se nejprve bezeztrátově komprimují algoritmem RLE (Runlength encoding) a následně jsou zakódované Huffmanovým kódováním.



Obrázek 2.6: Průchod zig-zag.

Lepton

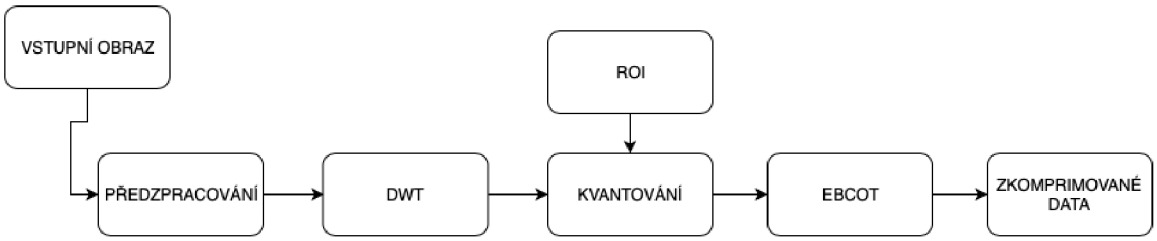
Lepton je bezeztrátová komprese určena výhradně pro obrazy ve formátu JPEG. Tento formát vyvinula firma Dropbox, jejíž cílem bylo zvýšit rychlost komprese a snížit velikost obrazových souborů na svém úložišti. Využívá více vláken pro zpracování komprese a zaměřuje se na přesnější predikci pro každou hodnotu koeficientu, a tak docílí menší velikosti souboru. V této sekci jsem čerpal z tohoto zdroje [16].

Proces komprese Leptonu není příliš odlišný v prvních krocích oproti JPEG komprese. Převod obrazu do barevného modelu $Y'C_bC_r$, následné rozdělení obrazu na bloky 8×8 a nakonec provedení nad těmito bloky DCT (diskrétní kosinusová transformace) transformaci.

Proces komprese Leptonu oproti JPEG kompresi se liší tedy v kódování samotných koeficientů. Většina obrazových formátů dává DC koeficienty před jakýmkoliv AC koeficientem ve formátu souboru. Lepton dostane výhodu komprese tím, že kóduje DC jako poslední hodnotu v každém bloku. Protože DC koeficienty jsou serializovány jako poslední, existuje velké množství informací z AC koeficientů dostupných pro predikci DC koeficientu. Definováním dobré a reprodukovatelné predikce můžeme odečíst skutečný DC koeficient od predikovaného DC koeficientu a pouze zakódovat deltu. V budoucnu pak můžeme využít predikci spolu s uloženou deltou k získání původního DC koeficientu. Téměř ve všech případech má tato technika za následek podstatně nižší počet symbolů, které se dostanou do našeho aritmetického kodéru. Lepton pro kódování využívá navíc VP8 aritmetický kodér, jenž je mnohem účinnější.

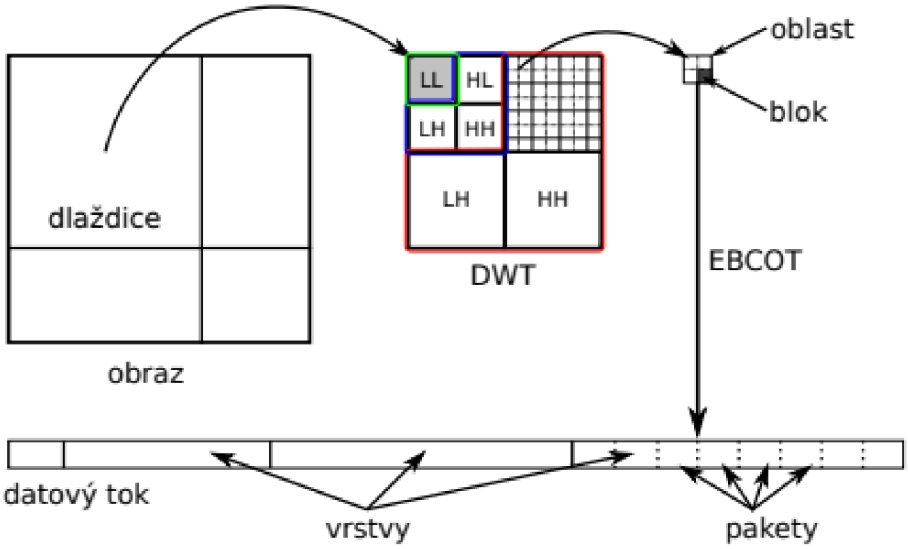
2.2 JPEG2000

JPEG 2000 je nástupce starého standardu JPEG. Zkratka formátu znamená opět Joint Photographic Expert Group, jedná se o skupinu, která vyvinula tento formát v roce 2000. Ten umí ztrátovou i bezeztrátovou kompresi obrazu. Formát dokáže pracovat s transparentním pozadím, má vyšší kompresní poměr a je odolnější vůči vzniku chyb. Hlavní změnou, co se týká procesu komprese oproti staršímu JPEG formátu, je použití DWT (diskrétní vlnkové transformace). JPEG 2000 umí pracovat s jakoukoliv bitovou hloubkou obrazu. Komprese rozděluje obraz na dlaždice o stejné velikosti. Každá z dlaždic je komprimována zvlášť. V komprimovaných obrazech JPEG 2000 se při vyšším kompresním poměru a nižším bitrate hodnotě objevují artefakty typu ringing a blurring. V této sekci jsem čerpal z těchto zdrojů [25] a [6].



Obrázek 2.7: Diagram JPEG 2000 komprese.

Proces komprese JPEG 2000 je znázorněn na obrázku 2.7. Prvním procesem JPEG 2000 komprese je převedení obrazu do barevného modelu $Y'C_bC_r$, tedy stejně jako JPEG standard. Po převedení do barevného modelu $Y'C_bC_r$ může dojít k podvzorkování barvonosných složek.



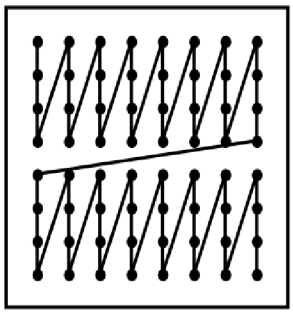
Obrázek 2.8: Rozsekání obrazu na dlaždice. Následné provedení DWT transformace nad každou dlaždicí. Rozdělení podpásma na bloky a oblasti. Kódování bloků algoritmem EBCOT. Uspořádání komprimovaného toku do vrstev. [6]

Dalším procesem komprese je po převodu barev každá barevná složka rozsekána na obdélníkové dlaždice stejných rozměrů, nad kterými se provádí poté transformace. Rozdělení obrazu lze vidět v obrázku 2.8. Velikost těchto dlaždic je volitelná, případně může být jediná nad celým obrazem. Tyto dlaždice jsou poté zpracovávány nezávisle, tím dochází ke snížení výpočetní náročnosti. Použitím příliš mnoho dlaždic začínají vznikat artefakty typu blocking. Nad vzniklými dlaždicemi se pak spočítá transformace DWT (diskrétní vlnková transformace). Konkrétně tedy se využívá DWT CDF 9/7, která se používá pro ztrátovou kompresi a vytváří kvantizační šum v závislosti na přesnosti daného dekodéru. Díky DWT dostaneme zdrojový obraz v několika úrovních rozkladu. Obraz je tedy rozdělen na 4 podpásma (LL,LH,HL,HH), kde pásmo LL představuje nejvyšší úroveň rozkladu. Úrovně rozkladu představují rozlišení, které se každou nadcházející úrovní nabývají poloviční rozměrem. Jednotlivá podpásma lze vidět na obrázku 2.4. Obraz je dále rozkládán do dalších rozlišení z aktuální úrovně LL podpásma 2.8. Podpásma se dále rozdělují na bloky kódování a na oblasti.

Po transformaci koeficientů se provede proces kvantování, při němž se redukují přesnosti koeficientů. Při ztrátové kompresi se využívá kvantovací krok, který je spočítán pro každé podpásmo zvlášť, nebo se odvíjí od nejvyššího podpásma LL. Hodnoty pro výpočet kvantovacího kroku udává sám kodér.

JPEG 2000 dále umí vytvořit oblast zájmu ROI (region of interest). Jedná se o část obrazu, která je důležitá a proto je komprimována přednostně. Tedy kompresor může například označit jako ROI nalezené obličeje ve fotografii a pak při dekompresi jsou nejprve dostupné k dekódování právě tyto tváře a pozadí je jenom v náhledu. Pozadí je dekódováním upřesněno později. Při případném přerušení datového toku jsou oblasti zájmu ROI dekomprimované ve vyšší kvalitě. Pro označování oblasti zájmů ROI využívá JPEG 2000 algoritmus Maxshift. Kodér vytvoří ROI masku (bitovou mapu), která určuje jaké koeficienty mají být kódované s lepší kvalitou. Následně Maxshift funguje na principu, že posouvá

a umisťuje bity z ROI masky na vyšší bitovou rovinu, než jsou bity koeficientů mimo ROI mapu.

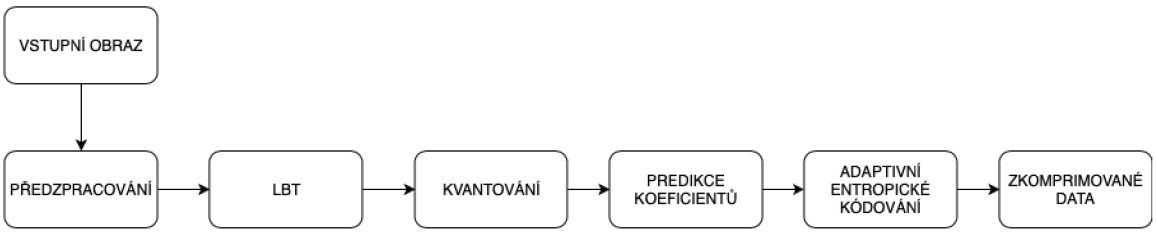


Obrázek 2.9: Pořadí použité pro kódování v EBCOT algoritmu ze standardu JPEG 2000. [5]

Finálním procesem komprese je entropické kódování pomocí metody EBCOT. Samotné kódování probíhá od nejvíce významného bitu po nejméně významný bit 2.9. Zpracovaná data jsou poté uložena ve vrstvách jako datový tok, aby se mohla zvyšovat kvalita obrazu s každou následující dekódovanou vrstvou.

2.3 JPEG XR

JPEG XR standard je zkratkou Joint Photographic Experts Group - Extended Range, taky známý jako Windows Media Photo nebo HD Photo. Tento formát vytvořila a vyvíjí od roku 2006 firma Microsoft. Zaměřuje se především na zobrazování statických obrazů s nepřetržitým tónem, jako jsou například fotografické obrazy. Má vysokou kompresní schopnost a nízké požadavky na výpočetní a paměťové prostředky. Podporuje jak ztrátovou, tak i bezeztrátovou kompresi obrazu. JPEG XR je specifický tím, že využívá k transformaci LBT (lapped biorthognal transform). V této sekci jsem čerpal z těchto zdrojů [12],[23], [18] a [17].

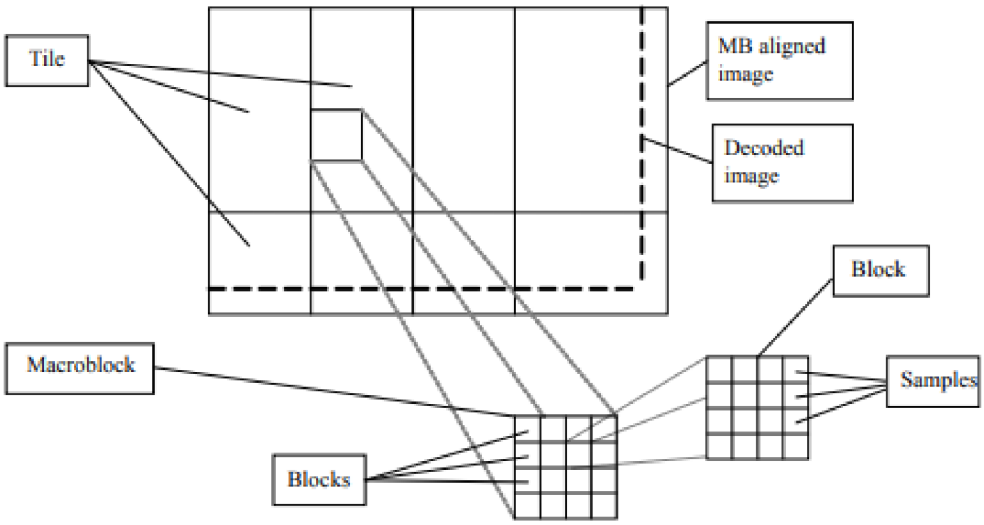


Obrázek 2.10: Diagram JPEG XR komprese.

Obrázek 2.10 znázorňuje proces komprese JPEG XR. Algoritmus komprese formátu JPEG XR nejprve provádí transformaci barev z barvového modelu RGB, či jiného modelu, do barvového modelu YUV. Následně se při potřebě provádí podvzorkování barvonosných složek.

Zarovnání a odsazení obrazu do makrobloků je dalším krokem komprese znázorněné v obrázku 2.11. Toto zarovnání se hlavně provádí, pokud šířka a výška obrazu není dělitelná 16. Poté se jinak provádí rozšíření na velikost dělitelnou právě šestnácti. Tohle rozšíření se

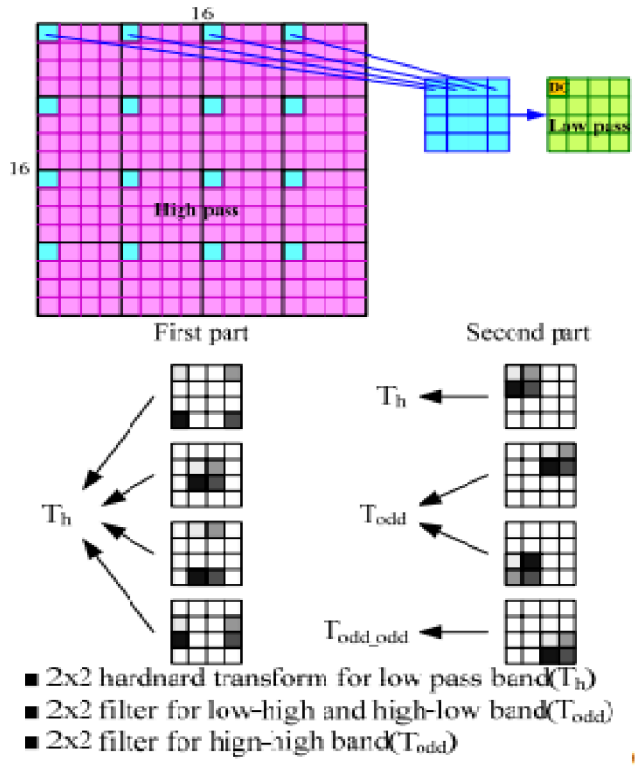
většinou provádí duplikací posledního řádku a posledního sloupce. Obraz se následně rozdělí na dlaždice o dané velikosti. Každá dlaždice obsahuje makrobloky. Samotný makroblok se skládá z 4×4 bloků a každý jeden blok obsahuje 4×4 pixelů.



Obrázek 2.11: Obrázek ukazuje zarovnání obrazu a rozdělení na makrobloky, které se dělí dále na bloky. [18]

Dalším procesem komprese je transformace LBT (lapped biorthognal transform), díky níž se převedou pixely z prostorové domény do frekvenční domény. LBT funguje na principu dvou etap. Každá etapa provádí tyto dvě transformace: FCT (Forward Core Transform) a OPF (Overlap Pre Filter). JPEG XR označuje transformaci FCT jako PCT (Photo Core Transform) a OPF jako POT (Picture Overlap Transform).

Pokud je vyžadováno, tak v každé etapě nejprve dojde k provedení předfiltrovácí transformací POT. Tato transformace je prováděna kvůli redundanci artefaktů v okolí hran bloků, jelikož frekvenční transformace nelze využít pro vyřešení tohoto problému. POT ve srovnání s PCT, který se provádí pouze uvnitř každého bloku 4×4 pixelů, je složitější na paměť, protože se provádí nejen uvnitř každého makrobloku, ale i mezi hranicemi přilehlých makrobloků. POT se skládá ze dvou hlavních operací: 4×4 pixelů uprostřed všech přilehlých 4 makrobloků a z oblastí hranic mezi markobloky o velikosti 2×4, 4×2 pixelů. Photo Core Transform (PCT) je druhotnou transformací v každé etapě. Transformuje pixel po přefiltrování na nejnižší frekvenci koeficient DC, nízkofrekvenční koeficienty AD a vysoké frekvenční koeficienty AC. Existují dvě části procesu PCT, tyto procesy lze vidět v obrázku 2.12. Makroblok je rozdělen do šestnácti 4×4 bloků. V každém dílčím procesu je každý blok 4×4 přefiltrován a poté transformován pomocí 4×4 PCT. V první etapě po provedení FCT nám vznike 4×4 bloků s 16 DC koeficienty (nesoucí informace o barvě), které označujeme jako vysoko frekvenční koeficienty. Vzniklých 16 DC koeficientů následně seskládáme jako DC blok koeficientů o velikosti 4×4, označován taky jako blok nízkofrekvenčních koeficientů, nad nímž se provede druhá etapa transformace. Druhá etapa transformace tedy provede POT transformaci nad blokem 4×4 DC koeficientů a následně PCT transformace. Po provedení obou etap transformací LBT získáme transformované makrobloky ve frekvenční



Obrázek 2.12: PCT transformace použité v první a v druhé etapě. Frekvenční hierarchie makrobloku. [18]

doméně. Tyto makrobloky se skládají z 15 koeficientů dolní propusti, 240 koeficientů horní propusti a 1 DC koeficientu.

Kvantování je proces přeškálování koeficientů po provedení transformace. Kvantizační proces používá kvantovanou hodnotu k dělení a zaokrouhlení koeficientů. V režimu ztrátového kódování je tato hodnota je větší než 1. Ke kvantování JPEG XR komprese se využívá celočíselná operace, která udržuje přesnost po přeškálování hodnot. Parametr pro kvantování DC hodnot, horní a dolní propusti se může lišit, kvůli odlišné citlivosti lidského vidění.

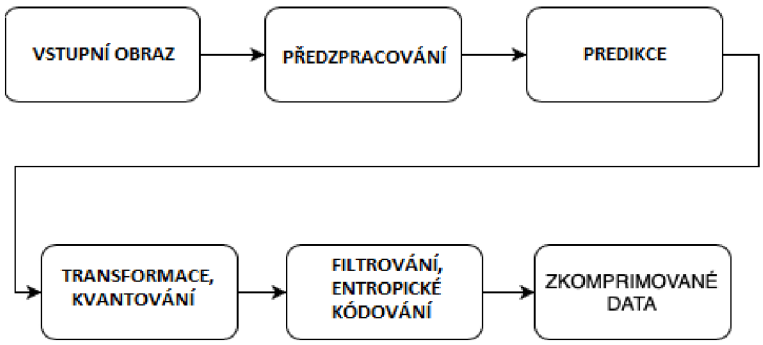
Následujícím procesem komprese je predikce koeficientů. Predikce DC koeficientů v makroblocích u JPEG XR komprese je odlišná oproti jiným obrazovým standardům. Pro predikci DC koeficentů JPEG XR používá predikci zleva, shora, zleva a zároveň shora. Režim predikce je zvolen na základě umístění makrobloku.

Adaptivní entropické kódování je posledním procesem komprese. Adaptivní entropické kódování má za cíl snížit objem dat, jelikož velké množství koeficientů momentálně nabývá hodnot 0 po provedení kvantování. K této redukci se zde využívá kódování RLE a VLC.

2.4 WebP

WebP je obrazový formát a standard, jenž byl vyvinut v roce 2010 společností Google. WebP ztrátový kompresní algoritmus využívá pouze intra snímkového režimu v kompresním formátu videa VP8, který vyvíjí taky společnost Google. WebP formát se snaží uspět

v oblasti ztrátových formátech obrazu jako náhrada formátu JPEG a v dostupnosti na webových službách. V této kapitole jsem čerpal z těchto zdrojů [13], [3], [4].



Obrázek 2.13: Diagram WebP komprese.

Následně bude popsán postup algoritmu VP8, který znázorňuje diagram komprese 2.13. Komprese VP8 nejprve převede vstupní obraz na barevný model YUV420. Po převedení barev se rozdělí vstupní obraz na makrobloky, ty jsou čtvercového tvaru. Každý makroblok se skládá z 16×16 jasových bloků Y a dvou 8×8 barvonosných bloků U a V. Tyto bloky se v makroblocích označují dále jako podbloky, které jsou reprezentovány jako 4×4 pixely. Toto rozdělení je prováděno právě z důvodu efektivnějšího a lepšího zpracování.

WebP využívá intra snímkovou predikci VP8, která je zásadním důvodem efektivní komprese. Intra snímek se nejprve rozdělí na bloky o velikostech 4×4, 8×8 nebo 16×16 a následně se na tyto bloky použije jeden z těchto režimů predikce: DC predikce, vertikální predikce, horizontální predikce nebo speciální algoritmus predikce bloků. Každý režim predikce je filtr. Vhodné vybrání filtru funguje na principu nejlepší podoby hodnot po provedení filtru s porovnáním s právě zpracovávaným blokem. Hodnoty po provedení vybraného filtru jsou následně odečteny od originálu a je uložen pouze rozdíl. Rozdíl se nazývá reziduum. Mezi nejpoužívanější predikční metodu patří právě speciální algoritmus predikce bloků (truemotion prediction). Tahle nejpoužívanější a nejefektivnější metoda predikce je znázorněna na obrázku 2.14.

C	A_0	A_1	A_2	A_3
L_0	X_{00}	X_{01}	X_{02}	X_{03}
L_1	X_{10}	X_{11}	X_{12}	X_{13}
L_2	X_{20}	X_{21}	X_{22}	X_{23}
L_3	X_{30}	X_{31}	X_{32}	X_{33}

Obrázek 2.14: Truemotion predikce VP8. [3]

Následně nad vzniklými novými podbloky s rozměry 4×4 pixely se provede transformace koeficientů pomocí DCT. Vznikne nám 16 DC koeficientů.

Po první transformaci následuje druhá transformace využívající DC koeficienty z první transforamce. Tyhle koeficienty jsou dále využity právě k vytvoření virtuálního bloku s roz-

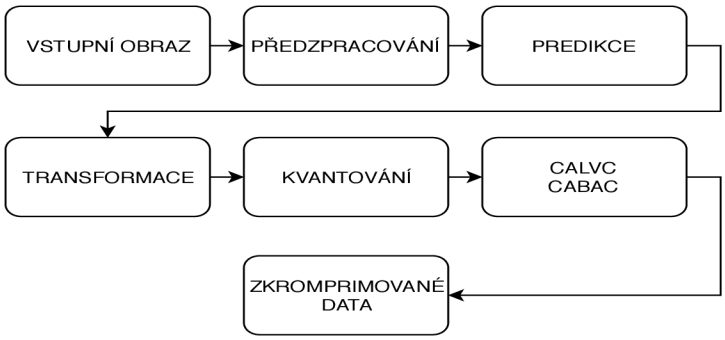
měry 4×4 pixely. Zbylé AC koeficienty jsou normálně kvantovány. Nad novými vzniklými virtuálními bloky je následně provedena transformace WHT, která má za úkol snížit redundanci DC koeficientů. Touto transformací získáme transformované koeficienty o velikosti 4×4, jež se kvantují na základě kvantovací tabulky.

Pro potlačení blokových artefaktů se využívá při procesu komprese loop filter. Tento filter je ve dvou verzích. Jednoduchý a normální filtr. Pokud chceme filtr lze vypnout, docílíme tím rychlejšího dekódování, ale dosáhneme nižší kvality. U filtru je možnost si zvolit, pro jaký typ snímků ho chceme použít, popřípadě si můžeme zvolit v jaké konkrétní oblasti jej použijeme. VP8 umožňuje provedení více úrovní filtrování na dané oblasti, jako je to u oblasti zájmu ROI v JPEG 2000. Pro intra snímky se většinou využívá normální filtr.

Data jsou následně seřazena zig-zag sekvencí a symboly jsou zakódované pomocí Huffmanovo kódování. Pak už se využívá využívá Boolovské entropické kódování. Zakódovaná data jsou ukládaná po oddílech, které lze potom zpracovat paralelně, a tudíž je dekomprese potom efektivní.

2.5 H.264 intra frame / AVC

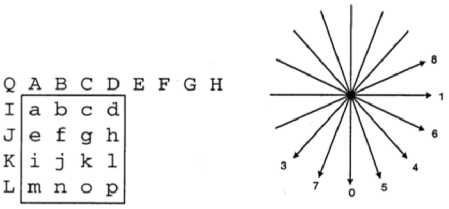
Video kódovací standard H.264 neboli MPEG AVC je nový formát kódování videa. Standard byl vyvinut organizacemi International Organization for Standardization's MPEG group (ISO/IEC/SG29/WG11) a International Telecommunication Union's video coding experts group (VCEG, ITU-T/SG16/Q.6). Hlavním cílem je přenést obraz v co nejvyšší kvalitě při mnohem nižší přenosové rychlosti. Standard vychází z MPEG-2, který má plně nahradit. Jak MPEG-2, tak i H.246 jsou založeny na hybridním kódovaní videa. Hlavní změnou je nová intra snímková predikční technika, která využívá sousedních hodnot pixelů k predikci aktuálně kódovaného bloku namísto jednoduchých předpovědí DC a AC v MPEG-4. Schopnost kódování v rámci rámce H.264 může dosáhnout vyšší efektivity kódování než v předchozích standardech, a dokonce je konkurenceschopná s nejnovějším standardem kódování statických obrázků JPEG 2000. V této kapitole jsem čerpal z těchto zdrojů [29], [19].



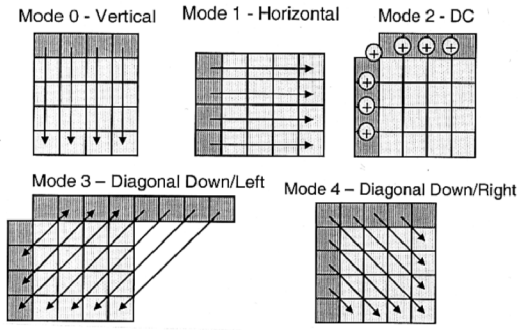
Obrázek 2.15: Diagram H.264 intra kodéru.

Nyní bude popsán proces komprese H.264, který je znázorněný na obrázku 2.15. Prvním krokem procesu komprese je rozdělení obrazu na makrobloky o velikosti 16×16 pixelů.

Následně je každý makroblok převeden do barevného prostoru $Y'C_bC_r$ s podvzorkováním 4:2:0.



Obrázek 2.16: Predikce AVC. [29]



Obrázek 2.17: Módy predikce AVC kodéru. [29]

Druhým krokem procesu komprese je predikce koeficientů, která je znázorněná na obrázku 2.16. Kodér zpracovává samotné makrobloky o velikosti 16×16 pixelů. Všechny barevné a jasové složky v makroblocích jsou predikovány dvěma možnými predikcemi. Predikcí pracující s 4×4 pixelů nebo 16×16 pixelů. Predikce 4×4 se používá na částech snímku, kde jsou významné detaily. Obrázek 2.16 vyobrazuje predikci pro bloky o velikosti 4×4 pixely a následný obrázek 2.16 zobrazuje možné módy predikcí. Naopak zase predikce 16×16 se používá u částí snímků, kde jsou hladké plochy a přechody. Komprese H.264 umí také vytvořit řezy (slices). Řezy jsou skupiny, skládající se z několika makrobloků. Tyto řezy se využívají pro označení důležitých oblastí v obrazu. Intra predikce se proto nekombinuje s ostatními makrobloky jiných řezů, je to proto, aby byly všechny řezy navzájem nezávislé. Predikční matice je následně odečtena od originální matice a vzniklé reziduum se dále zpracovává.

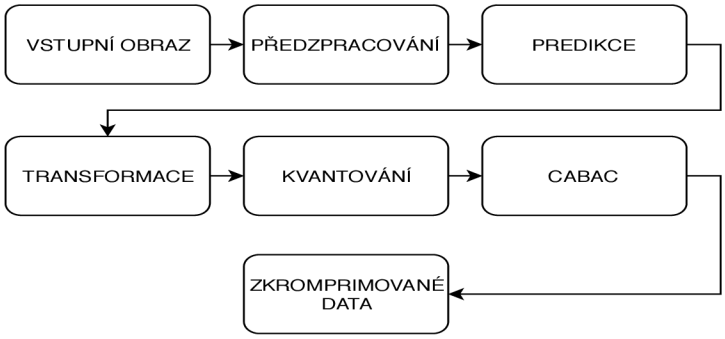
Po predikci koeficientů je dalším krokem procesu komprese transformace. H.264 pro transformaci koeficientů využívá celočíselnou diskrétní kosinovou transformaci (DCT) na blocích o velikosti 4×4, u High profilů může pracovat s velikostí 8×8 bloků. Díky celočíselné transformaci je frekvenční dělení o něco přesnější oproti DCT transformaci.

Následujícím procesem komprese je kvantování transformovaných koeficientů. Kvantizační parametr se používá pro stanovení kvantování transformačních koeficientů v AVC. Parametr může mít až 52 hodnot. Tyto hodnoty jsou uspořádány tak, že zvýšení o 1 hodnotu v kvantovacím parametru značí zvýšení kvantovacího kroku velikosti přibližně o 12 %.

Při konečném procesu H.264 podporuje několik metod entropického kódování. Jako základní kódování se používá CAVLC (Context Adaptive Variable Length Coding), ale pro lepší výsledky se využívá CABAC (Context based Adaptive Binary Arithmetic Coding), který je zase hardwarově náročnější.

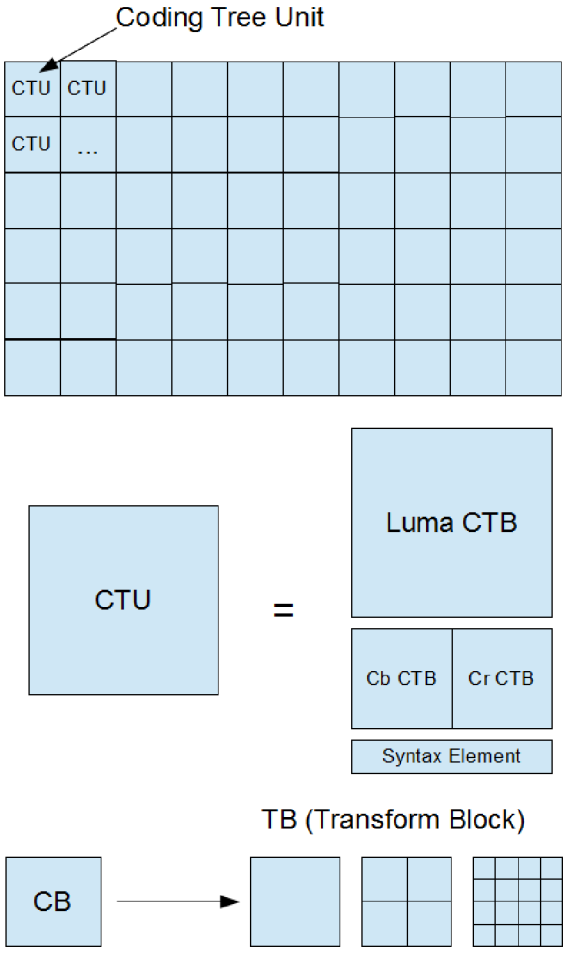
2.6 H.265 intra frame / HEVC

H.265, neboli HEVC (High Efficiency Video Coding), je vylepšená H.264 určená pro širší využití. V porovnáním s AVC se princip HEVC moc neliší. HEVC nabízí dvojnásobný poměr komprese dat při stejné kvalitě videa díky lepším predikcím. Kromě toho podporuje rozlišení až 8k UHD. V této kapitole jsem čerpal z těchto zdrojů [24], [20] a [27].



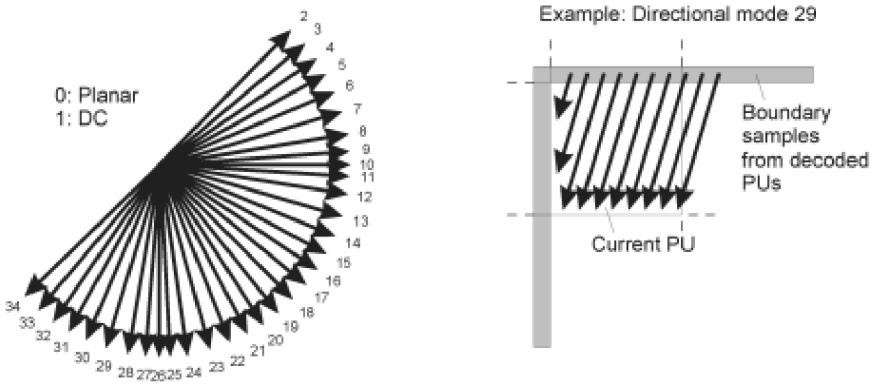
Obrázek 2.18: Diagram HEVC kodéru. [27]

Obrázek 2.18 znázorňuje proces komprese. Prvním krokem procesu HEVC komprese je obraz převeden do barevného prostoru $Y'C_bC_r$ s podvzorkováním 4:2:0.



Obrázek 2.19: Znázornění rozdělení obrazu na CTU (Coding Tree Unit). Rozdělení CU (Coding Unit). Transformace na TB (Transform Block). [22]

Druhým krokem procesu je rozdělení vstupního obrazu na kódovací bloky. HEVC již nepoužívá pevnou velikost makrobloků jako starší standardy MPEG. Namísto makrobloku používá kódovací stromovou jednotku CTU (Coding Tree Unit). CTU může být o rozměrech 16×16 pixelů, 32×32 pixelů, nebo dokonce až 64×64 pixelů. CTU se skládá z jasové složky CTB (Coding Tree Block) a 2 barvonosných složek CTB. CTB mají stejný rozměr jako zvolený rozměr CTU. CTB bloky můžou být značně velké na predikci. Proto každá složka CTB může být individuálně rozdělena na menší bloky, tzv. kódovací bloky CB (Coding Block). CB mohou nabývat rozměrů od 8×8 až po 64×64 bloků. Jelikož každá složka má vlastní kódovací bloky CB, poskládáním všech 3 složek CB vznikne CU (Coding Unit) 2.19, která obsahuje oddělené hodnoty barev. Pro samotnou predikci je CB stále velkým blokem, proto se CB dále dělí na různé bloky transformačního bloku TB (Transform Block). Rozdělení kódovacích bloků CB na transformační bloky TB pro intra predikci je znázorněna v obrázku 2.19.



Obrázek 2.20: Intra predikce HEVC. [27]

Dalším krokem procesu komprese je predikce bloků. To znamená, že kodér ukládá na základě predikční metody pouze rozdíly s ohledem na další bloky jako odkaz. Pro obrazové formáty tedy HEVC používá intra predikci, která funguje na principu jako H.264 a to na základě sousedních bloků. HEVC intra predikce podporuje 33 směrových režimů, 1 rovinnou predikci a několik DC predikčních režimů. Právě směrová predikce s 33 různými směry orientace, která je znázorněna v obrázku 2.20, je definovaná pro predikční bloky PU. Predikční matice je následně odečtena od originální matice a vzniklé reziduum se dále zpracovává v dalším procesu komprese.

Následujícím krokem procesu komprese je transformace. Po provedení predikce dostáváme nové matice hodnot, které je nutné transformovat do frekvenční domény. Kódovací bloky CB jsou moc velké, proto se využívají transformační bloky TB (Transform Block). Toto rozdělení je znázorněno v obrázku 2.19. Na každý tento blok je následně aplikována celočíselná DCT transformace.

Předposledním procesem komprese po transformaci je kvantování koeficientů. Pro kvantování používá HEVC v podstatě stejné URQ (uniform reconstruction quantization) řízené kvantovacím parametrem (QP) jako v H.264, ale s podporou kvantování škálovacích matic pro různé velikosti transformačních bloků TB.

Nakonec posledním krokem procesu komprese je entropické kódování. HEVC používá kontextově adaptivní binární aritmetické kódování (CABAC). To je podobné systému CA-

BAC v H.264, ale prošlo několika vylepšeními, které zlepšily kompresní výkon a snížily požadavky na kontextovou paměť.

HEIF

High Efficiency Image File Format (HEIF, ISO / IEC 23008-12) specifikuje ukládání jednotlivých obrazů, obrazových sekvencí a jejich metadat do kontejnerového souboru, který odpovídá formátu ISO Base Media File Format (ISO / IEC 14496-12). HEIF je tedy strukturální formát kontejnerového souboru, z něhož lze odvodit kodekově specifické obrazové formáty. Standard HEIF samozřejmě obsahuje specifikaci pro zapouzdření obrazů kódovaných HEVC. Standard vznikl na základě zjištění, že HEVC má dobrý kompresní výkon i se statistickými obrazy a může být využit v digitálních kamerách. HEIF však není výhradně omezen pouze na ukládání obrazů kódovaných HEVC. Hlavní součástí formátu je popis struktury souboru na high-levelu, který rozlišuje rozdíl mezi obrazem a ukládáním obrazových sekvencí. Fotografie se ukládají jako položky. Všechny obrazové položky jsou nezávisle kódovány a nezávisí na žádné jiné položce v jejich dekódování. Do stejného souboru lze zahrnout libovolný počet položek obrázků. Obrazové sekvence jsou uloženy jako stopy a místo obrazových položek lze použít stopu sledu snímků, pokud je mezi obrázky závislost kódování. V této kapitole jsem čerpal z těchto zdrojů [15] a [14] a [26].

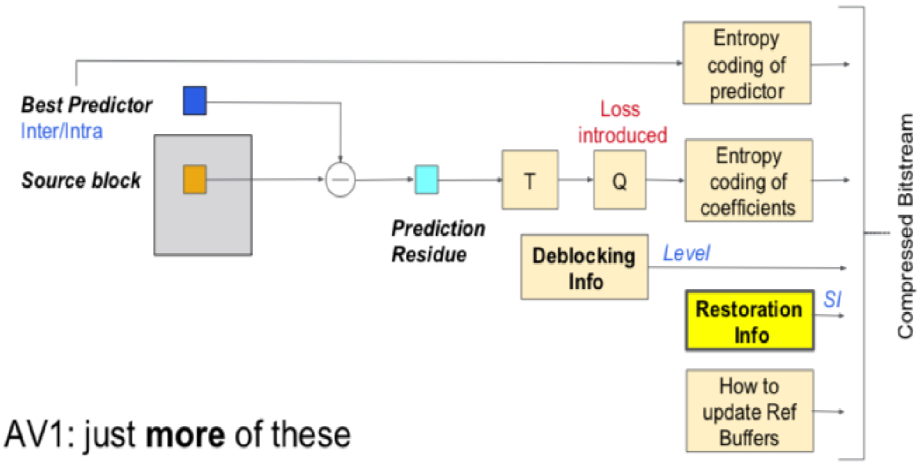
BPG

BPG (Better Portable Graphics) je nový obrazový formát, který je v podstatě profil HEVC Main 4:4:4 16 Still Picture, jenž může mít až 14 bitů na vzorek. BPG byl vytvořen programátorem Fabrice Bellardem v roce 2014. BPG podporuje barevné formáty se vzorkováním 4:2:2, 4:4:4 a 4:2:0. Podporovány jsou různé barevné formáty, včetně podpory stupňů šedi, RGB, $Y'C_gC_o$, $Y'C_bC_r$. Také podporuje animace a používá řadu meta dat pro efektivní konverzi včetně EXIF, ICC profilu a XMP. BPG má za úkol nahradit zastaralý ztrátový formát JPEG. Na rozdíl od JPEG, BPG nevyžaduje doplňkové pluginy prohlížeče pro zobrazení komprimovaného obrazu. BPG je podporován většinou webových prohlížečů s Javascriptovým dekodérem. BPG ukládá metadata na začátek souboru a jejich délka je definovaná v hlavičce obrazu, takže dekodér je dokáže lehce přeskočit. V této kapitole jsem čerpal z těchto zdrojů [2] a [8].

2.7 AV1 intra

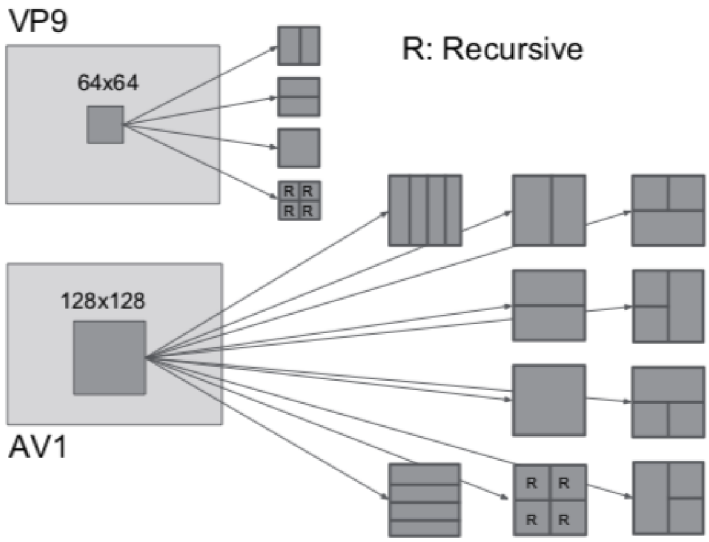
AOMedia Video 1 (AV1) je otevřený a bezplatný video kodek inspirovaný kodekem VP9 od Googlu. Vytvořila ho v roce 2015 Alliance for Open Media (AOMedia). AOMedia je konsorcium několik desítek technologických firem, jež se snažili vyvinout bezplatný video kodek, který je škálovatelný, s nízkou výpočetní stopou, optimalizovaný pro web a hardware, flexibilní pro komerční i nekomerční obsah. Namísto rozmazání a zmenšení detailů kodek skrývá kvantizační artefakty. Dále AV1 předpovídá koeficient barvonosné složky z koeficientů jasové složky. V této kapitole jsem čerpal z těchto zdrojů [10], [21] a [1].

Obrázek 2.21 znázorňuje proces komprese. Prvním krokem procesu AVC komprese se obraz rozdělí do bloků o stejné velikosti. Bloky jsou označované jako superbloky. Superbloky mají čtvercový tvar a mohou mít rozměr až 128×128. Superbloky lze rozdělit do menších bloků tzv. transformačních bloků podle různých vzorů dělení. Na obrázku 2.22 můžete vidět všech 10 vzorů dělení. Obdélníkové bloky nelze dále dělit na menší bloky, jako to lze



Obrázek 2.21: AV1 diagram. [1]

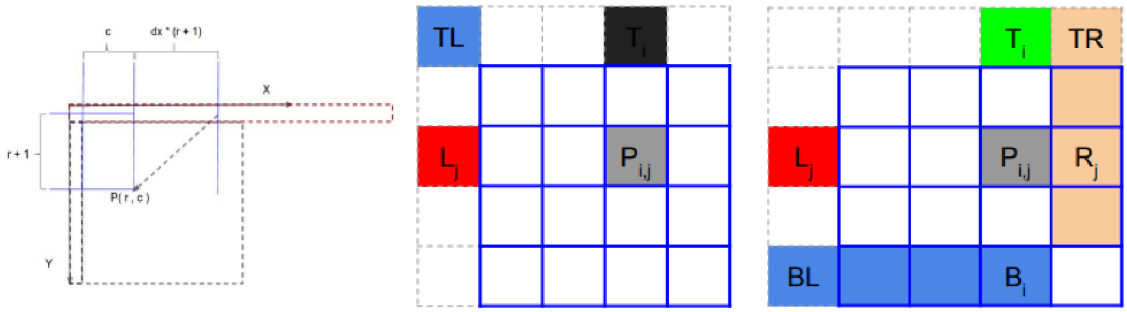
u čtvercových bloků. Druhým krokem procesu komprese je převod barev obraz do barevného modelu YUV.



Obrázek 2.22: Dělení bloků AV1 v porovnání s VP9. [10]

Dalším krokem procesu komprese je predikce. Konkrétně tedy intra predikce, která využívá směrové prediktory, jež extrapolují sousední pixely podle zadaného úhlu. V AV1 lze zvolit 1 z 8 hlavních směrových režimů, nebo 1 ze 3 nesměrových prediktorů: DC, Paeth prediktor a Smooth prediktor. Predikce jsou znázorněny v obrázku 2.23. Hlavní směrové režimy začínají v úhlu 45 stupňů a zvyšují se až do 203 stupňů. Oproti H.265 má AV1 3 nové hladké prediktory SMOOTH_V, SMOOTH_H a SMOOTH, které předpovídají blok pomocí kvadratické interpolace ve svislých nebo vodorovných směrech, nebo jejich průměr po aproximaci pravého a spodního okraje jako pixelu vpravo nahoře a dolního pixelu na levém okraji. Paeth prediktor pro každý pixel kopírujeme jeden z horního okraje sousedního bloku, jeden z levého okraje sousedního bloku a jednoho krajního okraje nahoře vlevo

sousedních bloků zleva a shora, který má hodnotu nejblíže k hodnotě: horní pixel + levý pixel - horní vlevo pixel. Další predikcí, jež zvládá AV1 komprese, je predikce barvonosných složek na základě jasové složky (CfL). Je to prediktor, který modeluje barvonosné pixely jako lineární funkci souběžných rekonstruovaných jasových pixelů.



Obrázek 2.23: Ukázka predikce. Zleva směrová predikce, Paeth prediktor, Smooth prediktor. [1]

Následujícím procesem komprese je transformace koeficientů. AV1 podporuje mnoho transformací: DCT, ADST, inverzní ADST a Identity. Tyto transformace jsou aplikovány na transformační bloky o velikosti až 64×64 pixelů.

Po transformaci koeficientů následuje další proces komprese a to kvantování. AV1 využívá pro kvantování kvantovací matici. AV1 zahrnuje jak lineární, tak nelineární kvantovací matice pro kvantování.

Poslední procesem komprese je entropické kódování. AV1 přechází k použití adaptivního multi-symbolického aritmetického kodéru symbol-to-symbol. Tímto způsobem je multi-symbolová abeceda zakódována až s 15bitovou pravděpodobností a abecední velikostí až 16 symbolů. S tímto entropickým kodérem se kombinuje více binárních symbolů do nebinárních symbolů. Účinnost je zvýšena zejména pro nižší přenosové rychlosti.

AV1 přechází k použití adaptivního multi-symbolického aritmetického kodéru symbol-to-symbol. Každý prvek syntaxe v AV1 je členem specifické abecedy N elementů a kontext se skládá ze sady N pravděpodobností spolu s počtem pro usnadnění rychlého brzkého přizpůsobení. Pravděpodobnosti jsou uloženy jako 15 bitové kumulativní distribuční funkce (CDF). Vyšší přesnost než binární aritmetický kodér umožňuje přesné sledování pravděpodobností méně běžných prvků abecedy. Pravděpodobnosti jsou přizpůsobeny jednoduchým rekurzivním škálováním s aktualizačním faktorem založeným na velikosti abecedy.

AVIF

AVIF je kompresní formát obrazu, který funguje na AV1 intra-frame kódovácím kodéru, jež byl vytvořen skupinou AOMedia (Alliance for Open Media). Formát má být konkurence obrazového formátu HEIF. Podporuje široký barevný gamut (WCG) a obrazy s vysokým dynamickým rozsahem (HDR). Kromě toho podporuje také alfa kanál, animace.

Kapitola 3

Srovnání formátů

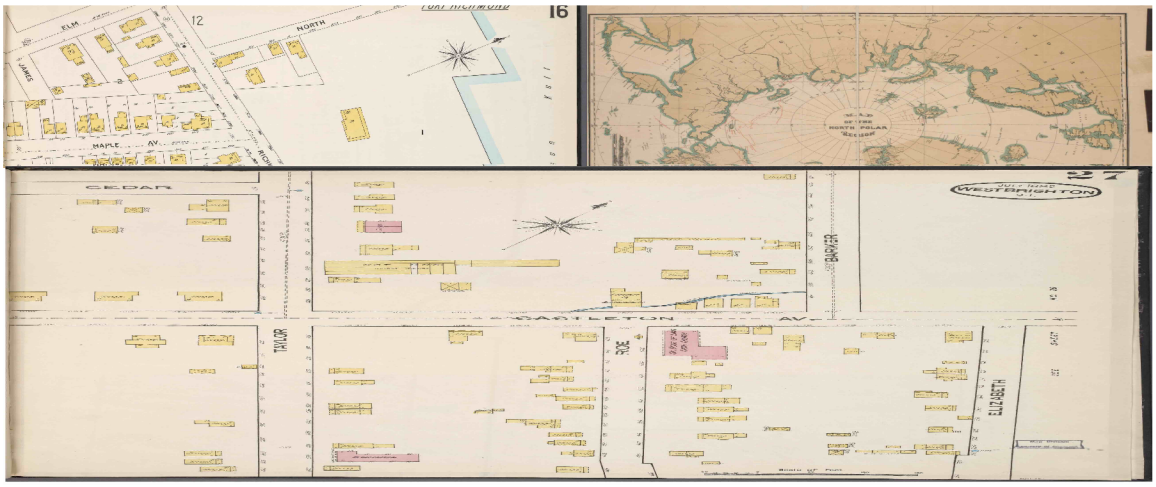
3.1 Datasety

Pro srovnání formátů s cílem zjistit výsledky, které budou objektivní, jsem vytvořil 3 různé typy datasetů, sestavující se z 50 fotografií s vysokým rozlišením 3.3, 20 obrázků skenovaných dokumentů 3.1 a 3 obrázky map 3.2. Dataset fotek s vysokým rozlišením obsahují záměrně obrázky s ostrými hranami, obličeji, a pestrými obrazy, aby testování na tomto datasetu bylo objektivní. Dataset map byl omezen pouze na 3 obrázky, s ohledem na čas komprese a nutnému strojovému výkonu.

Obrázek 3.1: Ukázka datasetu skenovaných dokumentů.

3.2 Metody vyhodnocení kvality komprese

Ztrátová komprese snižuje kvalitu na základě komprimace dat, od toho jsou odvozené ztrátové formáty, které už nelze zrekonstruovat, a je nutné posuzovat kvalitu výsledného obrazu. Obecně lze vyhodnocení provést dvěma způsoby. Subjektivně nebo objektivně. Při subjektivním pozorování je třeba pozorovatele, který nemá informace o referenčním obrazu. Následně pozorovatel určí obraz, jenž se mu líbí a ohodnotí ho. Objektivní vyhodnocení pracuje s matematickými vzorci a vyhodnocuje jej počítač. Objektivní vyhodnocení je tudíž spo-



Obrázek 3.2: Ukázka datasetu map.

Obrázek 3.3: Ukázka datasetu fotek s vysokým rozlišením.

lehlivější, ale hlavně přesnější. Proto při vyhodnocování jsou využity následující objektivní měřící metriky.

MSE

MSE (mean squared error) je střední kvadratická chyba. Výpočty přesnosti odhadů se provádí pomocí střední hodnoty druhých mocnin a rozdílem mezi odhadem a skutečností. Výhodou této metriky je, že je velmi jednoduchá a nenáročná na výpočty a paměť. MSE se spočítá na základě tohoto matematické vzorce 3.1. Při použití této metriky je nutné brát v potaz, že tato matrika vyžaduje originální a komprimovaný obraz se stejnými rozměry. Výsledné hodnoty MSE se pohybují od 0 do 1, přičemž MSE s hodnotou 0 vyjadřuje totožné obrazy. Pokud chceme vypočítat MSE pro RGB obraz, je nutné spočítat MSE pro každý barevný kanál zvlášť a následně sečíst a podělit 3.

$$MSE = \frac{1}{mn} \sum_{m-1}^{i=0} \sum_{n-1}^{j=0} [I(i,j) - K(i,j)]^2 \tag{3.1}$$

PSNR

PSNR (peak signal-to-noise ratio) určuje nám poměr mezi maximálně možným výkonem signálu a výkonem rušivého šumu, který ovlivňuje věrnost jeho reprezentace. Jelikož mnoho signálů má velmi široký dynamický rozsah, PSNR je obvykle vyjádřen v decibelech. Vyšší hodnota PSNR v decibelech určuje kvalitnější obraz. Hodnota PSNR nad 40 decibelů značí kvalitní obraz, naopak pokud obrazy nejsou shodné v žádném případě, PSNR hodnota je nedefinovaná, jelikož PSNR využívá metriku MSE, ta je v tomto případě nulová. Matematický výraz pro výpočet PSNR hodnoty lze vidět 3.2.

$$PSNR = 20 \times \log_{10}(MAX_I) - 10 \times (MSE) \tag{3.2}$$

$\mathbf{PSNR_{YUV}}$

PSNR_{YUV} funguje podobně jako normální PSNR zmíněné výše. $PSNR_{YUV}$ počítá zvlášť PSNR nad každým kanálem Y, C_b, C_r. Pro vypočtení PSNR pro každý kanál se provede výsledné vážení, které je vyjádřeno touto rovnicí 3.3.

$$PSNR_{YUV} = \frac{(6 \times PSNR_Y + PSNR_U + PSNR_V)}{2} \tag{3.3}$$

SSIM

Index strukturální podobnosti (SSIM) je metrika vnímání, jež se snaží indexovat kvalitu obrazů jako lidský pozorovatel a poohlíží na strukturu objektů. Kvantifikuje degradaci kvality obrazu způsobenou kompresí nebo ztrátou informací při přenosu dat. Jedná se tedy o referenční metriku, která potřebuje referenční a zpracovaný obraz (komprimovaný), jež jsou ze stejného obrazu. SSIM index se udává v rozmezí -1 až 1, kde index 0 udává shodný obraz. SSIM u barevných obrazů počítá pouze s jasovou složkou. SSIM algoritmus pracuje nad dvěma signály a měří zvlášť 3 části: jas (l), struktura (s) a kontrast (c) 3.4. Následně kombinací všech 3 složek získáme výsledný SSIM vzorec, který nám vrátí SSIM index obrazu. Zde jsem čerpal ze zdroje [9].

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2\mu_y^2 + c_1} \tag{3.4}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2\sigma_y^2 + c_2} \tag{3.5}$$

$$s(x,y) = \frac{\sigma_x y + c_3}{\sigma_x\sigma_y + c_3} \tag{3.6}$$

$$\tag{3.7}$$

$$SSIM(x,y) = [l(x,y)^\alpha \times c(x,y)^\beta \times s(x,y)^\gamma] \tag{3.8}$$

3.3 Srovnání vlastností ztrátových formátů

Tato sekce se zabývá teoretickým srovnáním vlastností jednotlivých ztrátových formátů, které byly rozebrány v kapitole 2. Dále se zaobírá popisem vznikajících artefaktů u některých formátů a způsobu zachování při korupci dat.

	JPEG	Lepton	JPEG-2000	JPEG-XR	WEBP	AVC	HEIF	BPG	AVIF
Transf.	DCT	DCT	DWT	LBT	WHT a DCT	DCT a IDCT	WHT	WHT	DCT, ADST, FADST
Rozděl. obrazu	MB 8×8	MB 8×8	dlažd.	dlažd.	MB 16×16	MB 16×16	CTU 64×64	CTU 64×64	SB 128×128
Ztrátový	ANO	NE	ANO	ANO	ANO	ANO	ANO	ANO	ANO
Beze-ztrátový	NE	ANO	ANO	ANO	ANO	ANO	ANO	ANO	ANO
Animace	NE	NE	NE	NE	ANO	ANO	ANO	ANO	ANO
HDR	NE	NE	ANO	ANO	NE	NE	ANO	ANO	ANO
Alfa	NE	NE	ANO	ANO	ANO	NE	ANO	ANO	ANO
Progres. přenos	ANO	ANO	ANO	ANO	NE	ANO	ANO	ANO	NE

Tabulka 3.1: Tabulka srovnání všech formátů. HDR (High Dynamic Range) - vysoce dynamický rozsah zobrazení. Alfa - podpora alfa kanálu.

Artefakty

Artefakty v obraze představují zkreslení jisté části obrazového obsahu. Vznikají zejména při ztrátové kompresi obrazu, při nedostatku dat k správné rekonstrukci obrazu. Pro zobrazení artefaktů jsem vybral rastrový obraz, který má ostré hrany a kontrastní rozdíl. Tedy pro testování artefaktů byl zvolen obrázek 3.4.

Typickým artefaktem pro JPEG formát je blocking. Blocking se objevuje při použití vysoké úrovni komprese, zobrazuje se ve formě různobarevných bloků o velikosti 8×8 pixelů tvořící mapy s viditelnými hranicemi bloků. Blocking artefakt lze vidět na obrázku 3.6. Tyto bloky vznikly po DCT (dikrétní kosinusové transformaci), při které jsou koeficienty kvantizované. Dalším možným JPEG artefaktem je ringing, který se objevuje v obraze jako rušivý signál v okolí ostrých přechodů barev. Tento artefakt je zapříčiněn odstraněním vysokých frekvencí.

U formátu JPEG 2000 se při vysoké kompresi začínají vznikat blurring artefakty. Blurring artefakt se projevuje jako rozmazání ostrých hran v obraze a vytváření prstenců v podobě rozamazných vlnek kolem těchto ostrých hran. Tento jev lze vidět na obrázku 3.7.

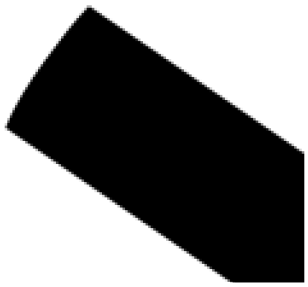
JPEG XR formát díky dvou etapám transformace, kde každá etapa může využívat předfiltrování, se snaží bojovat proti artefaktům typu blocking. I přesto můžou vznikat blocking artefakty nejen mezi makrobloky, ale také mezi dlaždicemi. Jemný blocking artefakt lze vidět na obrázku 3.8.

Formát WebP bez využití loop filtru při velké kompresi s nízkým datovým tokem produkuje blurring artefakty. Tento artefakt se projevuje v podobě blocích o velikosti 4×4 pixelů, jak lze vidět na obrázku 3.9.

Formát AVC vykazuje také při vysoké kompresi blocking a blurring artefakty. Blocking artefakty opět vznikají na základě použití DCT transformace koeficientů. Nedosahuje, ale až tak velkého blurring efektu jako JPEG formát. Tyto artefakty lze vidět na obrázku 3.11.

Ostatní nezmíněné formáty v této sekci artefaktů nevykazují značné artefakty, které by bylo vhodné reprezentovat. Pro srovnání pouze uvedu jeden z nejlepších formátů, jenž se nejvíce podobá originálnímu obrazu, a to je AV1 formát. Ten lze vidět na obrázku 3.11.

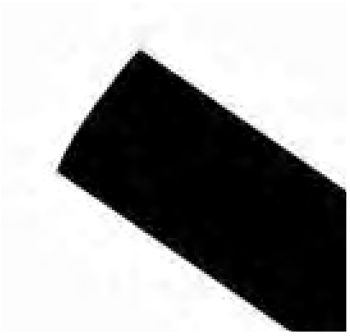
Obrázek 3.4: Testovaný obrázek.



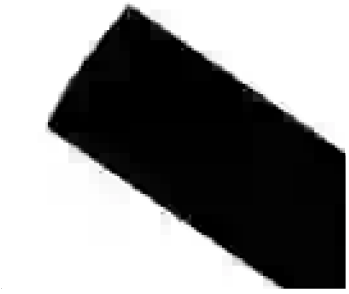
Obrázek 3.5: Originální podoba sledované části obrazu.

Obrázek 3.6: JPEG.

Obrázek 3.7: JPEG 2000.

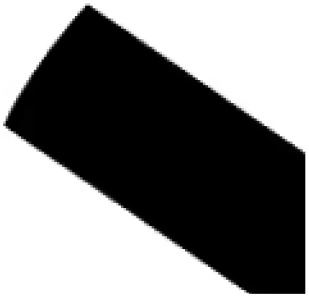


Obrázek 3.8: JPEG XR.



Obrázek 3.9: WebP.

Obrázek 3.10: AVC.



Obrázek 3.11: AV1.

Korupce dat

V této podsekci budou všechny vybrané jednotlivé formáty otestovany vůči odolnosti korupci dat. Každý vybraný formát zkomprimoval obrázek s kvalitou obrazu 60 %, poté bylo v obrazovém streamu zaměněno 50 bajtů v pořadí a to s hodnotou 0. Následně byl tento obraz zpětně převeden do původního bezeztrátového formátu. Cílem je poukázat na to, jak se jednotlivé formáty chovají při korupci dat.

Ukázka JPEG formátu po korupci dat lze vidět na obrázku 3.12. Obraz poukazuje, že od místa změny hodnot JPEG až po použitou záchytnou restartovací značku, kterou umožňuje JPEG, je obraz posunutý a zbarven do odlišných barev.

Ukázka JPEG 2000 formátu po korupci dat lze vidět na obrázku 3.13. JPEG 2000 si velmi dobře poradil se ztrátou pouze 50 bajtů. Z obrazu nelze vůbec rozeznat jakoukoliv chybu v obrazu, jelikož JPEG 2000 dekóduje výsledný obraz po vrstvách.

Ukázka JPEG XR formátu po korupci dat lze vidět na obrázku 3.14. Vidíme různobarevné kosinusoidy v rozmazaném obraze od místa korupce dat. Obraz kromě ztráty části obrazu se vyobrazuje jako rozmazaný.

Ukázka WebP formátu po korupci dat lze vidět na obrázku 3.15. WebP formát zde vykazuje poškozené části obrazu jako spleť barevných kombinací právě poškozené části. Tato část se konkrétně tedy vykresluje v podobě rozmazaných čtverců nebo obdélníků.

Ukázka AVC formátu po korupci dat lze vidět na obrázku 3.16. AVC formát od poškozené části se nevypořádal vůbec dobře. Od této poškozené části je následující každý řádek obrazu pouze zduplikován posledním nepoškozeným řádkem obrazu.

Ukázka HEIF formátu po korupci dat vidíme na obrázku 3.17. Od poškozené části obrazu HEIF formát zobrazuje směsici barevné kombinace v podobě čtvercových bloků. Toto znehodnocení obrazu se vyskytuje v ploše čtverce, kde uhlopříčku tvoří bod místa poškození a poslední pixel obrazu.

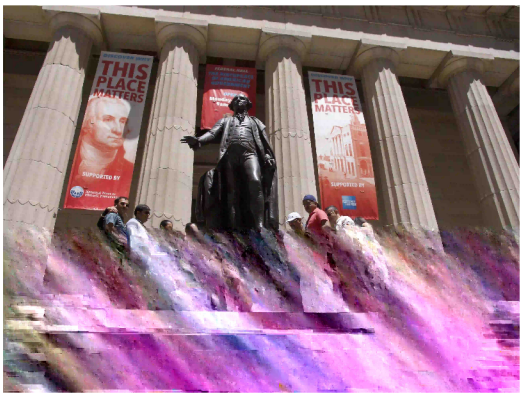
Ukázka BPG formátu po korupci dat lze vidět na obrázku 3.18. BPG formát vypadá ze všech formátů nejhůř. Ten od místa korupce dat v obrazu pouze vykreslil zbylou část obrazu barvou.

Poslední ukázkou je AV1 formát. AV1 formát po korupci dat vidíme na obrázku 3.19. AV1 formát se spolu s JPEG 2000 formátem poradil s korupcí dat nejlépe. Nenese žádné zmínky porušení nebo rozmazání obrazu.

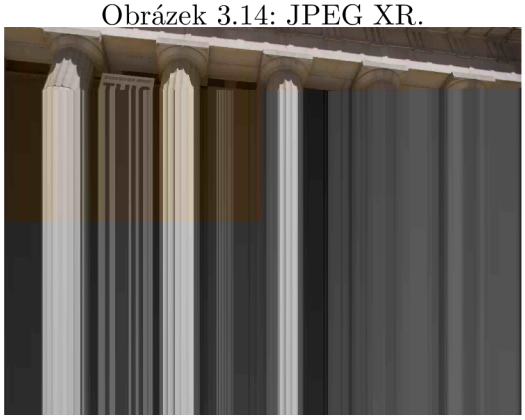
Obrázek 3.12: JPEG.

Obrázek 3.13: JPEG 2000.

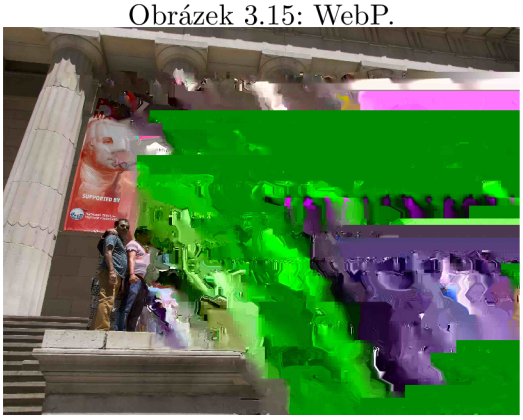
Obrázek 3.14: JPEG XR.



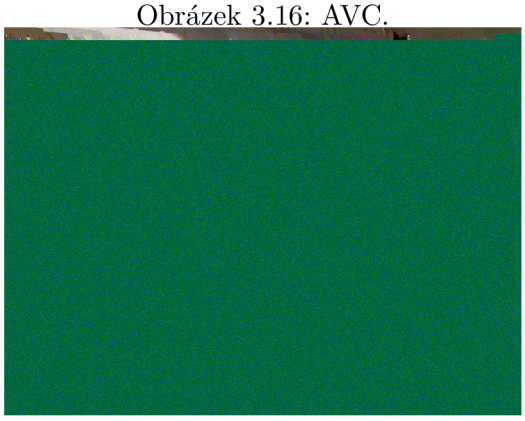
Obrázek 3.15: WebP.



Obrázek 3.16: AVC.



Obrázek 3.17: HEIF.



Obrázek 3.18: BPG.

Obrázek 3.19: AV1.

3.4 Testování

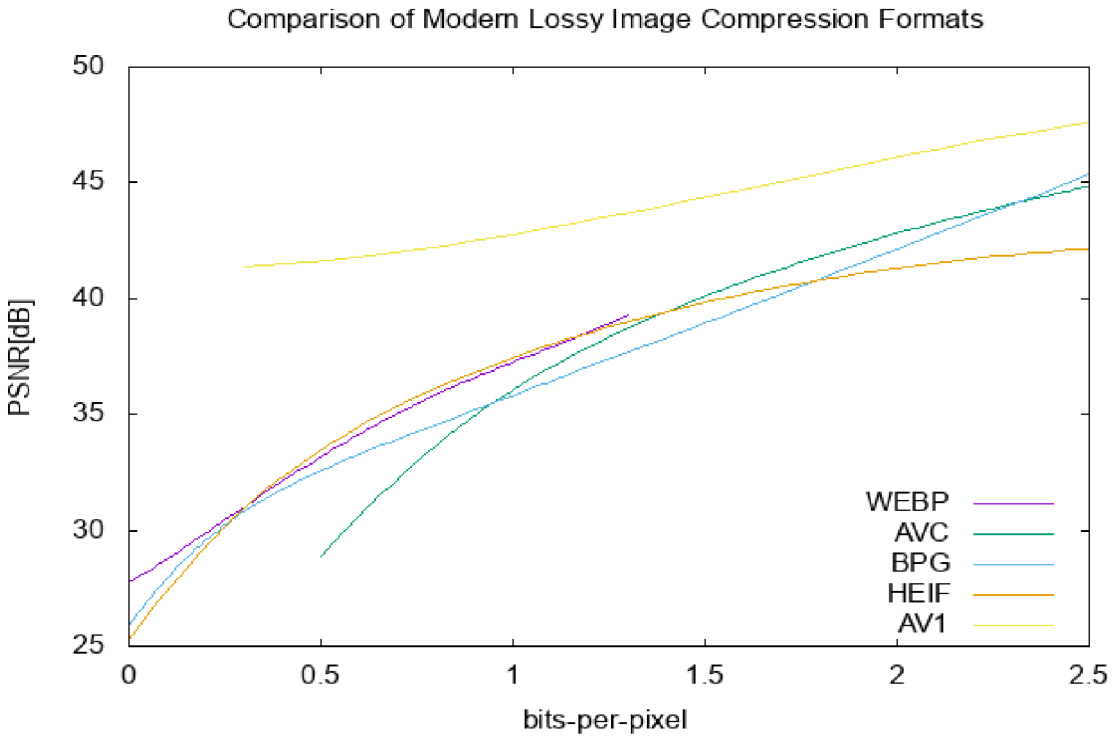
Pro kvalitní porovnání je nutné ze začátku určit správné obrázky v jednotlivých datasetech 3.1, které budou komprimovány jednotlivými knihovnami a následně budou měřeny metrikami kvality 3.2. Jako programovací jazyk programu pro testování a měření komprese obrázků byl zvolen Python. Nejprve bylo nutné implementovat porovnávací metriky kvality 3.2. Poté se musely najít vhodné knihovny, nebo nástroje pro samotnou kompresi a dekompresi obrazu u všech formátů. Pro testování JPEG a JPEG arithmetic byla využita knihovna MozJPEG (s parametry -quality, -optimize, -rgb). Pro testování Lepton

byla využita knihovna lepton. Pro testování JPEG 2000 byla využita demoverze knihovny Kakadu (s parametrem -rate). Pro testování JPEG XR byla využita knihovna jxrlib (s parametry -p, -q). Pro testování WebP byla využita knihovna webp (s parametrem -psnr). Pro testování AVC byla využita knihovna ffmpeg (s parametry -y, -c:v libx264 -crf). Pro testování BPG byla využita knihovna libbpg (s parametrem -q). Pro testování HEIF byla použita knihovna libheif (s parametrem -q). Pro testování AV1 byla využita knihovna ffmpeg_zeranoe (s parametry -y, -c:v libaom-av1, -b:v 0, -crf). Komprese každého formátů byla nastavena individuálně vzhledem k času komprese a dekomprese určitých formátů, počet měřených kvalit na jednom formátu se tedy pohybuje od 40 po 100. Každé jedno testování kvality je porovnáno s originálním obrazem pomocí všech metrik 3.2. Také byla provedena verifikace implementace, která splnila až na výjimky všechny předpoklady. Na základě této implementace byl následně program spuštěn nad každým datasetem 3.1. Výsledkem programu je vygenerování grafů pomocí knihovny Gnuplot. Výsledné grafy jsou popsány v následujících sekcích.

Fotky s vysokým rozlišením

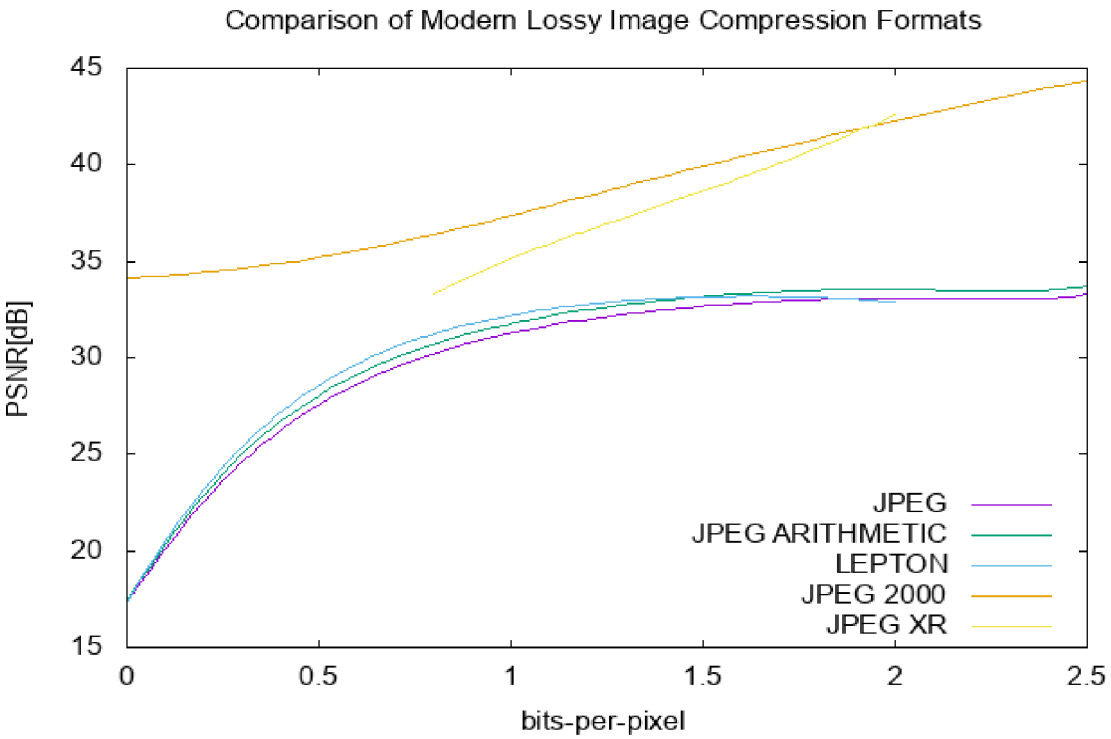
Výsledné obrázky grafů 3.21, 3.20 nám udávají hodnoty kvality obrazu s využitím metriky PSNR při použití datasetu s obrázky ve vysokých rozlišeních.

V obrázku grafu 3.20 s formáty určených pro kompresi videa se jednoznačně vyjímá na vrcholu formát AV1. Formáty HEIF a BPG se drží až na druhém místě. Tady je nutné podotknout, že komprese HEIF oproti AVC nedosahuje při větších bitech na pixel tak velké hodnoty PSNR, ale v nižších bitech na pixel vede razantně.



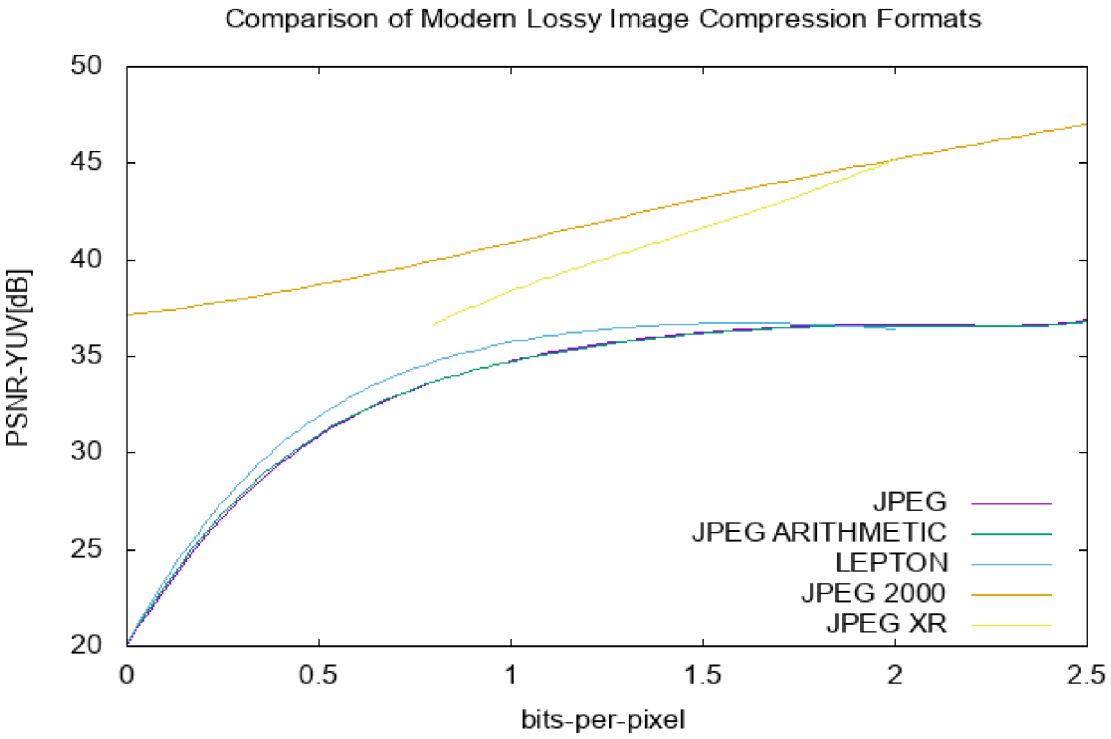
Obrázek 3.20: PSNR - dataset fotek s vysokým rozlišením.

Na obrázku grafu s formáty určených pro kompresi obrazu 3.21 se vyobrazuje jako nejlepší JPEG2000, který má obzvláště na nižších bitech na pixel vysokou kvalitu oproti modernějším kompresním formátům. Nejstarší formát JPEG i Lepton zde vykazují nejhorší výsledky.

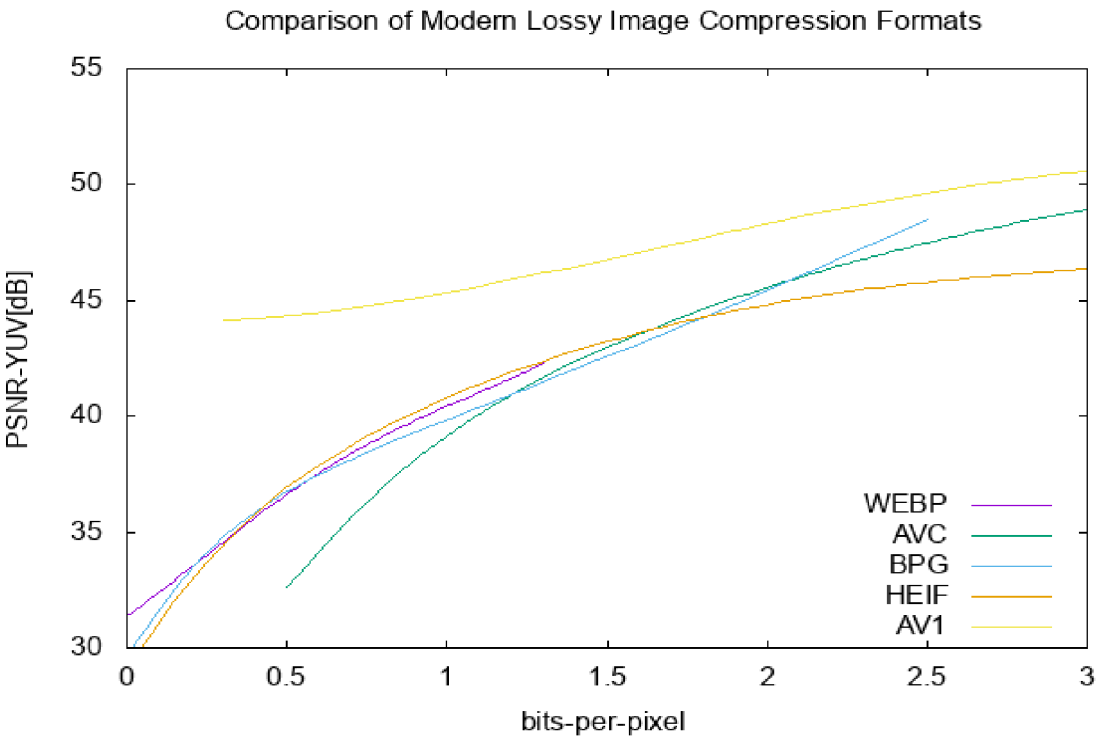


Obrázek 3.21: PSNR - dataset fotek s vysokým rozlišením.

Pro objektivní srovnání nám obrázky grafů 3.22, 3.23 udávají hodnoty kvality obrazu s využitím metriky PSNR_{YUV} při použití stejného datasetu. Graf PSNR_{YUV} nám potvrzuje výsledek grafu PSNR, kde opět vítězí na plné čáře kompresní formát AV1 v kvalitě obrazu.

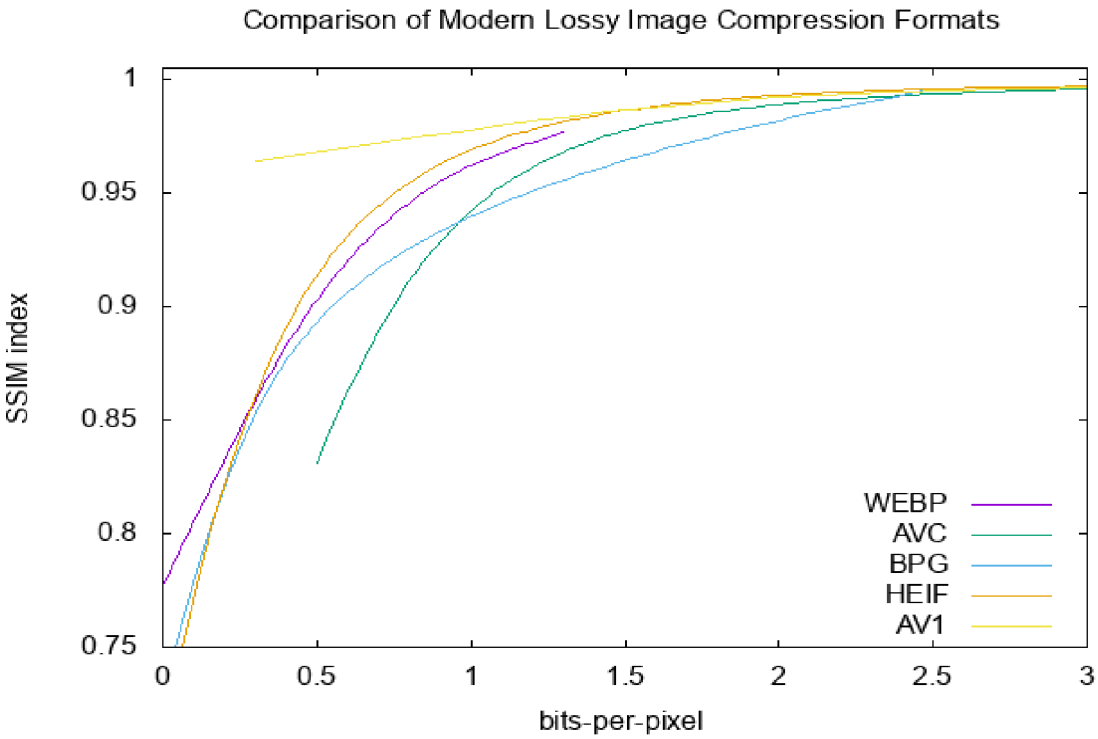


Obrázek 3.22: PSNR_{YUV} - dataset fotek s vysokým rozlišením.



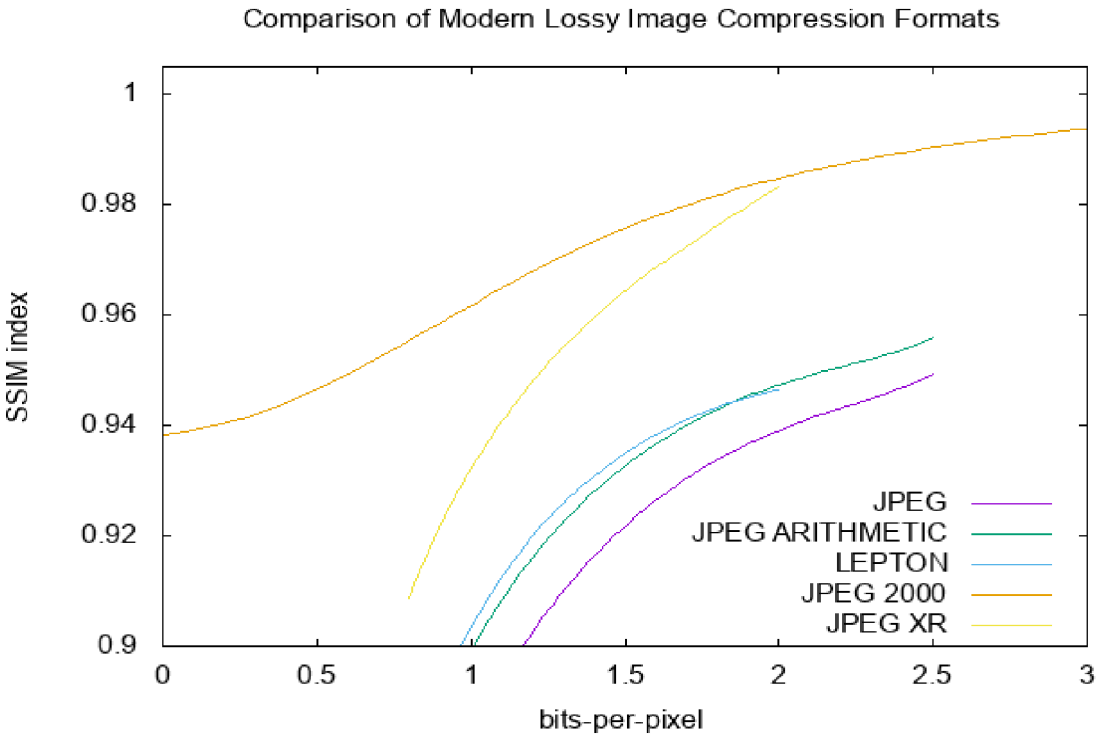
Obrázek 3.23: PSNR_{YUV} - dataset fotek s vysokým rozlišením.

Obrázky grafů 3.25, 3.24 nám udávají hodnoty kvality obrazu s využitím metriky SSIM. Opět AV1 formát zde vítězí. Kromě starších formátů jako je JPEG, JPEG s aritmetickým kódováním a Lepton, tak všechny modernější ztrátové formáty dosahují k indexu 1.

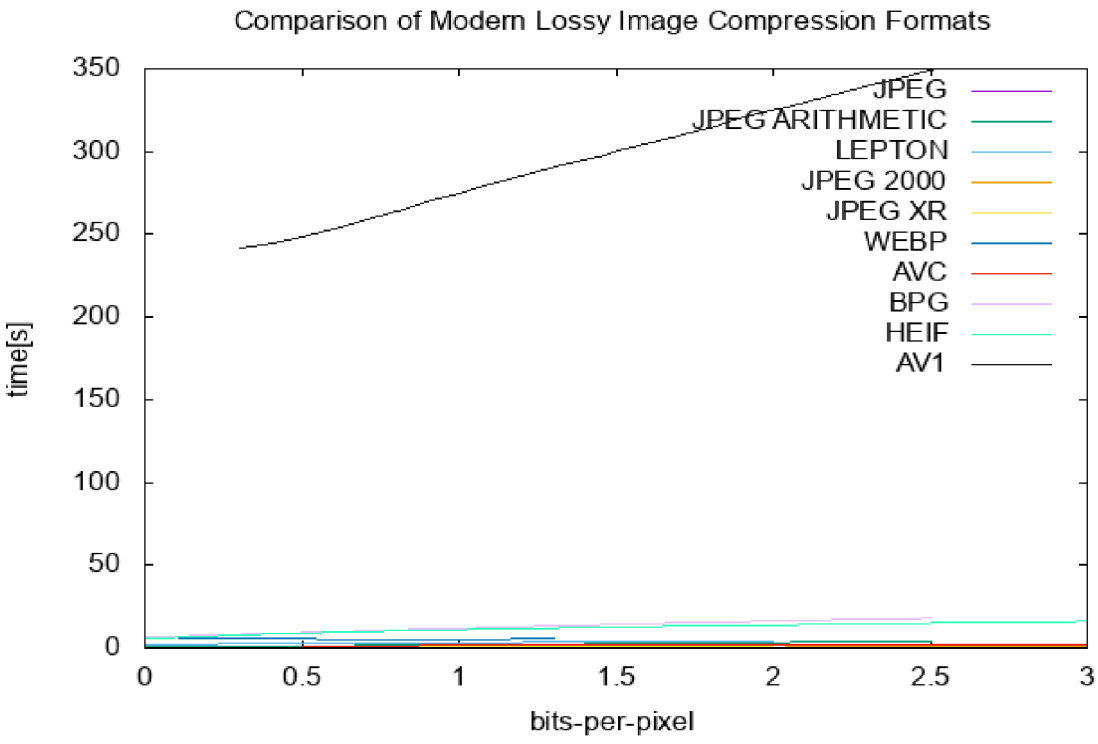


Obrázek 3.24: SSIM - dataset fotek s vysokým rozlišením.

Na obrázku grafu 3.26 můžeme vidět u jednotlivých formátu trvání procesu komprese při použití datasetu s obrázky ve vysokých rozlišeních. Všechny formáty mají průměrný čas komprese kolem 15 sekund, kromě komprese AV1, ta vyžaduje více jak 225 sekund na jediné kompresi.



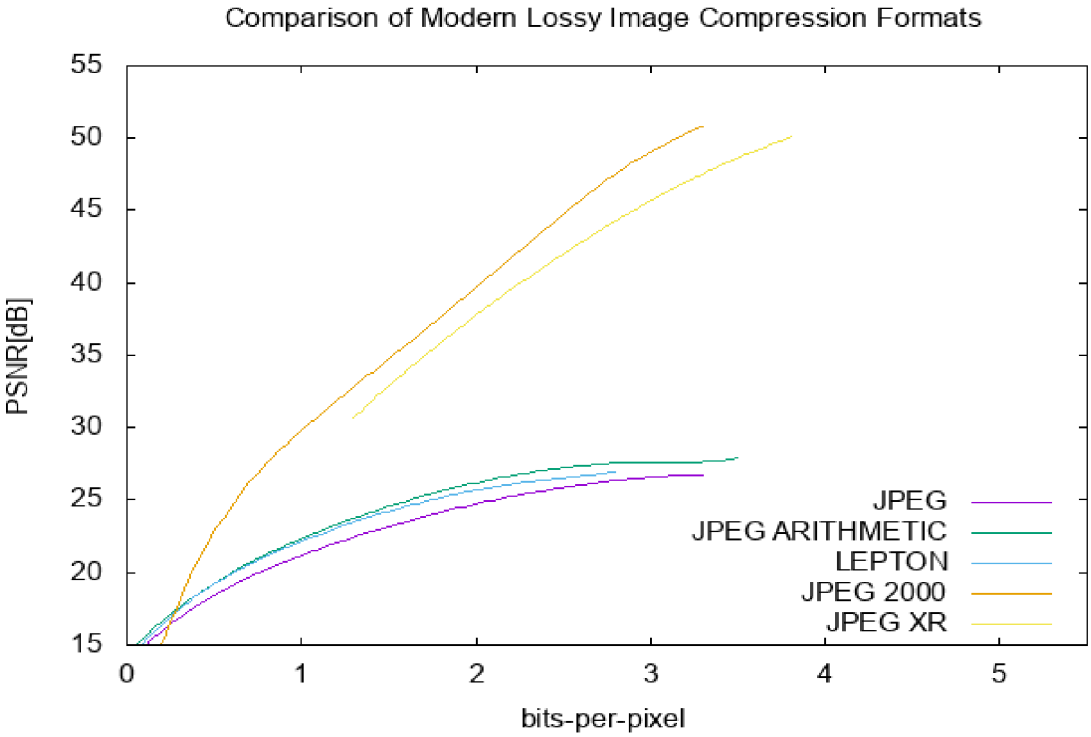
Obrázek 3.25: SSIM - dataset fotek s vysokým rozlišením.



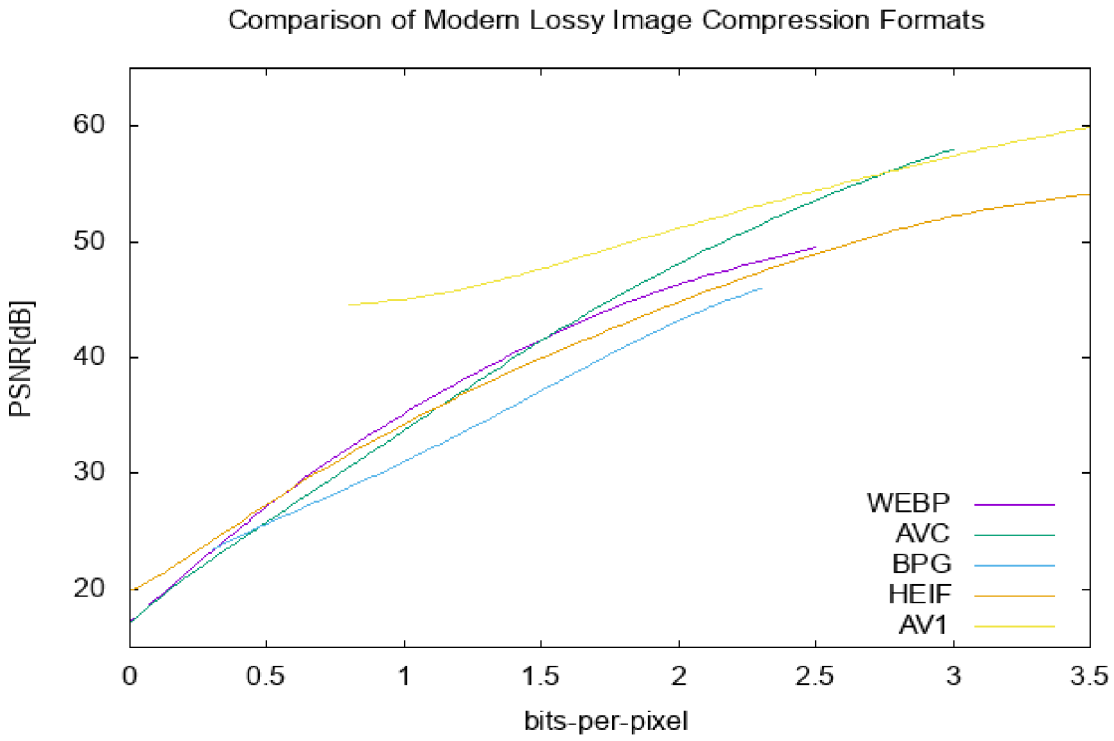
Obrázek 3.26: Čas komprese - dataset fotek s vysokým rozlišením.

Skenované dokumenty

Výsledné obrázky grafů 3.27, 3.28 nám udávají hodnoty kvality obrazu s využitím metriky PSNR při použití datasetu se skenovanými dokumenty. V grafu s formáty určených pro kompresi videa 3.28 patří mezi nejlepší formát AV1, který už nemá tak velký náskok oproti ostatním formátům jako u datasetu s obrázky ve vysokém rozlišení. Formát AVC dokonce má od 3 bitů na pixel stejnou kvalitu jako zmíněný formát AV1. Obrázek grafu 3.27 s formáty určených pro kompresi obrazu se vyobrazuje jako nejlepší opět JPEG 2000, který sice má nejlepší kompresní kvalitu, ale na nižších bitech na pixel ztrácí kvalitu obrazu. Nejstarší formát JPEG, JPEG s aritmetickým kódováním a Lepton opětovně jsou nejhorší z testovaných ztrátových formátů.

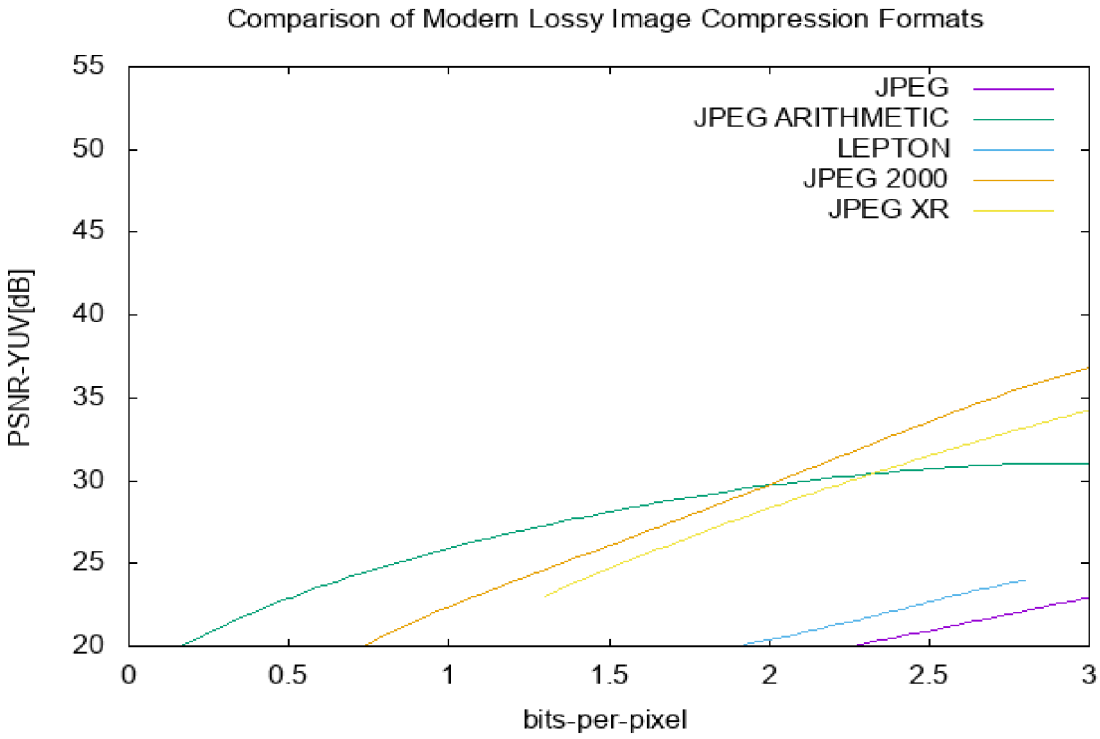


Obrázek 3.27: PSNR - dataset skenovaných dokumentů.

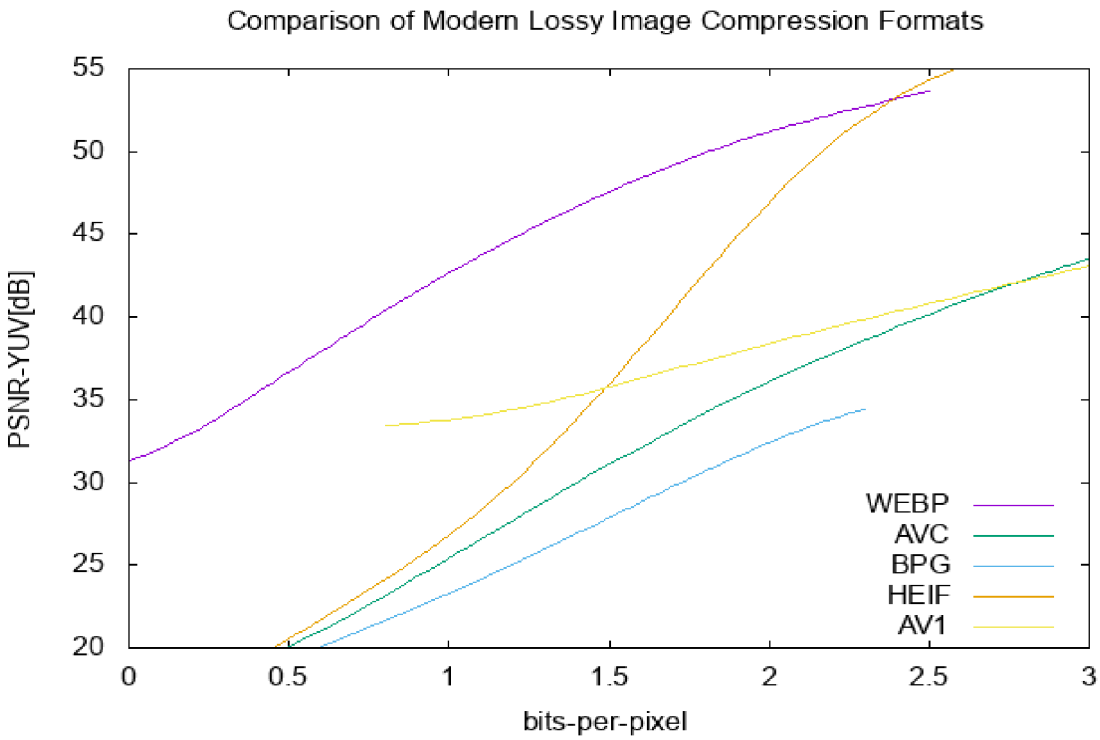


Obrázek 3.28: PSNR - dataset skenovaných dokumentů.

Pro objektivní srovnání nám obrázky grafů 3.29, 3.30 udávají hodnoty kvality obrazu s využitím metriky PSNR_{YUV} při použití stejného datasetu. Graf PSNR_{YUV} je překvapivě odlišný oproti metrice PSNR, kde nejlepším formátem vyšla AV1. U grafu s formáty určených pro kompresi videa 3.29 patří mezi nejkvalitnější formát WebP s razantním rozdílem od ostatních. Formát AV1 sice i zde má kvalitní obraz, ale nedosahuje tak velkou hodnotou kvality PSNR_{YUV}.

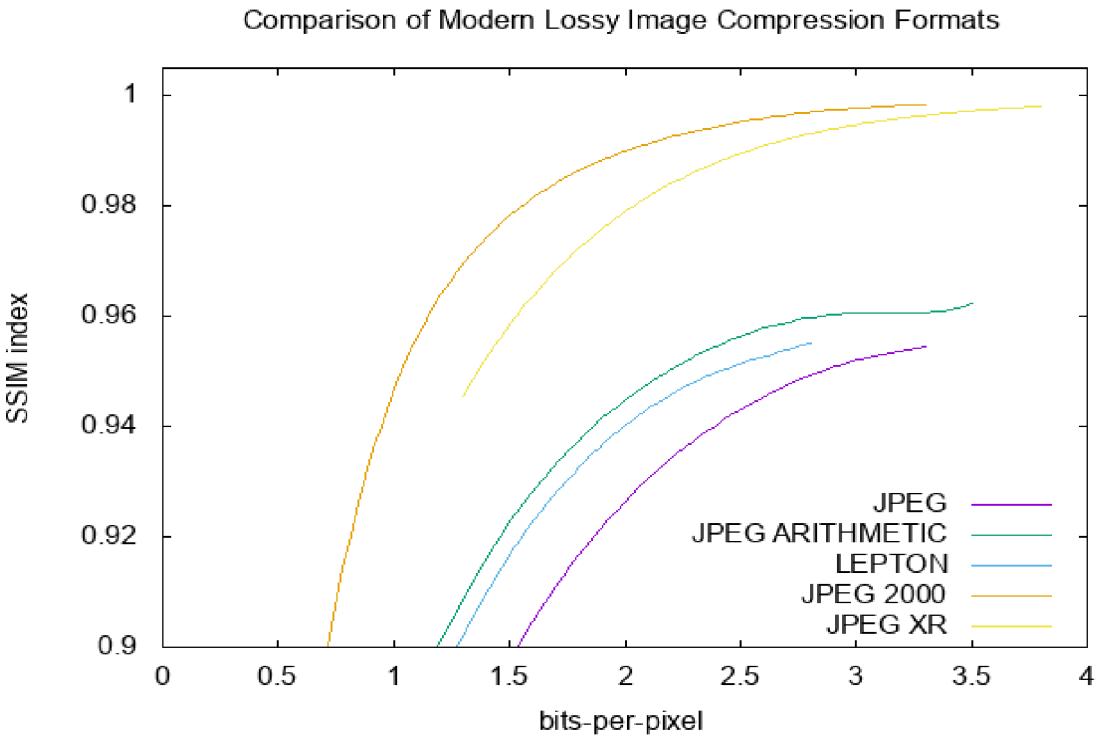


Obrázek 3.29: PSNR_{YUV} - dataset skenovaných dokumentů.



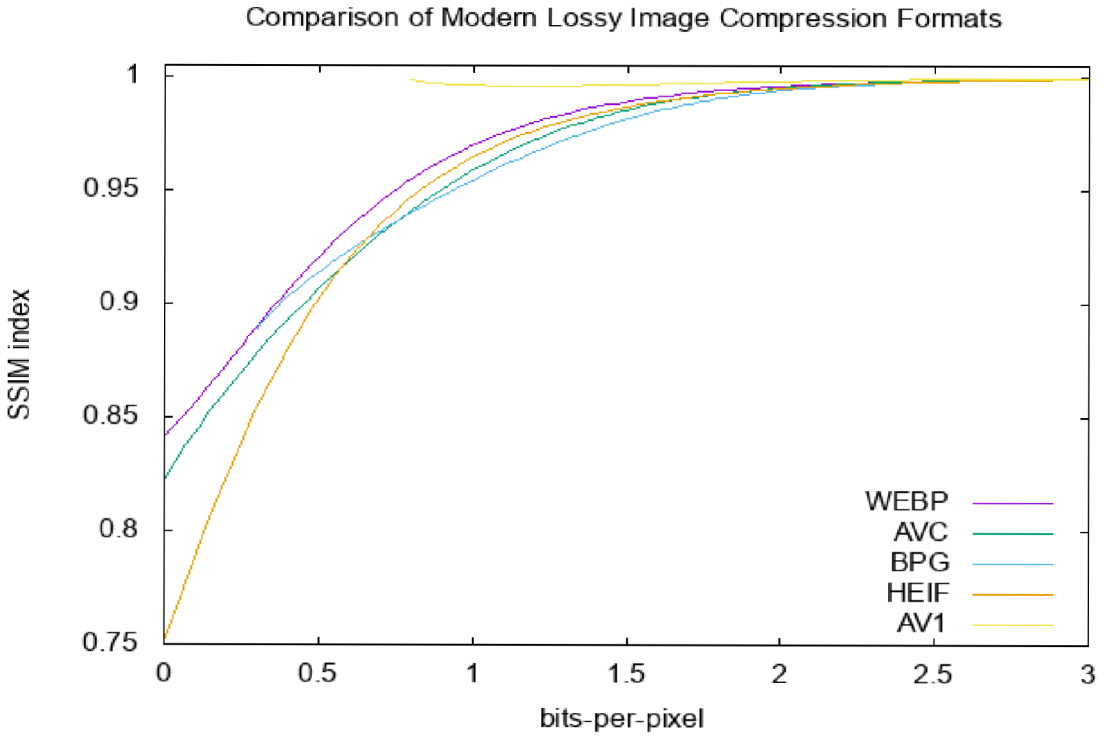
Obrázek 3.30: PSNR_{YUV} - dataset skenovaných dokumentů.

Obrázky grafů 3.31, 3.32 nám udávají hodnoty kvality obrazu s využitím metriky SSIM. Tento graf opět potvrzuje hodnoty metriky PSNR, kde se mezi nejkvalitnější formát zobrazuje AV1. Z kompresí určených pro obrazové formáty má nejlepší kvalitu JPEG2000.

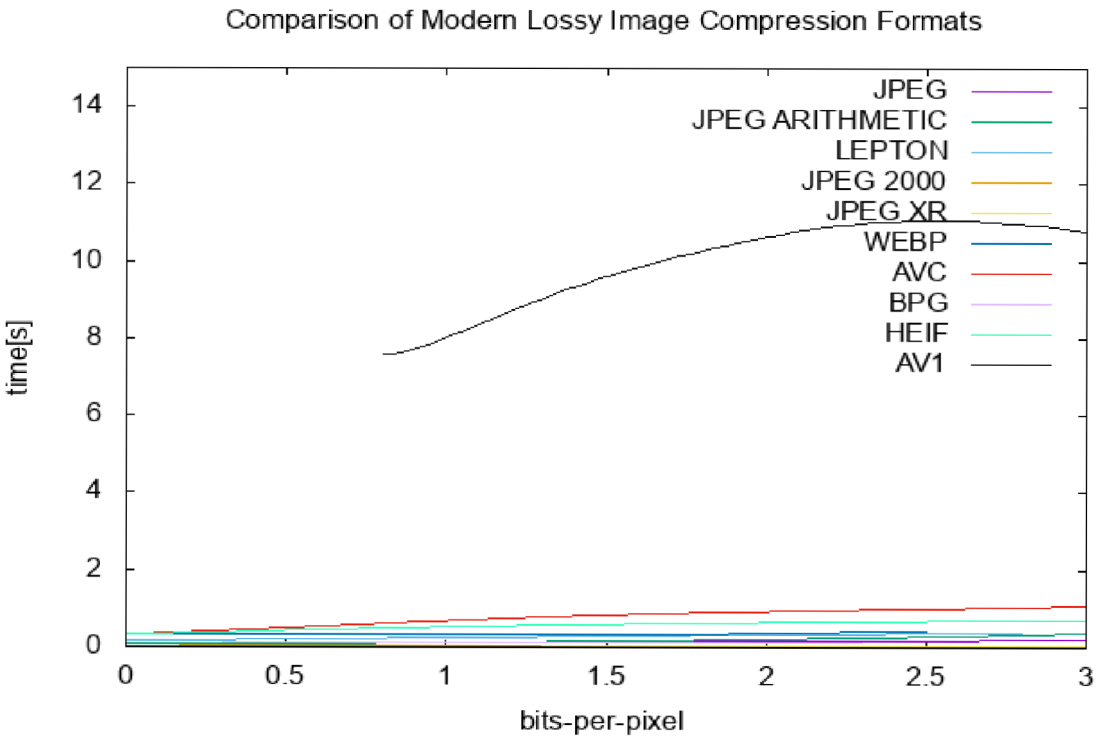


Obrázek 3.31: SSIM - dataset skenovaných dokumentů.

Na obrázku grafu 3.33 můžeme vidět u jednotlivých formátů trvání procesu komprese při použití datasetu se skenovanými dokumenty. U skenovaných dokumentů čas komprese je mnohem nižší oproti grafů z testování datasetu s obrázky s vysokým rozlišením. To je zapříčiněno velikostí originálního obrazu. Na grafu můžeme vidět, že komprese AV1 trvá mnohem déle než komprese jiných formátů.



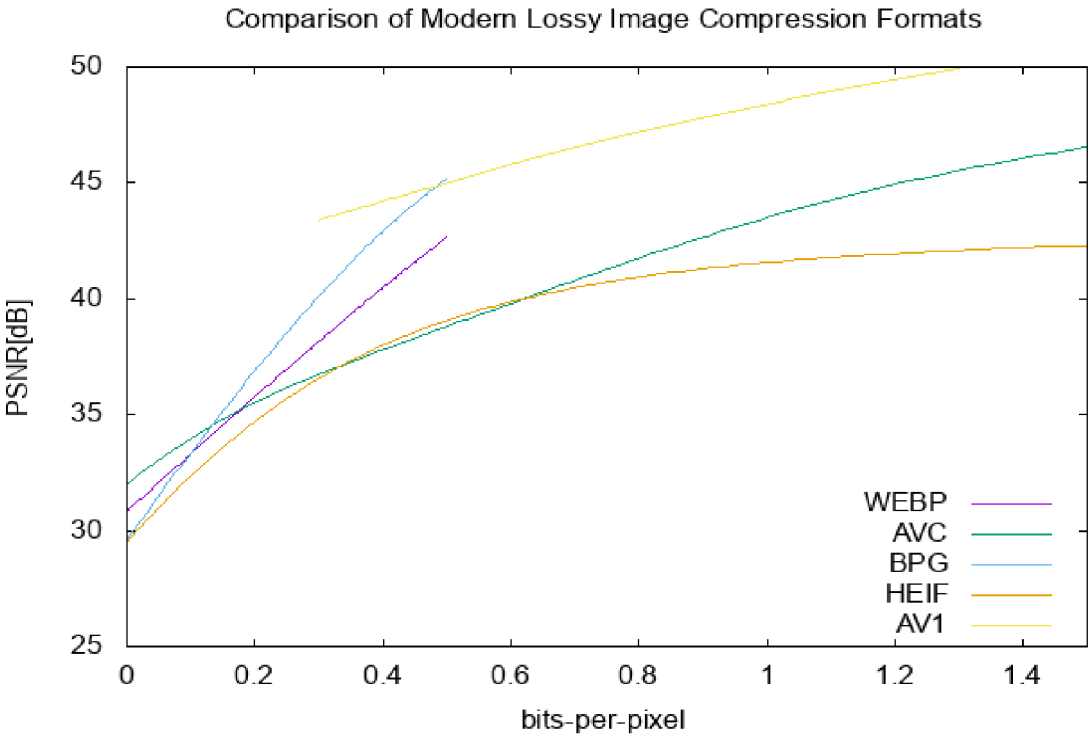
Obrázek 3.32: SSIM - dataset skenovaných dokumentů.



Obrázek 3.33: Čas komprese - dataset skenovaných dokumentů.

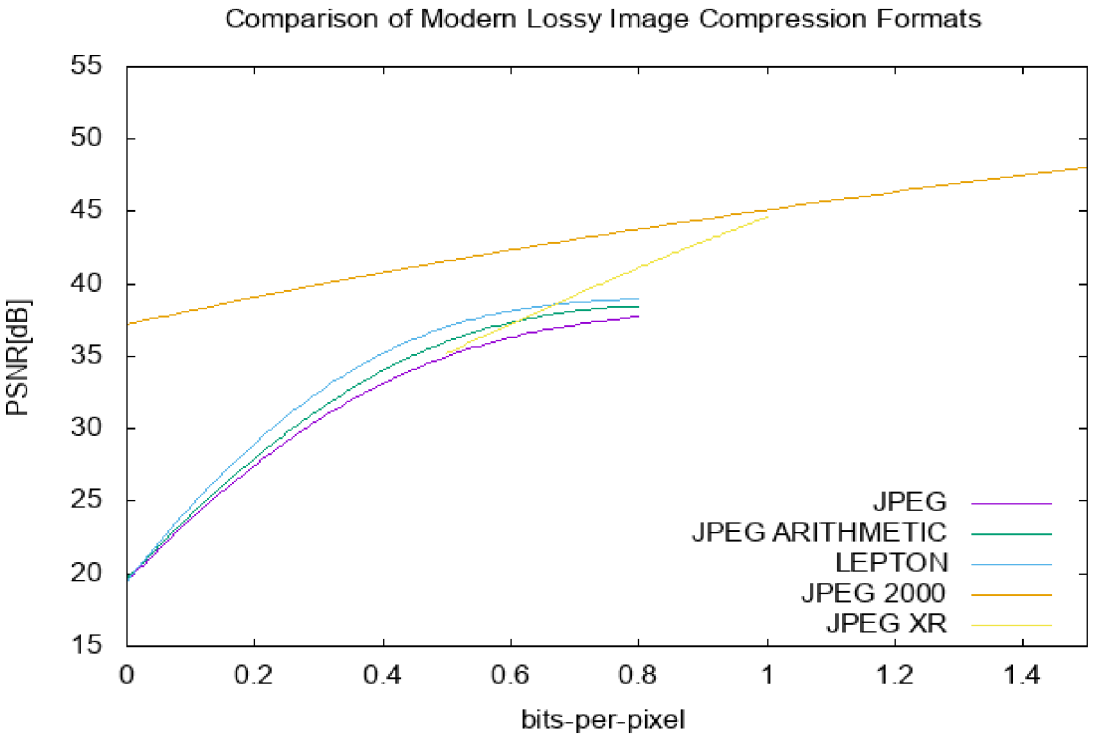
Mapy

Výsledné obrázky grafů 3.35, 3.34 nám udávají hodnoty kvality obrazu s využitím metriky PSNR při použití datasetu map. V grafu s formáty určených pro kompresi videa 3.34 patří mezi nejlepší formát opět AV1, protože má nejlepší poměr počtu bitu na pixel vůči kvalitě. Z formátů určených pro kompresi videa má překvapivě nejhorší hodnoty formát HEIF.

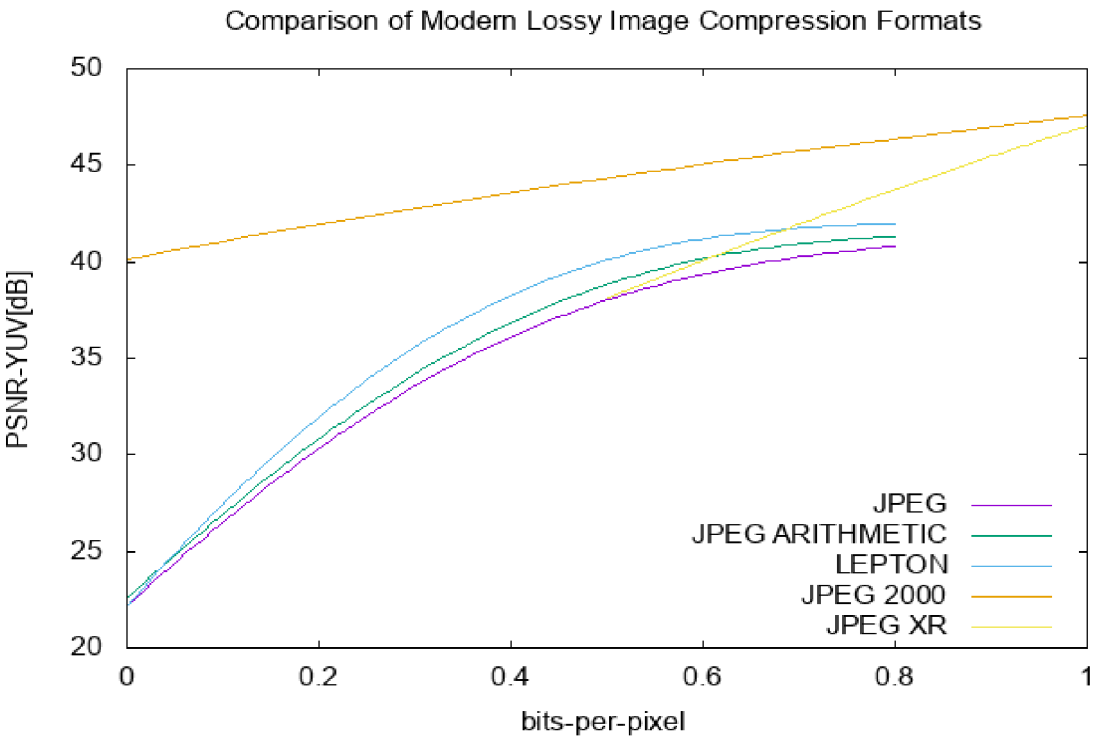


Obrázek 3.34: PSNR - dataset map.

Obrázek grafu s formáty určených pro kompresi obrazu 3.35 nám ukazuje podobné hodnoty formátů, jaké byly u ostatních testovaných datasetů. Z nich nejlepší kvality dosahuje formát JPEG 2000.

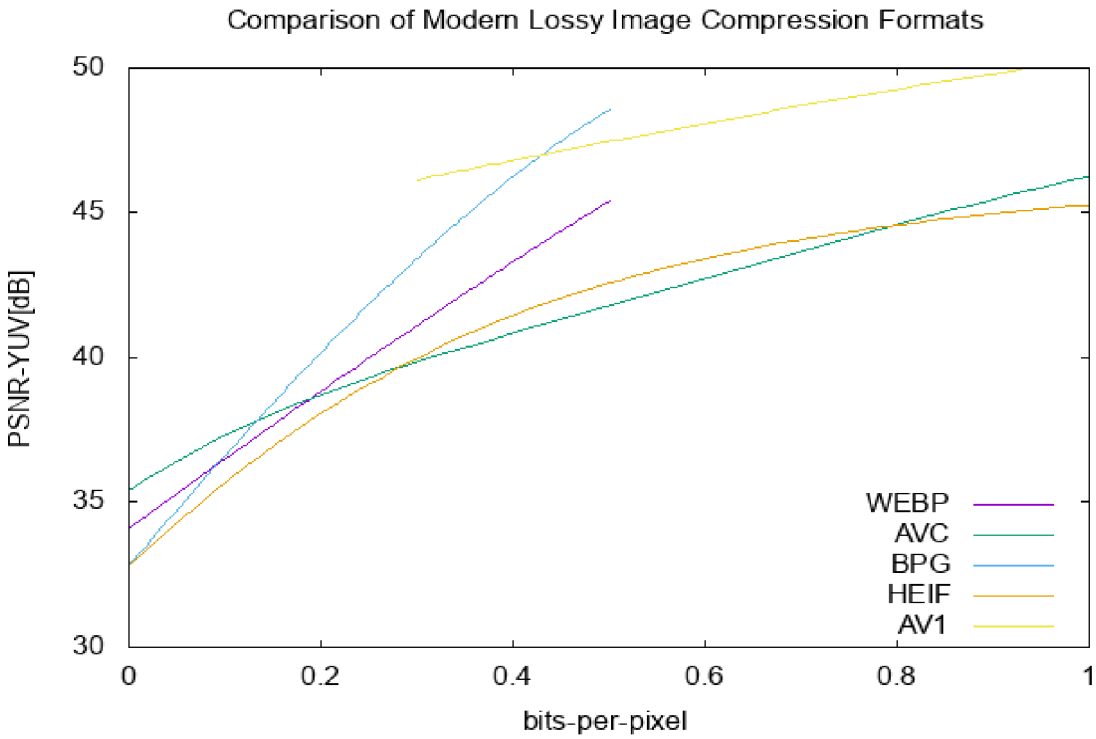


Obrázek 3.35: PSNR - dataset map.



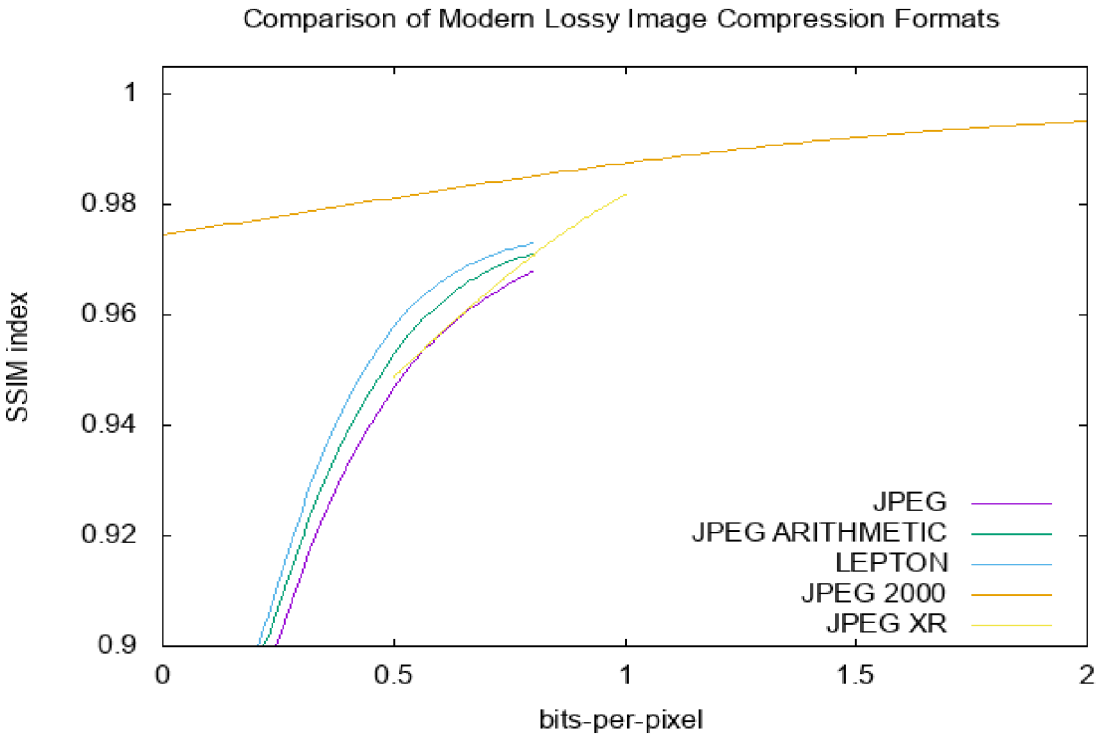
Obrázek 3.36: PSNR_{YUV} - dataset map.

Pro objektivní srovnání obrázky grafu 3.36 a 3.37 nám udávají hodnoty kvality obrazu s využitím metriky PSNR_{YUV} při použití stejného datasetu. Graf PSNR_{YUV} a graf PSNR vypadají podobně. Zde má nejlepší výsledky AV1 formát.

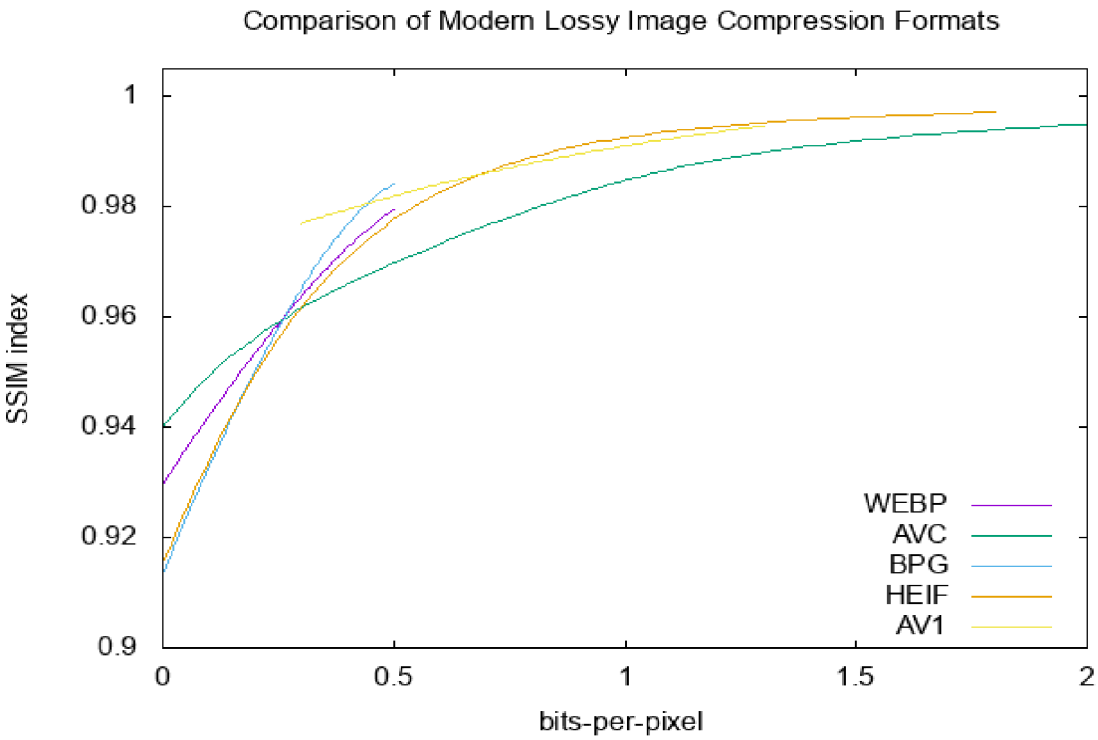


Obrázek 3.37: PSNR_{YUV} - dataset map.

Obrázky grafů 3.38, 3.39 nám udávají hodnoty kvality obrazu s využitím metriky SSIM. Na obrázku grafu 3.38 je hned zřejmé, že nejlepší kvalitou vůči kompresi obrazu je zde JPEG 2000. Ten má již od malého počtu bitů na pixel velký index SSIM. Na obrázku grafu 3.39 již není takhle jednoznačně zřetelný skok mezi nejlepšími formáty. AV1 formát má i zde velmi výborné výsledky, spolu s formátem JPEG 2000.

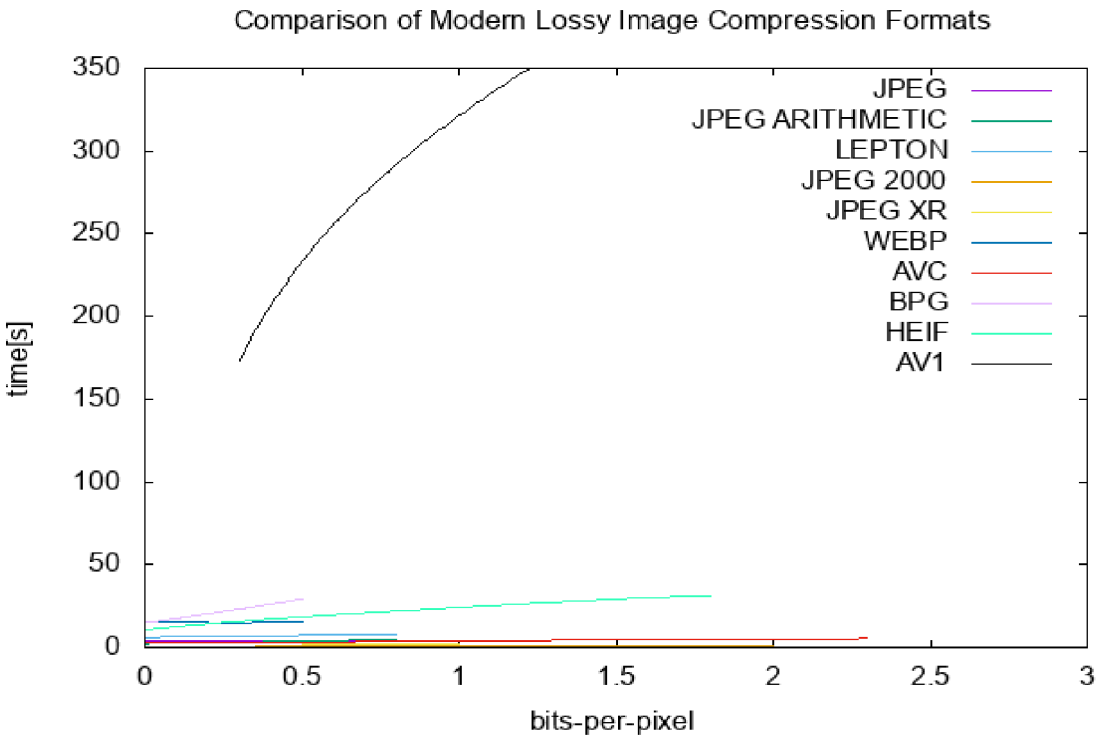


Obrázek 3.38: SSIM - dataset map.



Obrázek 3.39: SSIM - dataset map.

Na obrázku grafu 3.40 můžeme vidět u jednotlivých formátů trvání procesu komprese při použití datasetu map. Nyní lze opět vidět, že velikost obrazu může být pro některé formáty zdlouhavá na úkor kvality. Komprese formátu AV1 pracuje řádově déle než jiné formáty.



Obrázek 3.40: Čas komprese - dataset map.

Vyhodnocení

Formát AV1 v metrikách kvality obrazu PSNR, $PSNR_{YUV}$ a SSIM měl až na výjimky ve všech typech datasetů nejlepší výsledky. AV1 formát navíc neprojevuje žádné viditelně vznikající artefakty a s korupcí dat si poradil nejlépe ze všech formátů. Na základě analýzy z proběhlého testování ztrátových kompresních formátů, lze dojít k vyhodnocení, že AV1 dosahuje nejlepších celkových výsledků, a to na všech 3 testovaných datasetech.

Kapitola 4

Závěr

V této bakalářské práci byla vysvětlena ztrátová komprese. Konkrétně byly objasněny barevné modely, konverze mezi modely RGB, Y'C_bC_r a YUV. Jednoduše bylo v práci vysvětleno podvzorkování barvonosných složek a jednotlivé procesy ztrátových i bezeztrátových kompresních metod a algoritmů, jež jsou používané moderními ztrátovými formáty. Kapitola dále popisuje a srovnává všechny moderní ztrátové formáty. Jedná se o formáty JPEG, Lepton, JPEG2000, JPEG XR, WebP, BPG, HEIF, H.265 intra frame, H.264 intra frame a AV1.

Třetí kapitola popisuje zvolené datasety, implementaci testovacího programu, výsledky testování v podobě grafů a metriky pro měření kvality obrazu PSNR, MSE, SSIM. Testování ztrátových kompresních formátů mělo za cíl zjistit, jaké formáty jsou nejvhodnější pro určité typy datasetu. Kvalita obrazu byla měřena pomocí zmíněných metrik PSNR, $PSNR_{YUV}$ a SSIM.

Jak ukázaly výsledky testování v předchozí kapitole, nejlepší variantou ve všech datasetech je ztrátový formát AV1. AV1 komprese je tedy nejvhodnější pro barevné obrazy s vysokým rozlišením, pro obrazy v podobě skenovaných dokumentů a také pro obrazy map. AV1 formát nevytváří artefakty a i při malé korupci dat zachovává obraz ve velmi dobré kvalitě.

Tato práce může být v budoucnu dále rozšířena o novější moderní formáty, jako je VVC a XVC, které se teprve vyvíjí a které momentálně tato práce nepokryla. Práce se v budoucnu může případně rozšířit o jiné testovací metriky kvality.

Na základě této práce byla vytvořena představa kvality moderních ztrátových kompresních formátů obrazu v poměru velikosti komprese na kvalitu obrazu. A také byly vyobrazeny artefakty formátů spolu s odolností vůči korupci dat.

Literatura

[1] AOM AV1 - How does it work? Google, The address of the publisher: Pascal Massimino, 7 2017.

[2] Albalawi, U.; Mohanty, S. P.; Kougianos, E.: A Hardware Architecture for Better Portable Graphics (BPG) Compression Encoder. In *2015 IEEE International Symposium on Nanoelectronic and Information Systems*, Dec 2015, s. 291–296, doi:10.1109/iNIS.2015.12.

[3] Bankoski, J.; Koleszar, J.; Quillio, L.; aj.: *VP8 Data Format and Decoding Guide*. RFC 6386, 2011.

[4] Bankoski, J.; Wilkins, P.; Xu, Y.: Technical overview of VP8, an open source video codec for the web. In *2011 IEEE International Conference on Multimedia and Expo*, July 2011, ISSN 1945-788X, s. 1–6, doi:10.1109/ICME.2011.6012227.

[5] Bařina, D.; Zemčík, P.: *Multimédia - Studijní opora*. FIT VUT v Brně, 2013, 115 s.

[6] Bařina, D.; Zemčík, P.: *Multimédia - Studijní opora*. FIT VUT v Brně, 2017, 117 s.

[7] Bařina, D.; Zemčík, P.; Beran, V.: *Zpracování obrazu - Studijní opora*. FIT VUT v Brně, 2011, 92 s.

[8] Bellard, F.: BPG Image format. [Online; navštíveno 12.11.2018].
URL https://bellard.org/bpg/

[9] Bovik, A. C.; Sheikh, H. R.; Simoncelli, E. P.: Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, ročník 13, č. 4, April 2004: s. 600–612, ISSN 1057-7149, doi:10.1109/TIP.2003.819861.

[10] Chen, Y.; Murherjee, D.; Han, J.; aj.: An Overview of Core Coding Tools in the AV1 Video Codec. In *2018 Picture Coding Symposium (PCS)*, June 2018, ISSN 2472-7822, s. 41–45, doi:10.1109/PCS.2018.8456249.

[11] de Queiroz, R. L.: Processing JPEG-compressed images and documents. *IEEE Transactions on Image Processing*, ročník 7, č. 12, Dec 1998: s. 1661–1672, ISSN 1057-7149, doi:10.1109/83.730378.

[12] Dufaux, F.; Sullivan, G. J.; Ebrahimi, T.: The JPEG XR image coding standard [Standards in a Nutshell]. *IEEE Signal Processing Magazine*, ročník 26, č. 6, November 2009: s. 195–204, ISSN 1053-5888, doi:10.1109/MSP.2009.934187.

[13] Google: Compression Techniques | WebP | Google Developers. [Online; navštíveno 12.11.2018].
URL https://developers.google.com/speed/webp/docs/compression

[14] Hannuksela, M. M.; Lainema, J.; Malamal Vadakital, V. K.: The High Efficiency Image File Format Standard [Standards in a Nutshell]. *IEEE Signal Processing Magazine*, ročník 32, č. 4, July 2015: s. 150–156, ISSN 1053-5888, doi:10.1109/MSP.2015.2419292.

[15] HEIKKILÄ, L.: *High Efficiency Image File Format Implementation*. Diplomová práce, Tampere University of Technology, 2016.

[16] Horn, D. R.: Lepton image compression: saving 22% losslessly from images at 15MB/s. Jul 2016, [Online; navštíveno 10.11.2018].
URL https://blogs.dropbox.com/tech/2016/07/lepton-image-compression-saving-22-losslessly-from-images-at-15mbs/

[17] ISO/IEC 29199-2: Information technology – JPEG XR image coding system – Image coding specification. Technická zpráva, ITU-T T.832, 2016.

[18] ISO/IEC TR 29199-1: Supplement on information technology – JPEG XR image coding system – System architecture. Technická zpráva, ITU-T, 2011.

[19] Ku, C. .; Cheng, C. .; Yu, G. .; aj.: A High-Definition H.264/AVC Intra-Frame Codec IP for Digital Video and Still Camera Applications. *IEEE Transactions on Circuits and Systems for Video Technology*, ročník 16, č. 8, Aug 2006: s. 917–928, ISSN 1051-8215, doi:10.1109/TCSVT.2006.879992.

[20] Lainema, J.; Hannuksela, M. M.; Vadakital, V. K. M.; aj.: HEVC still image coding and high efficiency image file format. In *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, ISSN 2381-8549, s. 71–75, doi:10.1109/ICIP.2016.7532321.

[21] Laude, T.; Adhisantoso, Y. G.; Voges, J.; aj.: A Comparison of JEM and AV1 with HEVC: Coding Tools, Coding Efficiency and Complexity. *2018 Picture Coding Symposium (PCS)*, 2018: s. 36–40.

[22] Moto: HEVC – What are CTU, CU, CTB, CB, PB, and TB? [Online; navštíveno 12.04.2019].
URL https://codesequoia.wordpress.com/2012/10/28/hevc-ctu-cu-ctb-cb-pb-and-tb/

[23] Pan, C.; Chien, C.; Chao, W.; aj.: Architecture design of full HD JPEG XR encoder for digital photography applications. *IEEE Transactions on Consumer Electronics*, ročník 54, č. 3, August 2008: s. 963–971, ISSN 0098-3063, doi:10.1109/TCE.2008.4637574.

[24] Pastuszak, G.; Abramowski, A.: Algorithm and Architecture Design of the H.265/HEVC Intra Encoder. *IEEE Transactions on Circuits and Systems for Video Technology*, ročník 26, č. 1, Jan 2016: s. 210–222, ISSN 1051-8215, doi:10.1109/TCSVT.2015.2428571.

[25] Skodras, A.; Christopoulos, C.; Ebrahimi, T.: The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, ročník 18, č. 5, Sep. 2001: s. 36–58, ISSN 1053-5888, doi:10.1109/79.952804.

[26] ISO/IEC FDIS 23008-12 3rd edition Image File Format. Standard, MPEG-H, Červen 2015.

[27] Sullivan, G. J.; Ohm, J.; Han, W.; aj.: Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, ročník 22, č. 12, Dec 2012: s. 1649–1668, ISSN 1051-8215, doi:10.1109/TCSVT.2012.2221191.

[28] Wallace, G. K.: The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, ročník 38, č. 1, Feb 1992: s. xviii–xxxiv, ISSN 0098-3063, doi:10.1109/30.125072.

[29] Wiegand, T.; Sullivan, G. J.; Bjontegaard, G.; aj.: Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, ročník 13, č. 7, July 2003: s. 560–576, ISSN 1051-8215, doi:10.1109/TCSVT.2003.815165.

Příloha A

Obsah přiloženého DVD

DVD obsahuje:

- Python script pro testování formátů
- Složka s datasety obrázků
- Složka s knihovnama
- Složka pro výstupy Python scriptu
- README soubor
- Bakalářskou práci ve formátu PDF
- Bakalářskou práci ve formátu LaTeX