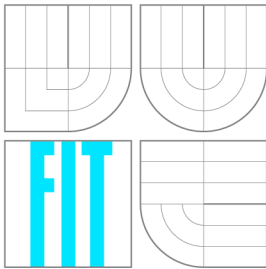


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SROVNÁNÍ FORMÁTŮ PRO ZTRÁTOVOU KOMPRESI OBRAZU

COMPARISON OF LOSSY IMAGE COMPRESSION FORMATS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VEDOUCÍ PRÁCE

SUPERVISOR

LADISLAV KOLÁČEK

Ing. DAVID BAŘINA

BRNO 2014

Abstrakt

Tato bakalářská práce se zabývá srovnáním formátů pro ztrátovou kompresi obrazu. Kapitoly práce popisují principy komprese obrazu a samotné ztrátové formáty. Konkrétně se jedná o ztrátové formáty JPEG, JPEG 2000, JPEG XR a WebP. Práce zahrnuje teoretické srovnání výše zmíněných formátů spolu s metodami, které se používají pro měření kvality obrazu. Praktická část obsahuje výsledky měření kvality jednotlivých formátů a popis návrhu společně s implementací výukového programu. Měření prokázalo, že nejlepším kompresním poměrem disponuje formát JPEG 2000 a v nízkých datových tocích formát WebP.

Abstract

This bachelor thesis deals with comparing of lossy image compression formats. The chapters describe principles of image compression and lossy formats themselves. Specifically the lossy formats JPEG, JPEG 2000, JPEG XR and WebP. The thesis includes theoretical comparison of the formats mentioned above, along with the methods used for measuring image quality. The practical part contains the results of the quality measurements of each format and description of design together with implementation of the tutorial. The measuring demonstrate, that the best compression ratio has format JPEG 2000 and at lower bitrates format WebP.

Klíčová slova

DFT, JPEG, DCT, entropické kódování, RLE, Huffmanovo kódování, JPEG 2000, DWT, EBCOT, aritmetický kodér, JPEG XR, LBT, OPF, FCT, VLC, WebP, WHT, PSNR, SSIM.

Keywords

DFT, JPEG, DCT, entropy encoding, RLE, Huffman coding, JPEG 2000, DWT, EBCOT, arithmetic encoder, JPEG XR, LBT, OPF, FCT, VLC, WebP, WHT, PSNR, SSIM.

Citace

Ladislav Kolářek: Srovnání formátů pro ztrátovou kompresi obrazu, bakalářská práce, Brno, FIT VUT v Brně, 2014

Srovnání formátů pro ztrátovou kompresi obrazu

Prohlášení

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

.....
Ladislav Koláček
21. května 2014

Poděkování

Na tomto místě bych především rád poděkoval svému vedoucímu Ing. Davidu Bařinovi za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování bakalářské práce. Dále bych rád poděkoval své rodině a přítelkyni za podporu, kterou mi poskytovali po celou dobu studia.

© Ladislav Koláček, 2014.

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

Obsah

1	Úvod	2
2	Formáty ztrátové komprese obrazu	3
2.1	JPEG	9
2.2	JPEG 2000	11
2.3	JPEG XR	15
2.4	WebP	20
3	Srovnání formátů	25
3.1	Metriky srovnání	25
3.2	Vlastnosti formátů	26
3.3	Kvalita komprese	28
3.4	Odolnost proti chybám	32
3.5	Interaktivní tutoriál	34
4	Závěr	37
A	Odolnost vůči chybám	40
B	Srovnání kompresního poměru	42

Kapitola 1

Úvod

Tato práce se věnuje popisu ztrátových formátů komprese obrazu a jejich vzájemnému srovnání. Dané téma jsem si vybral zejména proto, že mě tato problematika zajímá a dále jsem chtěl také konkrétně zjistit, jakým způsobem probíhá samotný proces komprese a jak je možné poměřovat kvalitu obrazu.

Hlavním cílem této práce je nalézt ztrátový formát, který disponuje nejlepším kompresním poměrem při zachování vysoké kvality obrazu. Za tímto účelem jsem porovnával nejvíce používané formáty jako JPEG, JPEG 2000, JPEG XR a WebP.

Tato práce je členěna do tří hlavních kapitol. Kapitola **Formáty ztrátové komprese obrazu** zahrnuje popis jednotlivých formátů a základní principy komprese obrazu, jejichž znalost je nezbytná k pochopení dané problematiky. Další kapitola **Srovnání formátů** se zabývá metrikami, které je možné uplatnit pro zjištění kvality komprimovaného obrazu. Dále je v této kapitole provedeno teoretické a praktické srovnání všech porovnávaných formátů. V sekci **Interaktivní tutoriál** je popsán návrh a implementace výukového programu, který by měl uživateli sloužit k lepšímu pochopení této problematiky a názornému srovnání kvality komprese jednotlivých formátů. V kapitole **Závěr** nakonec zhodnotím své poznatky z naměřených hodnot z provedených testování a určím nejkvalitnější ztrátový formát.

Kapitola 2

Formáty ztrátové komprese obrazu

V této kapitole budou nejprve probrány základní principy komprese obrazu a dále zde budou popsány jednotlivé ztrátové formáty.

Mezi základní principy komprese se řadí transformace barev do vhodného barevného prostoru spolu s podvzorkováním složek. Při kompresi obrazu se často využívá transformace barev z *RGB*, *CMYK*, popř. jiného formátu do barevného prostoru YC_bC_r . Složka Y zde nese informaci o světlosti pixelů. Složky C_b a C_r nesou rozdílové hodnoty barev pixelů oproti složce Y . V této sekci jsem čerpal informace z materiálu [3].

$$\begin{aligned} Y &= 0,299 R + 0,587 G + 0,114 B \\ C_b &= -0,1687R - 0,3313G + 0,5 B + 128 \\ C_r &= 0,5 R - 0,4187G - 0,0813B + 128 \end{aligned} \tag{2.1}$$

Jelikož je lidské oko mnohem více citlivé na změny jasu (Y) než na barvonosné složky (C_b , C_r), nabízí se využití podvzorkování těchto složek. Člověk obvykle není schopen zaznamenat ztrátu informace barvonosných složek. Pixely blízko u sebe, které mají stejnou barvu, dohromady tvoří blok.

Blok má obvykle šířku 4 a výšku 2 vzorky. Podvzorkování vyjadřuje trojice čísel $J : a : b$, kde J značí šířku bloku. Hodnota a vyjadřuje počet barvonosných složek v horním řádku bloku a hodnota b počet barvonosných složek ve spodním řádku. Typicky se používají hodnoty 4 : 4 : 4 (nepodvzorkováno) a 4 : 1 : 1 (horizontální podvzorkování barvonosných složek na čtvrtinu). V JPEGu se využívá podvzorkování 4 : 2 : 2 a 4 : 2 : 0.

2.0.1 Diskrétní Fourierova transformace

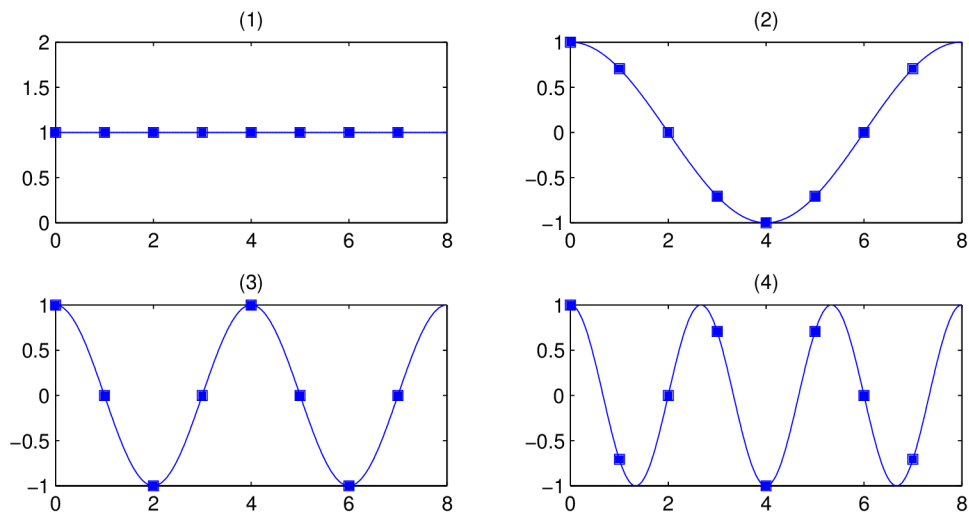
Diskrétní Fourierova transformace (DFT) je základním nástrojem pro zpracování obrazu a řadí se mezi frekvenční transformace. Uplatňuje se v návrhu a implementaci lineárních filtrů a dále v algoritmech pro kompresi obrazu.

Fourierova transformace (FT) nám popisuje, kolik energie nese signál dané frekvence. Využívá srovnání signálů s různě velkými komplexními exponenciálami. Diskrétní Fourierova transformace se řadí mezi Fourierovy transformace pro diskrétní signály konečné délky. Sekvence N čísel diskrétní Fourierovy transformace je definována ve vztahu 2.2. V této sekci jsem čerpal informace z materiálu [4].

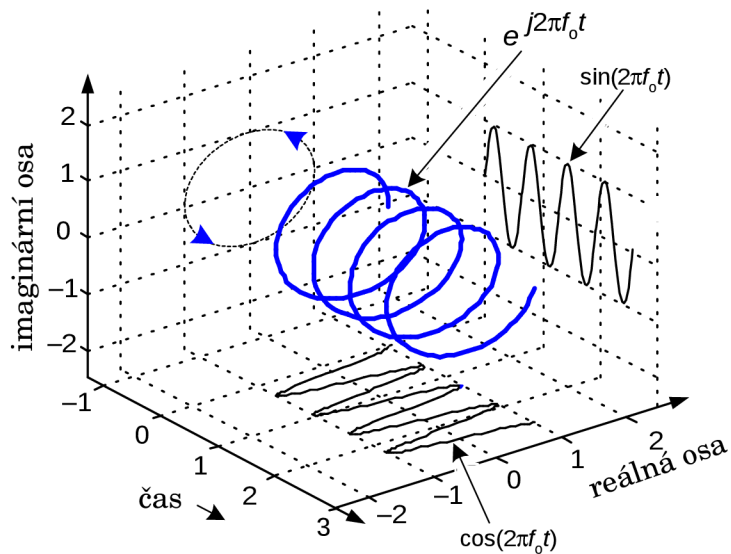
$$\hat{f}[k] = \sum_{n=0}^{N-1} f[n]e^{-i\frac{2\pi}{N}kn} \quad (2.2)$$

pro $0 \leq k < N$. Použitím této transformace vznikne N komplexních koeficientů \hat{f} . Tyto koeficienty lze označit za spektrum signálu. Tyto koeficienty jsou symetrické. Koeficient s indexem $k = 0$ je vyjádřením stejnosměrné složky signálu. Tento koeficient bude dále označován jako koeficient DC (*direct current*).

Pro analýzu obrazu se používá dvourozměrná diskretní Fourierova transformace. Na Obrázku 2.1 jsou zobrazeny první 4 báze kosinové funkce a na následujícím Obrázku 2.2 je vyobrazena komplexní exponenciála v komplexní rovině.



Obrázek 2.1: Průběhy báze kosinových funkcí u DFT. Graf (1) vyjadřuje stejnosměrnou složku signálu (DC).



Obrázek 2.2: Komplexní exponenciála DFT zobrazená v komplexní rovině. [12]

2.0.2 Diskrétní kosinová transformace

Pokud chceme použít transformaci za účelem např. komprese obrazu, je více vhodné namísto diskrétní Fourierovy transformace využít diskrétní kosinovou transformaci (DCT). Diskrétní kosinová transformace je klíčová metoda pro ztrátovou kompresní metodu **JPEG**. Využívá srovnání signálů s různě velkými kosinusoidami. Podobně jako je tomu u diskrétní Fourierovy transformace, transformací N reálných čísel vznikne N reálných koeficientů, které lze nazývat spektrem. I zde je koeficient $k = 0$ stejnosměrnou složkou DC. Koeficienty diskrétní kosinové transformace zde nejsou symetrické. Sekvence N čísel je definována ve vztahu 2.3. V této sekci jsem čerpal informace z materiálů [4] a [16].

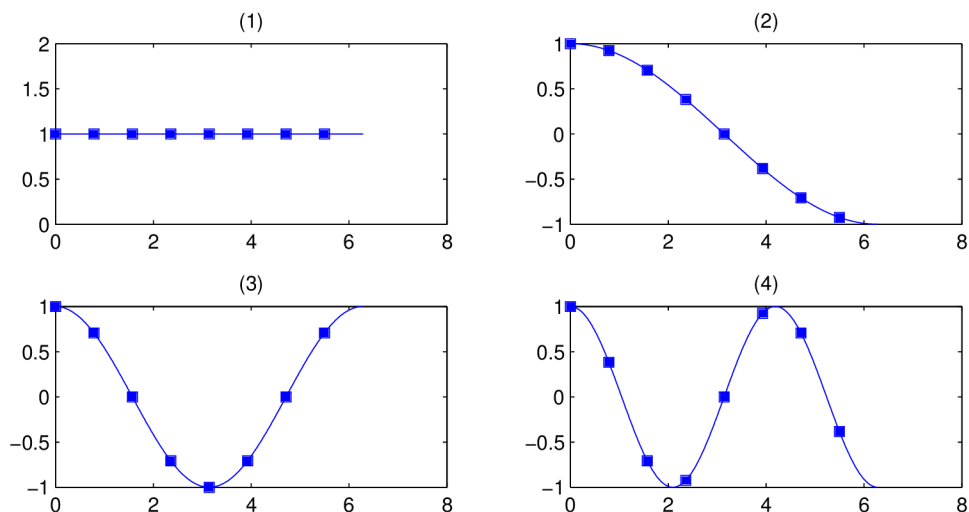
$$c[k] = \sum_{n=0}^{N-1} f[n]g_k[n] \quad (2.3)$$

pro $0 \leq k < N$, kde

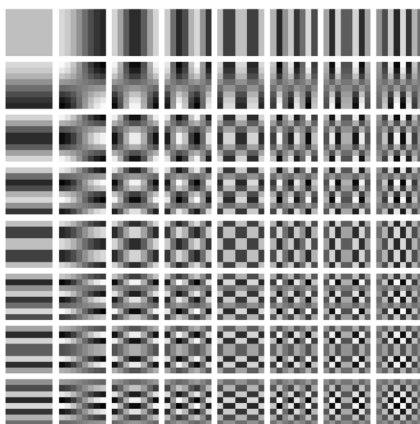
$$g_k[n] = \lambda_k \sqrt{\frac{2}{N}} \cos \left[\frac{k\pi}{N} \left(n + \frac{1}{2} \right) \right] \quad (2.4)$$

$$\lambda_k = \begin{cases} 1/\sqrt{2} & : k = 0 \\ 1 & : k \neq 0 \end{cases} \quad (2.5)$$

Na Obrázku 2.3 jsou pro lepší ilustraci vyobrazeny první 4 bázevé funkce jednorozměrné diskrétní kosinové transformace a na následujícím Obrázku 2.4 jsou zobrazeny bázevé funkce dvourozměrné diskrétní kosinové transformace, které se používají pro analýzu obrazu.



Obrázek 2.3: První čtyři průběhy bázevých kosinových funkcí u DCT. Graf (1) vyjadřuje stejnosměrnou složku signálu (DC).



Obrázek 2.4: Bázové funkce DCT pro 8×8 vzorků. Čtverec vlevo nahoře vyjadřuje stejnosměrnou složku signálu (DC).

2.0.3 Diskrétní vlnková transformace

Jedná se o diskrétní variantu vlnkové transformace. Diskrétní vlnková transformace (DWT) je tvořena několika typy koeficientů. Přibližné koeficienty $c_j(n)$ vyjadřují hrubý tvar signálu. Podrobné koeficienty $d_j(n)$ udávají detaily signálu. Index j zde značí úroveň rozkladu.

Je možné použít výpočet pomocí konvoluce s filtry h pro dolní propust a filtry g pro horní propust. Tento postup výpočtu diskrétní vlnkové transformace nazýváme rychlou vlnkovou transformací (FWT):

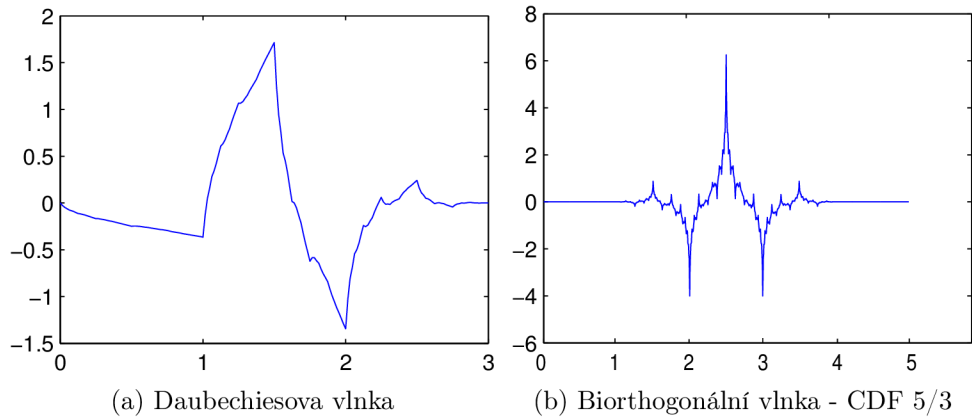
$$c_{j+1}(n) = \sum_k h(k - 2n)c_j(k), \quad (2.6)$$

$$d_{j+1}(n) = \sum_k g(k - 2n)c_j(k). \quad (2.7)$$

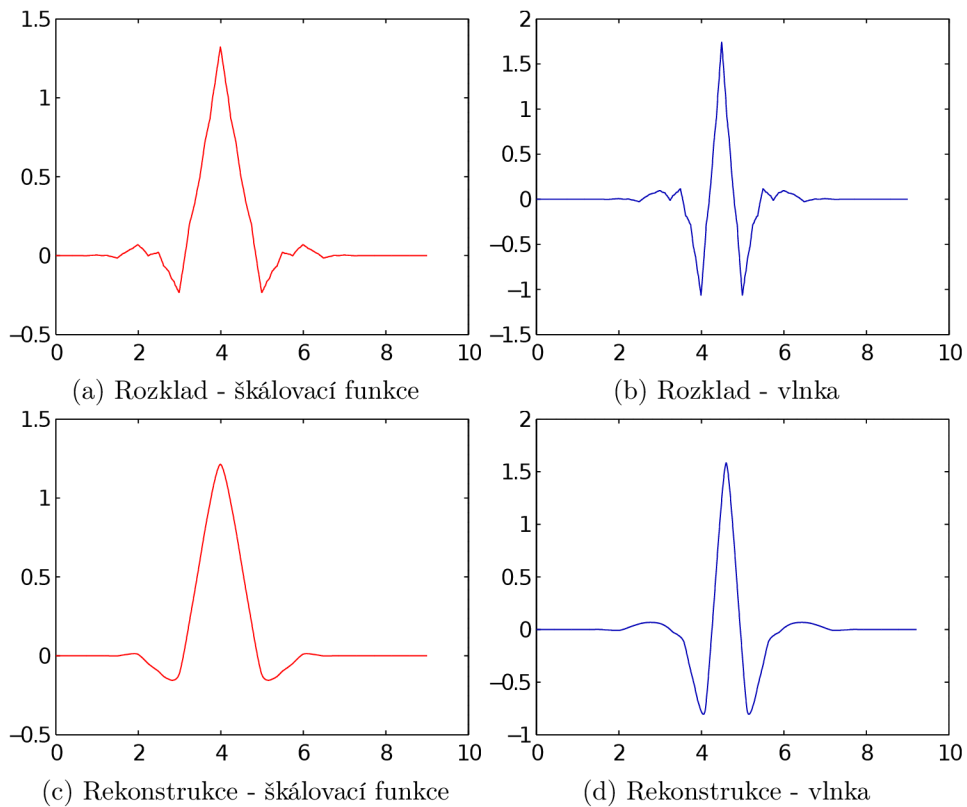
Při provádění transformace nejprve dosadíme za c_0 hodnotu signálu, kterou chceme rozložit. Využívá se operace podvzorkování dvěma, při které dochází k zahazení druhého vzorku signálu. V případě, že budeme chtít provést úplný rozklad, musíme provést rozklad až do úrovně jednoho koeficientu.

Rekonstrukce signálu se provádí inverzními postupy pomocí rekonstrukčních filtrů. Diskrétní vlnková transformace je klíčovou metodou pro ztrátovou kompresní metodu **JPEG 2000**. V této sekci jsem čerpal informace z materiálu [4].

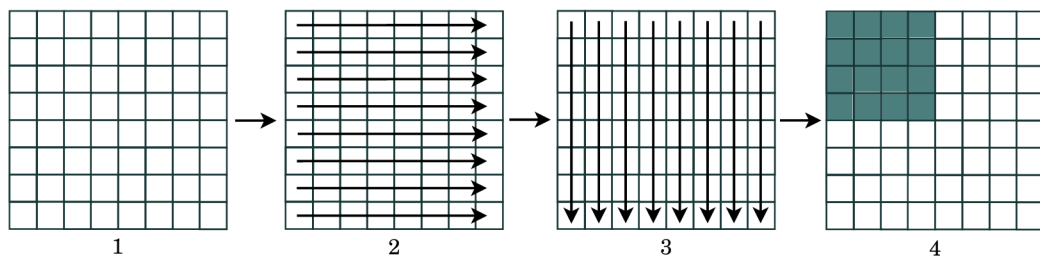
Na Obrázcích 2.5 a 2.6 jsou zobrazeny vlnky DWT. JPEG 2000 využívá pro účely komprese dvě vlnky, a to CDF 5/3 a CDF 9/7. Vlnka CDF 5/3 je zobrazena na Obrázku 2.5b a využívá se v bezztrátové kompresi. Vlnka CDF 9/7 se používá ke ztrátové kompresi obrazu a můžeme ji vidět na Obrázku 2.6.



Obrázek 2.5: Na těchto obrázcích jsou demonstrovány vlnky diskrétní vlnkové transformace. Na Obrázku 2.5b se nachází biorthogonální vlnka CDF 5/3, která se používá ve formátu JPEG 2000 v bezztrátové kompresi.

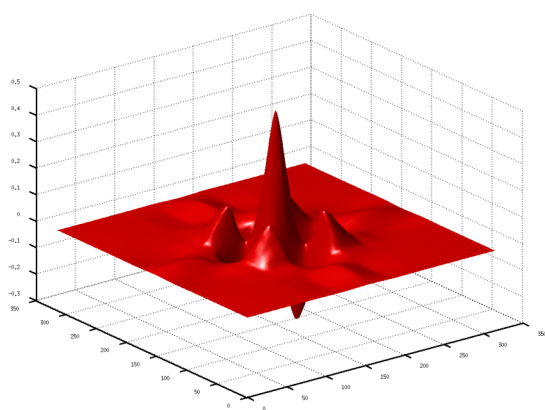


Obrázek 2.6: Na těchto obrázcích jsou demonstrovány biorthogonální vlnky CDF 9/7, které se využívají ve formátu JPEG 2000 ke ztrátové kompresi.

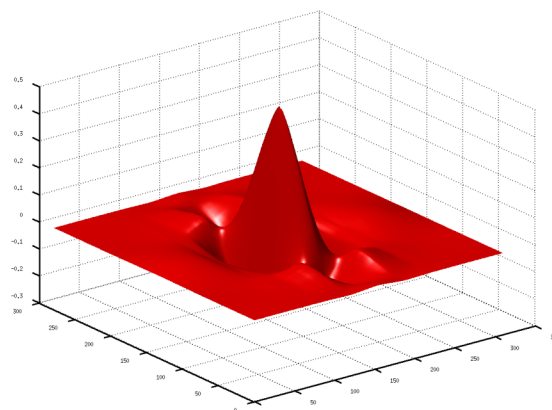


Obrázek 2.7: 2D DWT - provedení jedné úrovně rozkladu

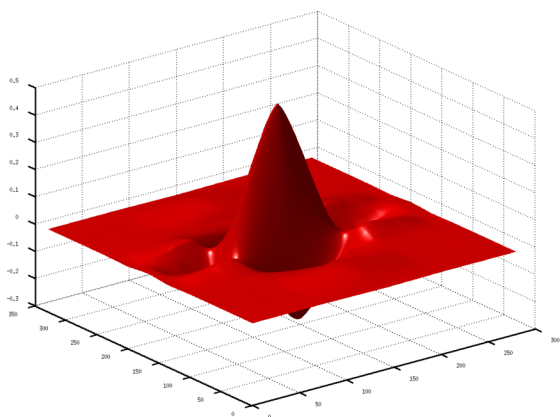
Na Obrázku 2.7 je zobrazeno, jakým způsobem probíhá jeden stupeň DWT transformace.



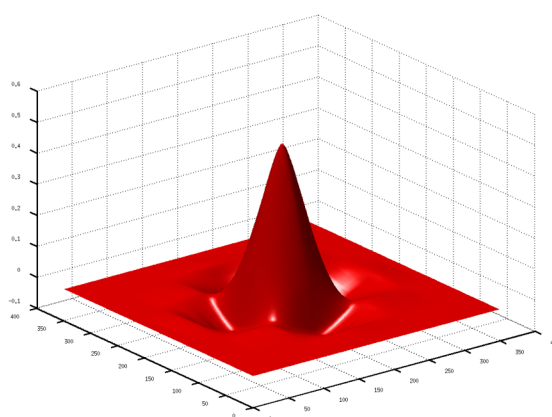
(a) HH



(b) HL



(c) LH



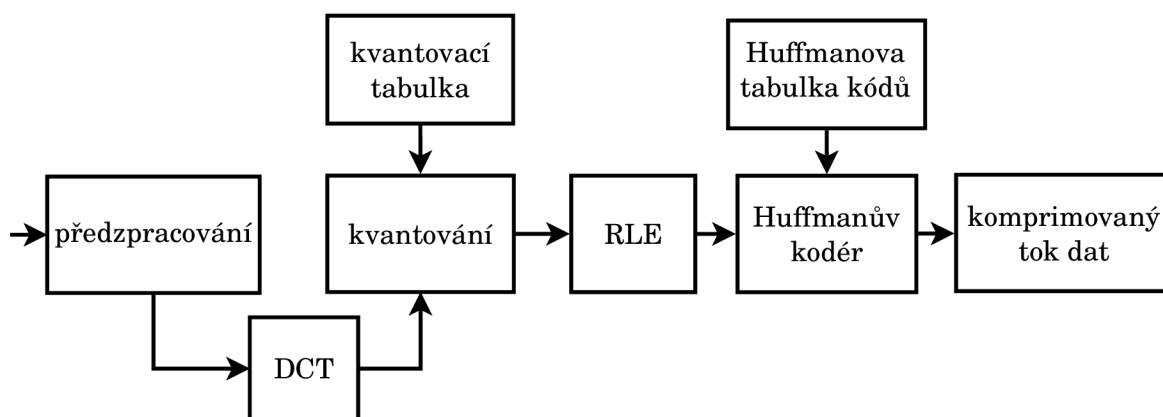
(d) LL

Obrázek 2.8: Na těchto obrázcích jsou demonstrovány 2D bazové funkce diskrétní vlnkové transformace.

2.1 JPEG

Joint Photographic Expert Group (JPEG) je názvem výboru, který vyvinul v roce 1992 standard ISO/IETC 10918-1, který je zároveň popsán i v doporučení CCITT/ITU-T T.81. JPEG je primárně určen ke ztrátové kompresi, ale podporuje i bezztrátovou variantu. JPEG může podporovat jeden ze čtyř postupů kódování, mezi které se řadí sekvenční kódování, progresivní kódování, bezztrátové kódování a hierarchické kódování. V této sekci jsem čerpal informace z materiálů [3], [9] a [15].

Tato kapitola bude zaměřena převážně na sekvenční kódování, které je založené na DCT a které je blíže specifikováno v sekci **Diskrétní kosinová transformace**. Nyní bude popsán základní proces komprimace, který je zobrazen na Obrázku 2.9.



Obrázek 2.9: Schéma toku dat u formátu JPEG

V samotném procesu komprese je nejprve provedeno předzpracování. V této části komprese se provádí transformace do barevného prostoru $YCbCr$ a případné podvzorkování složek. Tohle je více specifikováno v kapitole **Formáty ztrátové komprese obrazu**. Dále je nutné, aby každá vstupní složka byla rozdělena do bloku o rozměrech 8×8 hodnot pixelů (vzorků), které tvoří datovou jednotku.

Pokud některá složka nemá rozměry přesně v násobcích 8×8 , dojde k rozšíření na patřičnou velikost. Rozšíření se provádí duplikací nejbližšího řádku nebo sloupce.

Prioritou pro výpočet dopředné diskrétní kosinové transformace (FDCT) jsou o úroveň posunutá vstupní data přiřazená k dvojkovému doplňku. Pro 8bitový vstup je posunu o úroveň dosaženo odečtením hodnoty 128.

Po provedení posunu střední hodnoty signálu na každý blok uplatníme FDCT a získáme 64 hodnot, které jsou koeficienty DCT. V levém horním rohu na indexu $[0,0]$ se nachází koeficient DC, který vyjadřuje stejnosměrnou složku signálu. Zbývajících 63 koeficientů je označováno jako koeficienty AC (*alternating current*). Tyto koeficienty AC zde udávají váhy, které určují přítomnost dvourozměrné kosinusoidy v bloku. Při tomto postupu nedochází ke ztrátě dat.

2.1.1 Kvantování

Každý ze 64 koeficientů je kvantován na základě kvantovací tabulky Q . Každý koeficient DCT je podělen odpovídající hodnotou v kvantovací tabulce Q .

Kvantovací tabulka udává, jak velké ztráty dat se při kompresi dopustíme. Při použití vyšších hodnot v tabulce dojde k hrubšímu kvantování a tím pádem k méně kvalitnímu

obrazu. Nejvyšší hodnoty v kvantovací tabulce se nachází u vyšších frekvencí, kde se můžeme dopustit větší ztráty dat. Při výběru vhodné kvantovací tabulky je důležité zohlednit vlastnosti obrázku, zobrazovací zařízení a světelné podmínky.

Hodnoty kvantovaných koeficientů DCT jsou přiřazeny dvojkovému doplňku celého čísla s přesností 11 bitů na 8bitovou vstupní přesnost.

Pokud byla zvolena vhodná kvantovací tabulka, tak bude větší část koeficientů po kvantování nabývat hodnoty nula. Zbývající koeficienty budou soustředěny v levém horním rohu. Jedná se o koeficienty s nízkou frekvencí.

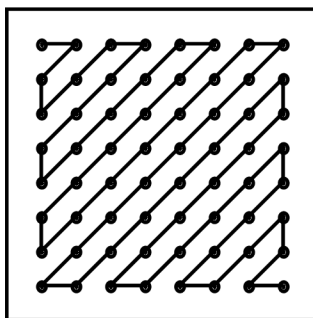
2.1.2 Entropické kódování

Po provedení kvantování je koeficient DC a 63 koeficientů AC připraveno k entropickému kódování. Koeficienty jsou reprezentovány dvojkovým doplňkem celého čísla.

Na koeficienty DC se použije diferenciální kódování. Vypočítá se rozdíl aktuální hodnoty DC a předchozí hodnoty DC a ten se následně zakóduje. Pro 63 kvantovaných koeficientů AC se žádné diferenciální kódování neprovádí, ale je na ně uplatněna linearizace pomocí zig-zag průchodu a dochází zde k převodu na jednodimenzionální sekvenci. Zig-zag průchod je zobrazen na Obrázku 2.10.

V této sekvenci se často vyskytují opakující se nulové hodnoty. Jako vhodná metoda pro zakódování opakujících se sekvencí nulových hodnot se nabízí metoda run length encoding (RLE).

Zig-zag sekvence prochází nejprve hodnoty s nižší frekvencí, jelikož budou mít větší amplitudu a je tedy větší šance, že budou nenulové. Za poslední nenulovou hodnotu se vkládá symbol end of block (EOB).



Obrázek 2.10: Zig-zag průchod

Následuje další kódovací postup, při kterém jsou data dále komprimovány. Je možné použít jednu ze dvou metod, kdy se nám nabízí metoda Huffmanova kódování a aritmetického kódování.

Aritmetické kódování je na rozdíl od Huffmanova kódování o 5–10 % účinnější, ale je výpočetně náročnější. V praxi se nejčastěji používá Huffmanovo kódování, které je možné použít v libovolném kódovacím procesu a které využívá Huffmanovu tabulku specifikací pro kompresor.

Při kódování DC koeficientů pro 8×8 bloků je možné použít až dvě DC tabulky určení. To samé platí i pro kódování AC koeficientů. Ve specifikaci nejsou uvedeny žádné výchozí tabulky, tedy pro každé kódování se sestavují nové.

Jak už bylo dříve nastíněno, při kódování DC koeficientů se nejprve vypočítá rozdíl aktuální hodnoty DC a prediktoru (nejbližší kvantovaná hodnota DC ze stejné složky).

Tato získaná hodnota se zakóduje.

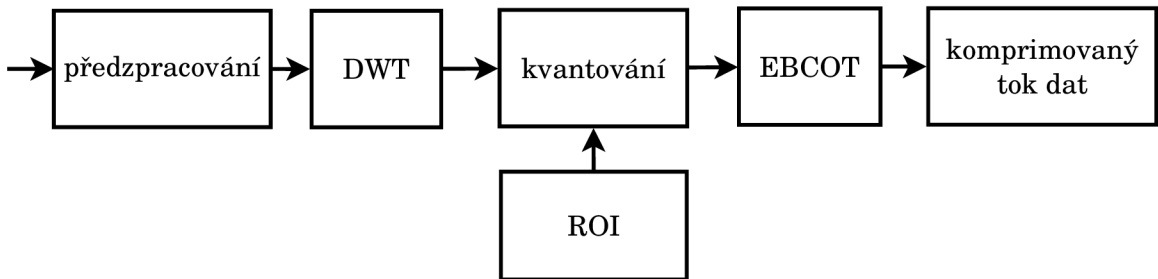
Mnoho koeficientů AC nabude po kvantování hodnoty nula. Zakódování AC koeficientů probíhá za pomoci 8bitového čísla složeného ze dvou hodnot, a to za prvé délky sekvence nul a dále nenulového koeficientu, který je bezprostředně za touto sekvencí. Výsledkem všech těchto kroků je komprimovaný tok dat.

2.2 JPEG 2000

Stejně jako v kapitole **JPEG** je Joint Photographic Expert Group (JPEG) názvem výboru, který vyvinul v roce 2000 nový formát, který nese název JPEG 2000. Jádro tohoto formátu je popsáno ve standardu ISO/IEC 15444-1 a zároveň i v doporučení ITU-T T.800. V této sekci jsem čerpal informace z materiálů [3] a [8].

JPEG 2000 měl původně nahradit JPEG, jelikož je více odolný ke vzniku chyb. Kromě toho ale JPEG 2000 přináší i další četná vylepšení. Za zmínku stojí především vyšší kompresní poměr při zachování stejné kvality obrazu, podpora alfa kanálu a mnohá další vylepšení.

Stejně jako JPEG i tento formát podporuje ztrátovou i bezztrátovou kompresi obrazu. Klíčovou metodou pro JPEG 2000 je DWT, která je blíže specifikována v sekci **Diskrétní vlnková transformace**.



Obrázek 2.11: Schéma toku dat u formátu JPEG 2000

Na Obrázku 2.11 je nastíněno, jak probíhá proces komprese. Opět se zde nejprve využívá předzpracování, do kterého je zahrnuta transformace barev do barevného prostoru $YCbCr$ a která je podrobněji popsána v kapitole **Formáty ztrátové komprese obrazu**.

Pro převod z RGB modelu se využívají dvě transformace vstupních složek. RCT (reversible color transform) transformace se používá v rámci bezztrátové komprese a naopak pro ztrátovou kompresi se využívá transformace ICT (irreversible color transform). Po převedení do barevného prostoru $YCbCr$ je možné provést podvzorkování barvonosných složek.

Dále je možné uplatnit posun střední hodnoty signálu (DC level shifting) na vzorky složek. Lze jej provést ještě před výpočtem transformace dopředných složek (RCT nebo ICT), pokud nebyly tyto transformace použity, výpočet se provede před transformací DWT.

Obraz může být rozdělen do několika dlaždic. Tyto dlaždice mají tvar obdelníkového pole stejných rozměrů, kdy tyto dlaždice obsahují stejný podíl všech složek a dohromady vytváří obraz.

Rozklad na dlaždice má několik zásadních výhod. Jednou z nich je, že se tím sníží výpočetní náročnost komprese a druhou je, že zpracování dlaždic může probíhat nezávisle na sobě. Příklady operací prováděných nad dlaždicemi mohou být extrahování, dekódování a zpětná rekonstrukce.

2.2.1 Transformace

K provedení komprese se používá dopředná diskretní vlnková transformace (FDWT) a při dekompresi naopak inverzní diskretní vlnková transformace (IDWT). Při FDWT se převádí jednorozměrná dekompozice podpásma z jednorozměrné sady vzorků na dolní propust (low-pass) koeficientů reprezentujících podvzorkování nízkého rozlišení verze původní sady vzorků.

Horní propust (high-pass) vzorků reprezentuje podvzorkování zbytkové verze původní sady vzorků, které jsou potřebné ke kvalitní rekonstrukci originální sady vzorků z dolní propusti (low-pass).

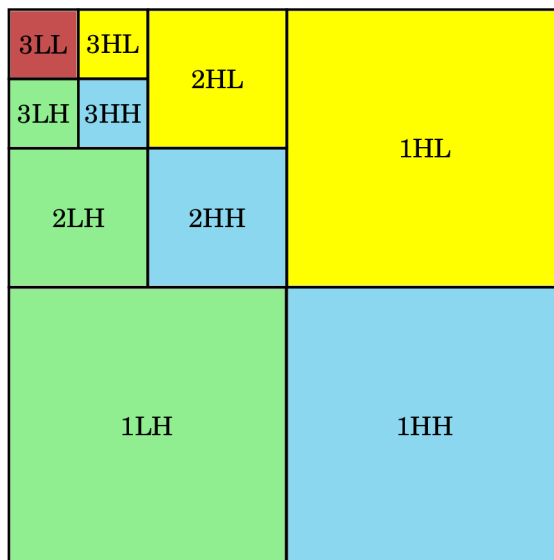
Každá složka dlaždice je transformována do sady dvourozměrných signálů podpásma. Každý rozklad reprezentuje aktivitu signálu v různých frekvenčních skupinách a prostorových rozlišeních. Jednotlivá rozlišení odpovídají jednotlivým úrovním rozkladu.

Rozklad úrovně souvisí s následujícím rozkladem úrovně o prostorové síle dva. Tento pojem znamená, že pro každý úspěšný rozklad úrovně podpásma má přibližně polovinu horizontálních a polovinu vertikálních rozlišení z předchozí úrovně. Obrazy s nižším rozlišením než je originál jsou generovány dekódováním vybrané podmnožiny těchto podpásem.

FDWT transformuje vzorky s posunutou střední hodnotou signálu (DC-level shifted) z dlaždice do sady podpásem s koeficienty. Tahle procedura závisí na parametru N_L , který reprezentuje počet iterací, též známý jako počet úrovní rozkladu.

Celkový počet podpásem je tedy $(3 \times N_L) + 1$. Každé podpásma je označeno dvěma indexy (LL, LH, HL, HH). První index značí horizontální filtrování a druhý index filtrování vertikální. Indexy tedy mohou nabývat dvou hodnot, a to horní propusti (H) a dolní propusti (L).

Na nejvyšším rozkladu úrovně N_L se nachází podpásma LL, které má nejnižší rozlišení. Tento princip je znázorněn na následujícím obrázku.



Obrázek 2.12: Každá úroveň rozkladu obsahuje 3 podpásma (HL, LH, HH), poslední úroveň se sestává z LL podpásma.

Jednotlivá podpásma jsou rozdělena na oblasti a bloky kódování.

2.2.2 Kvantování

Tato kapitola popisuje formy kvantování a dekvantizace, které jsou využívány ke kódování a rekonstrukci složek dlaždic obrazu. Při procesu kvantování dochází ke snížení přesnosti transformovaných koeficientů. Tato operace je bezztrátová, pokud hodnota kroku kvantování je rovna 1 a koeficienty jsou celými čísly. V jakékoli jiné situaci je operace ztrátová.

Pro ztrátovou kompresi se využívá vlnkový filtr 9/7 a naopak pro bezztrátovou kompresi se používá vlnkový filtr 5/3, který je možné spočítat v celočíselné aritmetice. Pro 9/7 vlnkový filtr platí, že jeho kvantovací kroky (step-sizes) pro všechny podpásma jsou získány z bitového toku.

Další důležitou součástí procesu kvantování je zesílení rozkladu (analysis gain) podpásma. Zesílení rozkladu podpásma je rekurzivně definováno jako rozklad předchozího podpásma vynásobeného odpovídajícími horizontálními a vertikálními filtry použitých k vyprodukování podpásma. Filtr dolní propusti má hodnotu rozkladu 1, zatímco filtr horní propusti hodnotu 2. Tahle hodnota určuje počet filtrovacích kroků nutných k vyprodukování podpásma.

V rámci kvantování se dále používá pojem bitová hloubka. Bitová hloubka představuje počet bitů, které se využívají k reprezentaci vzorků původního obrazu.

Kvantovací krok při ztrátové kompresi, jak bylo nastíněno výše, musí mít hodnotu vyšší než 1. Jeho výpočet se provádí pomocí nominálního dynamického rozsahu R_b , exponentu ε_b a mantisy μ_b . Pár (ε_b, μ_b) může být signalizován v syntaxi bitového toku pro každé podpásma (tzn. pro každé podpásma budou nabývat jiných hodnot), anebo může být signalizován pouze v bitovém toku podpásma LL. Tyto hodnoty si určuje kodér samostatně.

Maximální počet kódovaných bitových rovin, který lze očekávat v průchodu kódu (code stream) pro podpásma, je možné získat pomocí výše zmíněného exponentu.

2.2.3 Oblast zájmu a Maxshift

Tato kapitola popisuje oblast zájmu (ROI). Oblast zájmu (region of interest) je část obrazu, která je kódována dříve než zbytek kódu při jeho průchodu (codestream). Pro uvedení příkladu si lze ROI představit jako oblast obrazu, kde se nachází tváře osob. Rozhodně bychom chtěli, aby se tváře osob dekodovaly dříve než je tomu například u pozadí. K vyznačení oblasti zájmu se používá algoritmus Maxshift.

Kódování kvantovaných koeficientů se moc neliší od kódování oblastí bez použití ROI. Na straně kodéru je vytvořena maska popisující, které koeficienty mají být kódovány s lepší kvalitou a které mohou být dokonce kódovány zcela bezztrátově. ROI maska je bitová mapa popisující tyto koeficienty.

Kvantované koeficienty mimo ROI masku (koeficienty pozadí) jsou v rámci bitového posunu posunuty směrem dolů (scaled down) tak, že bity ROI jsou umístěny ve vyšší bitové rovině než je pozadí. To znamená, že ve chvíli, kdy entropický kodér kóduje kvantované koeficienty, tak bitové roviny související s ROI jsou kódovány přednostně (např. vzhledem k pozadí). Posun bitových rovin s (scaling value) musí být velký natolik, aby nejmenší nenulový koeficient ROI byl větší, než je největší koeficient pozadí.

Tuto metodu lze popsat následovně:

- Vygenerování ROI masky,
- Nalezení posunu s (scaling value),
- Posun všech koeficientů pozadí (v rámci bitových rovin) směrem dolů pomocí posunu s ,

- Zapsání posunu s do průchodu kódu.

Po provedení těchto 4 kroků jsou kvantované koeficienty entropicky kódovány.

Po operaci snížení (scaling operation) je počet kódovaných bitových rovin zvýšen o posun s .

Posun s musí být vybrán tak, aby platila rovnice 2.8, kde $\max(M_b)$ značí největší počet magnitud bitových rovin.

$$s \geq \max(M_b) \quad (2.8)$$

Na straně dekodéru jsou přijaté koeficienty porovnávány s prahovou hodnotou (threshold value) 2^s , kde s vyjadřuje posun pro tuto složku v rámci průchodu kódu (codestream). Všechny koeficienty nižší než je hodnota 2^s patří do pozadí.

2.2.4 Entropické kódování

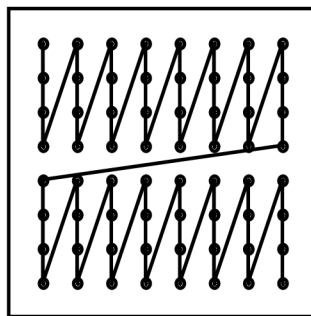
Tato kapitola je věnována entropickému kódování. Tento formát využívá aritmetický kódér MQ a kódovací algoritmus EBCOT.

Podpásma jsou za účelem kódování koeficientů algoritmem EBCOT záměrně rozděleny do obdelníkových oblastí, které byly již výše zmíněny a které nazýváme bloky kódování (code blocks). Rozměry takového bloku jsou vždy v mocninách 2.

EBCOT kóduje tyto koeficienty bloků do toku bitů. Kódování se zde provádí podle významnosti bitu po bitových rovinách (od MSB k LSB). Kódování bitových rovin by se dalo shrnout do tří průchodů, mezi které řadíme propagaci významnosti, upřesnění magnitudy a úklid (cleanup).

Při propagaci významnosti se kódují bity takových koeficientů, které mají za sousedy koeficienty s významnými bity. U těchto koeficientů se předpokládá, že i jejich bity se stanou významnými. Při upřesnění magnitudy dochází k upřesnění bitů koeficientů, které již dříve byly významnými. Na závěr se při úklidu zakódují všechny ostatní bity, které zbyly.

V rámci bitové roviny jsou bity zpracovávány způsobem, který je znázorněn na následujícím obrázku:



Obrázek 2.13: Pořadí použité pro kódování DWT v algoritmu EBCOT

Všechna zpracovaná data je na závěr nutno uložit do datového toku. Datový tok se dělí na takzvané vrstvy (layers). Důvodem tohoto rozdělení je postupné zvyšování kvality obrazu, a to s každou další dekódovanou vrstvou. Každá vrstva navíc obsahuje další bity získané z bloku kódování. Všechna komprimovaná data jsou ve finále zakomponovány do paketů.

Paket je spojitým segmentem, který v sobě zahrnuje data o konkrétní dlaždici, vrstvě, barevné složce, rozlišení a pozici, kde se nachází. Pořadí paketů je v datovém toku udáváno na základě 4 parametrů, mezi které patří vrstva (určuje kvalitu), barevná složka, rozlišení a oblast (určuje pozici). Pořadí parametrů se v průběhu datového toku může měnit!

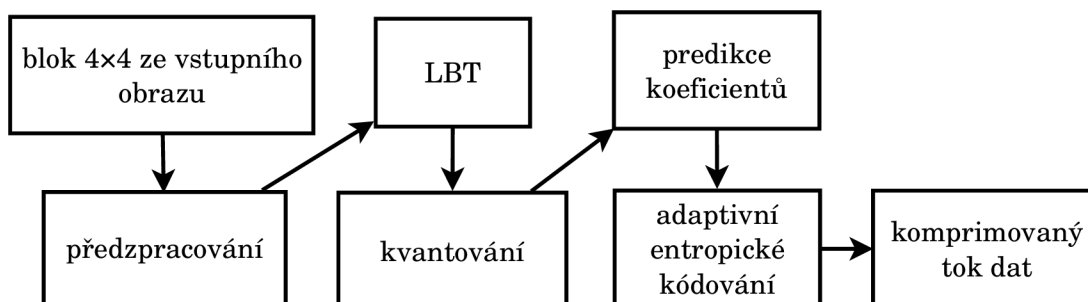
2.3 JPEG XR

V této kapitole bude vysvětlen základní princip formátu JPEG XR, jehož základ je podrobně popsán ve standardu ITU-T T.832. V této sekci jsem čerpal informace z materiálů [10] a [11].

JPEG XR efektivněji využívá komprimovaný snímek a přidává rozšířenou rozmanitost (broadened diversity) pro splnění náročných požadavků aplikace. JPEG XR podporuje širokou paletu barevných kódování formátů (černobílá, RGB, CMYK) a kódování velkého množství různých číselných hodnot s různou bitovou hloubkou. Nově obsahuje podporu pro aplikace s vysokým obrazovým rozsahem (high dynamic range) (HDR).

JPEG XR kombinuje výhody optimalizované kvality obrazu a efektivnosti komprese spolu s nízkou náročností požadavků na implementaci kódování a dekódování. Takového výsledku by nebylo možné docílit bez zavedení nových funkcí, jako příklad lze uvést vyšší schopnost komprese, nízké výpočetní a paměťové nároky, segmentace obrazu pro náhodný přístup velkých obrazových formátů, podpora alfa kanálu a mnoho dalších funkcí.

Stejně jako JPEG a JPEG 2000 i tento formát podporuje ztrátovou i bezztrátovou kompresi obrazu. Klíčovou metodou pro JPEG XR je transformace LBT (lapped biorthogonal transform).



Obrázek 2.14: Schéma toku dat u formátu JPEG XR

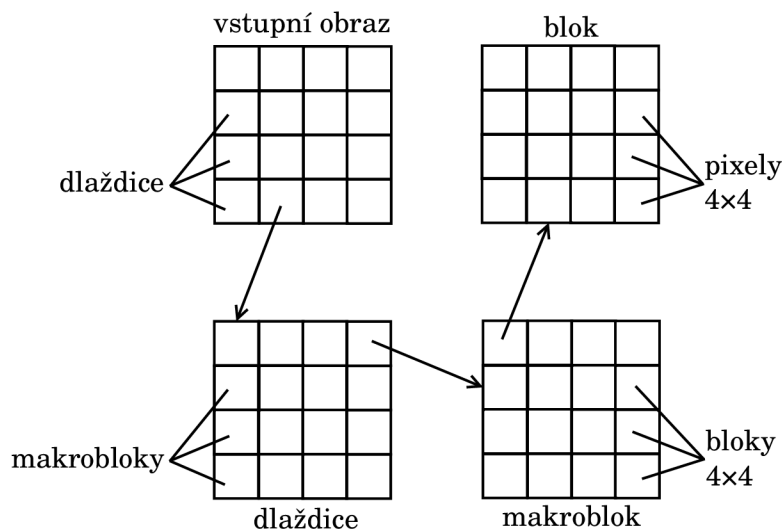
Kompresní proces se skládá z několika kroků, které je možné vidět na Obrázku 2.14. Nejprve se provádí předzpracování, jehož součástí je operace předškálování (pre-scaling) 16bitových a více bitových hodnot (celočíselných i s plovoucí řádovou čárkou).

Po předškálování hodnot se provádí převod barevného modelu (color conversion). Převod lze provádět z velkého množství různých variant modelů RGB a CMYK (BGR24, RGB48 atd.) do barevného modelu YUV. Pokud nemáme v plánu podvzorkovávat barvosložky, zvolíme formát obrazu YUV444. Naproti tomu, pokud vyžadujeme podvzorkování barvosložek, použijeme formát obrazu YUV420 nebo YUV422. Více informací o podvzorkování složek je uvedeno v kapitole **Formáty ztrátové komprese obrazu**. Pokud chceme barevné složky úplně eliminovat, použijeme formát YONLY, který využívá pouze jasovou složku Y.

Následujícím krokem kompresního procesu je zarovnání a odsazení do makrobloku. Tohle je nezbytné ve chvíli, kdy výška a šířka obrazu není beze zbytku dělitelná šestnácti. V takovém případě rozšíříme obraz na rozměry, které jsou násobky šestnácti. Rozšíření lze

provést zduplikováním řádku (horizontal replication) spodní části a pravého okraje vzorků hodnot.

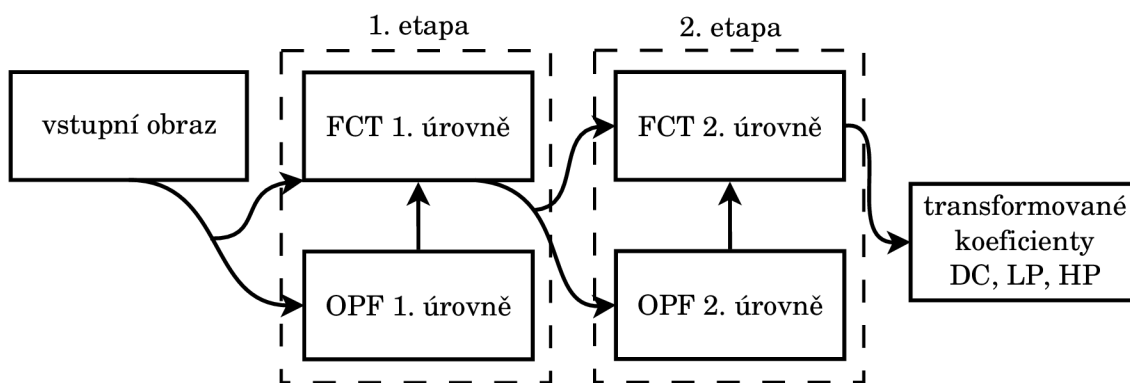
Nyní bude popsáno, jak se rozděluje obraz za účelem efektivního zpracování. Obraz se rozděluje do dlaždic, kde každá dlaždice se skládá z makrobloků. Samostatný makroblok je tvořen z 16 bloků, kde každý blok se skládá z 16 obrazových pixelů. Lépe tohle rozdělení lze pochopit z Obrázku 2.15.



Obrázek 2.15: Bližší pohled na rozdělení obrazu. Je třeba si uvědomit, že pevnou velikost má pouze blok (4×4 pixely) a makroblok (4×4 bloky). Detailnější pohled na makroblok je zobrazen na Obrázku 2.18.

2.3.1 Transformace

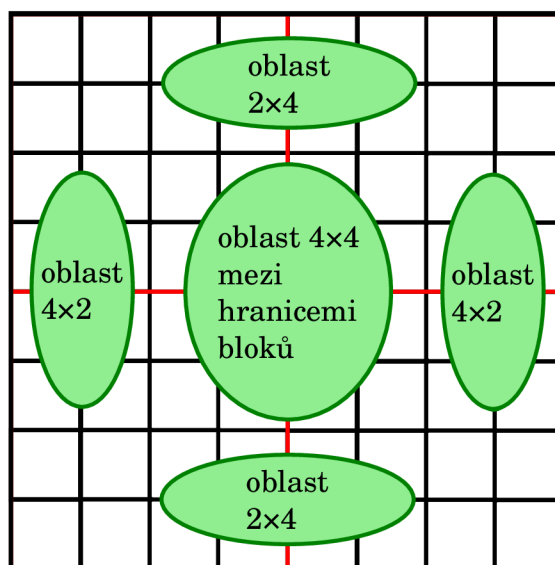
Srdcem kompresního procesu je transformace. JPEG XR využívá transformaci LBT [14] (lapped biorthogonal transform), která transformuje pixely z prostorové domény (spatial domain) do frekvenční domény (frequency domain). LBT se skládá ze dvou etap, kdy každá etapa se skládá z OPF (Overlap Pre Filter) a FCT [6] (Forward Core Transform).



Obrázek 2.16: Detailnější pohled na transformaci LBT. LBT se skládá ze dvou etap, kdy každá z etap se skládá z volitelného předfiltrování OPF a povinné transformace FCT. Každá transformace FCT se rovněž skládá ze dvou etap, kdy každá zahrnuje čtyři 2×2 transformace.

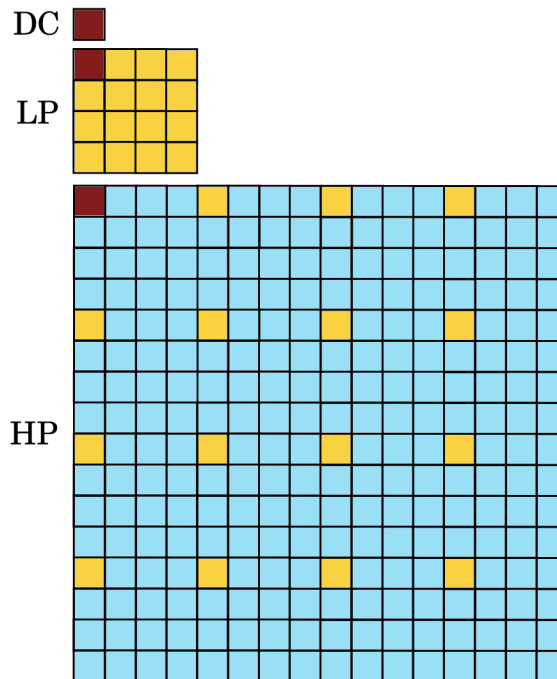
Průběh transformace LBT:

- Jak je patrné z Obrázku 2.16, operace předfiltrování není nezbytně vyžadována. První úroveň operace předfiltrování OPF se uplatňuje na hranice mezi bloky o oblasti 2×4 a 4×2 a do čtyř rohů o velikosti 2×2 . Volitelně může být využita i na středovou oblast mezi makrobloky o velikosti oblasti 4×4 . Pro lepší pochopení jsou tyto oblasti znázorněny na Obrázku 2.17.
- Nyní se uplatní FCT na bloky o velikosti 4×4 , tedy na celý makroblok. Tato operace završuje první etapu LBT transformace.
- Dále je nutné získat DC-LP pole. Každý ze 4×4 bloků obsahuje DC koeficient. Z tohoto vyplývá, že na jeden makroblok připadá 16 DC koeficientů. Tyto DC koeficienty reprezentují barevnou složku makrobloku a jsou strukturovány do bloku o velikosti 4×4 a tento blok nazýváme DC-LP pole.
- Předfiltrovací operace ve druhé etapě je opět volitelná. Druhá úroveň operace předfiltrování OPF se uplatňuje na hranice mezi bloky o oblasti 2×4 a 4×2 a do čtyř rohů o velikosti 2×2 s tím rozdílem, že tentokrát se pracuje s DC-LP polem (4×4 DC koeficientů).
- Znovu se uplatňuje FCT, tentokrát na blok o velikosti 4×4 , který odpovídá DC koeficientům v makrobloku. Tímto je ukončena druhá etapa LBT transformace.



Obrázek 2.17: Oddíly oblastí mezi hranicemi bloků. Na obrázku se nachází 4 bloky o velikosti 4×4 , které jsou odděleny červenou čarou (hranicí mezi bloky). Zelené bubliny znázorňují velikosti a umístění oblastí, na které se uplatní operace předfiltrování (OPF). Výsledkem této operace je potlačení artefaktů na hranicích mezi bloky.

Výsledkem LBT transformace jsou transformované makrobloky frekvenční domény, které jsou složeny z jednoho DC koeficientu, 15 LP koeficientů dolní propusti (low-pass) a 240 koeficientů horní propusti (high-pass).



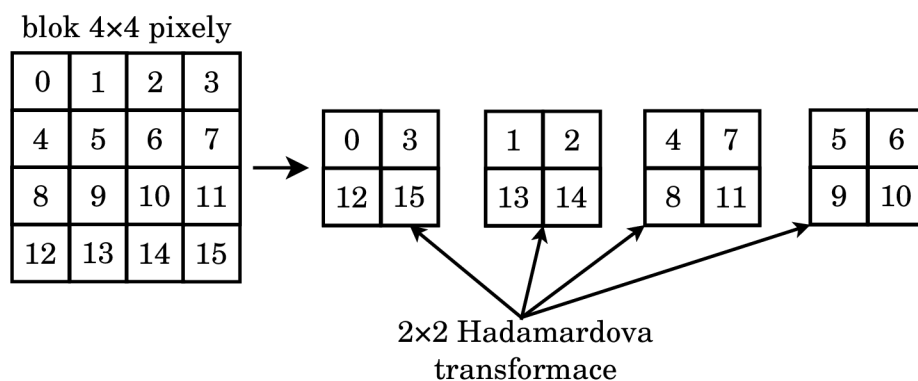
Obrázek 2.18: Frekvenční hierarchie makrobloku

K LBT transformaci je třeba dodat, že operace předfiltrování OPF se uplatňuje nejen na hranice mezi makrobloky, ale volitelně ji lze uplatnit i na hranice mezi dlaždicemi. Tato operace sice zvyšuje výpočetní náročnost komprese, ale je nutná k většímu potlačení artefaktů, které vznikají na hranicích mezi makrobloky.

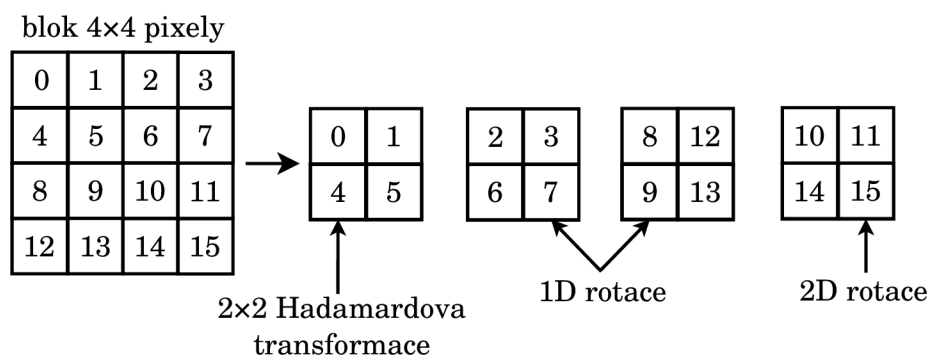
Operace FCT má velmi podobnou funkci jako **Diskrétní kosinová transformace** ve formátu **JPEG**. Výhodou FCT oproti DCT je nižší výpočetní náročnost operace. FCT využívá 3 transformační operátory (2×2 Hadamardova transformace, 1D rotace a 2D rotace) a permutační funkci. Vstupem operace FCT je blok o velikosti 4×4 pixely. FCT se rovněž jako samotná LBT skládá ze dvou etap transformace (transform stages). První etapa FCT je zobrazena na Obrázku 2.19 a druhá etapa je na Obrázku 2.20. Každá etapa zahrnuje čtyři 2×2 transformace, které mohou být prováděny v libovolném pořadí anebo dokonce současně v rámci jedné etapy. V případě, že jsou transformace prováděny paralelně v rámci jedné etapy je nezbytné, aby první etapa skončila dříve, než bude zahájena druhá etapa. Po provedení obou etap je nutné koeficienty seřadit, k čemuž nám slouží permutační funkce.

Nyní opustíme transformaci a přesuneme se k další části kompresního procesu, a tím je kvantování. V průběhu kvantování dochází k dělení transformovaných koeficientů hodnotou kvantovacího měřítka (quantization scaling factor). Výsledné hodnoty jsou zaokrouhleny na celá čísla, která nazýváme kvantované hodnoty. Pro bezztrátovou kompresi je kvantovací měřítka nastaveno na hodnotu 1 a pro ztrátovou kompresi jej musíme nastavit na hodnotu vyšší než je 1. Použité kvantovací měřítka ovlivňuje proces kvantování. Vyšší hodnoty kvantovacího měřítka mají za následek větší úroveň kvantování, a tím dochází k nižší kvalitě výsledného obrazu. Kvantovací měřítka může být různé pro jednotlivá pásma (HP, LP, DC). Index kvantovacího parametru DC je pevně daný, ale může se lišit napříč dlaždicemi, tato skutečnost pak musí být popsána v průchodu kódu (codestream).

Další částí kompresního procesu, která bude podrobněji popsána, je predikce koeficientů, která ovšem není nedílnou součástí specifikace. Jsou definovány čtyři režimy pro predikci DC



Obrázek 2.19: Na blokovém diagramu je znázorněna první etapa operace FCT, ve které se v jednom transformovaném bloku používá čtyřikrát 2×2 Hadamardova transformace. Tyto transformace mohou být prováděny paralelně, ale je velmi důležité, aby první etapa skončila dříve než započne druhá etapa, která je zobrazena na Obrázku 2.20.



Obrázek 2.20: Na blokovém diagramu je znázorněno, na které pixely se v druhé etapě operace FCT uplatňují transformační operátory 2×2 Hadamardova transformace, 1D rotace a 2D rotace. I v této etapě je možné provádět transformace paralelně.

koeficientů v makrobloku. Jedná se o predikci shora (predict from top), predikci zleva (predict from left), predikci zleva i shora (predict from left and top) a žádnou predikci (no prediction). Predikční režim určujeme z polohy makrobloku, jelikož očekáváme DC koeficienty v levém horním rohu makrobloku. DC koeficienty, jak už bylo výše nastíněno, nesou informaci o barevné složce, pokud se tedy jedná o barevný obraz. Po predikci koeficientů se provádí jejich skenování.

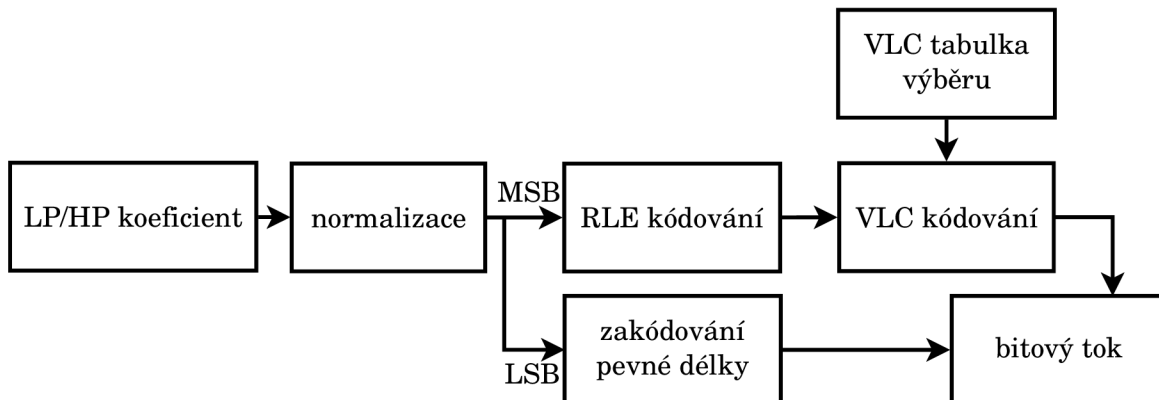
Finálním krokem kompresního procesu je adaptivní entropické kódování. Využívá se zde kódování VLC (variable-length coding) a RLE (run level encoding), jelikož mnoho koeficientů bude nabývat hodnoty nula po procesu kvantování.

Transformované koeficienty (DC, LP a HP) jsou rozdělovány pomocí normalizace na kódované části s pevnou délkou (FLC) a proměnlivou délkou (VLC) např. podle (nejméně, nejvíce) významného bitu.

Kódovaná část pevné délky složená pouze z koeficientů DC a LP se nazývá FLC vylepšení (refinement) a kódovaná část pevné délky složená z koeficientů HP se nazývá pružné bity (flexbits). Tyto bity se přímo zakódují do bitového toku.

Nyní přejdeme ke kódování s proměnlivou délkou. Koeficienty DC jsou VLC kódovány a zapsány do bitového toku. Jelikož makroblok obsahuje pouze jediný DC koeficient, nemá smysl uvažovat o RLE kódování. Náročnější kódování se provádí u koeficientů LP a HP.

Na tyto koeficienty je nejprve uplatněno RLE kódování a následně VLC kódování a nakonec se provede zápis do bitového toku. VLC kódování je náročnější než u DC koeficientů a využívá se zde pomocná VLC tabulka výběru. Výstupem entropického kódování je komprimovaný datový tok.

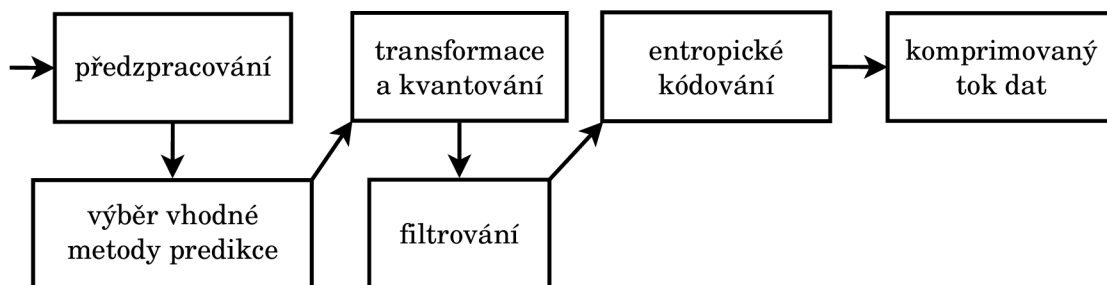


Obrázek 2.21: Zjednodušená ukázka adaptivního entropického kódování HP a LP koeficientů

2.4 WebP

V této kapitole bude probrán základní princip obrazového formátu WebP, který je založen na kompresním formátu videa (VP8). Princip kompresního formátu VP8 je popsán v dokumentu RFC 6386 a zároveň je na něm postaven i video formát WebM, který je spolu s obrazovým formátem WebP vyvíjen firmou Google. V této sekci jsem čerpal informace z materiálů [1] a [2].

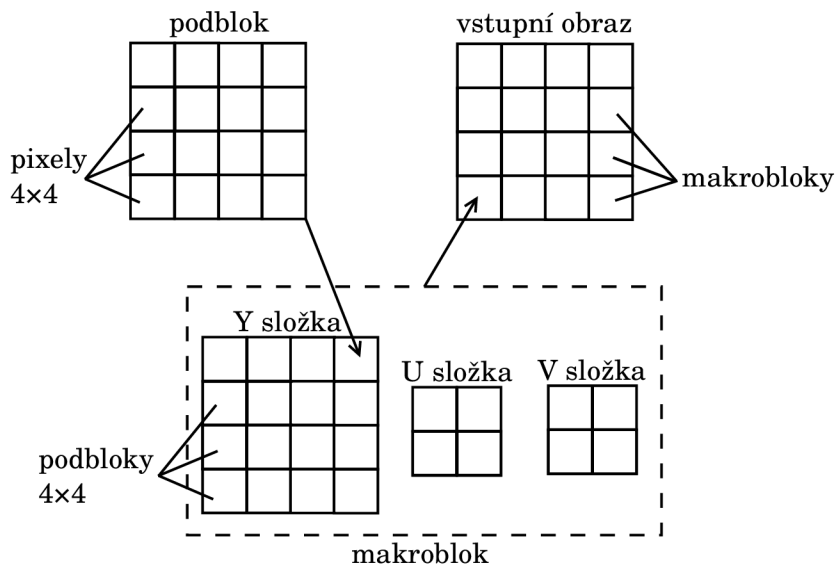
WebP podobně jako ostatní formáty pro ztrátovou kompresi usiluje o nahrazení formátu JPEG a konkrétně se snaží uspět na poli různých webových služeb. Mezi výhody VP8 se řadí propojení vysoké kompresní efektivity s nízkou dekodovací složitostí (complexity). WebP slibuje snížení paměťové náročnosti komprimovaného obrazu při stejné kvalitě o více než dvakrát oproti formátu JPEG, což potvrdila i řada testů. Na Obrázku 2.22 je znázorněno, jak přibližně probíhá kompresní proces u formátu VP8.



Obrázek 2.22: Schéma toku dat u formátu VP8

Nyní bude podrobněji popsán kompresní proces formátu VP8. Nejprve se provádí předzpracování, jehož součástí je převod do barevného modelu YUV. VP8 podporuje pouze formát obrazu YUV420. Více informací o barevných modelech a podvzorkování složek je uvedeno v kapitole [Formáty ztrátové komprese obrazu](#).

Vstupní obraz je rozdělen do pole makrobloků (macroblocks). Makrobloky jsou čtvercové oblasti pixelů, kde jasová složka Y je oblast o velikosti 16×16 pixelů a barevné složky U a V jsou oblasti o velikosti 8×8 pixelů. Makrobloky jsou dále rozdělovány do podbloků (subblocks) o velikosti 4×4 pixely za účelem lepšího zpracování v následujících procesech komprese. Makroblok se poté sestává z 16Y, 4U a 4V podbloků. Rozdělení obrazu na makrobloky a jejich podbloky je vyobrazeno na Obrázku 2.23.



Obrázek 2.23: Bližší pohled na rozdělení obrazu. Je třeba si uvědomit, že pevnou velikost má pouze podblok (4×4 pixely) a 3 složky, kterými je reprezentován makroblok (složka Y 4×4 podbloky a složky U a V 2×2 podbloky).

Velmi důležitou součástí a zároveň důvodem, proč je VP8 tak efektivní při kompresi, je zvolení vhodné predikce. VP8 používá dva typy snímků predikce, a to intra-snímek (intraframe) a inter-snímek (interframe). U intra-snímku je snímek dekódován bez jakéhokoli odkazu na jiný snímek a zároveň obsahuje podporu pro náhodný přístup k bodům ve video (datovém) toku (popř. jejich vyhledávání).

U inter-snímku je snímek dekódován s odkazem na všechny dřívější snímky, které jsou odvozeny od nějakého intra-snímku. Bez intra-snímku není možné správně dekódovat inter-snímky, jelikož, jak již bylo nastíněno, je tento algoritmus velmi citlivý na případné zahazování snímků.

U intra-snímku lze na bloky (4×4 , 8×8 a 16×16 pixelů) uplatnit několik režimů predikce:

- Horizontální predikce, při které vyplňujeme pixely ve sloupci kopií z levého sloupce (L).
- Vertikální predikce, při které získáváme pixely na řádku kopií z horního řádku (A).
- DC predikce, při které určíme průměrnou hodnotu pixelu z horního řádku (A) a levého sloupce (L).
- Speciální algoritmus predikce bloků (truemotion prediction), při kterém se výpočet nového pixelu provádí navíc ještě pomocí pixelu umístěného vlevo nahoře (C) od nově rekonstruovaného bloku. Výpočet se provádí na základě vzorce: $X_{ij} = L_i + A_j - C$.

VP8 nejvíce využívá právě tuto predikci, díky které se dosahuje vysoké efektivity komprese.

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 2.24: Znázornění speciální predikce bloků (truemotion prediction) u intra-snímku.

U inter-snímku se vychází z několika předešlých snímků a k rekonstrukci nového snímku se dále využije i pohybový vektor (motion vector), který se používá především pro vyhledání pozice kopírovaného podbloku, kdy je schopen jej vyhledat s přesností na čtvrt pixelu.

nový	levý	levý	levý
horní	nový	levý	horní
horní	horní	levý	horní
nový	levý	levý	horní

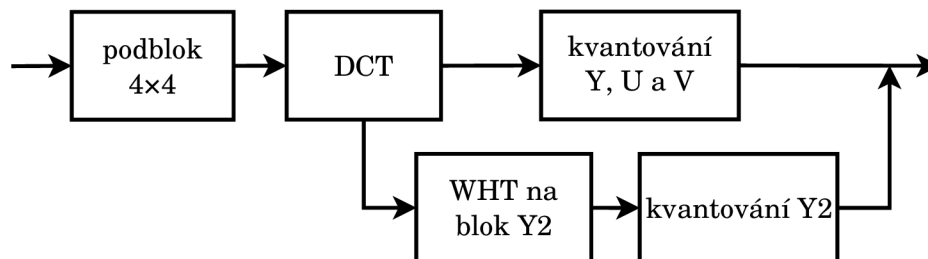
Obrázek 2.25: Znázornění predikce u inter-snímku. Nový blok znázorňuje blok 4×4 kódovaný pomocí nového pohybového vektoru a bloky levý a horní využívají pohybový vektor z bloku nový. Dohromady schéma tvoří makroblok 16×16, který je rozdělen do tří částí s třemi pohybovými vektory.

Následuje další kompresní proces a tím je transformace a následné kvantování. Na 24 reálných podbloků (16 podbloků jasové složky a 8 podbloků barevných složek) o velikosti 4×4 pixely je uplatněna diskretní kosinová transformace (DCT). Touto transformací získáváme transformované koeficienty DC a AC. Diskretní kosinová transformace je popsána podrobněji v sekci **Diskretní kosinová transformace**.

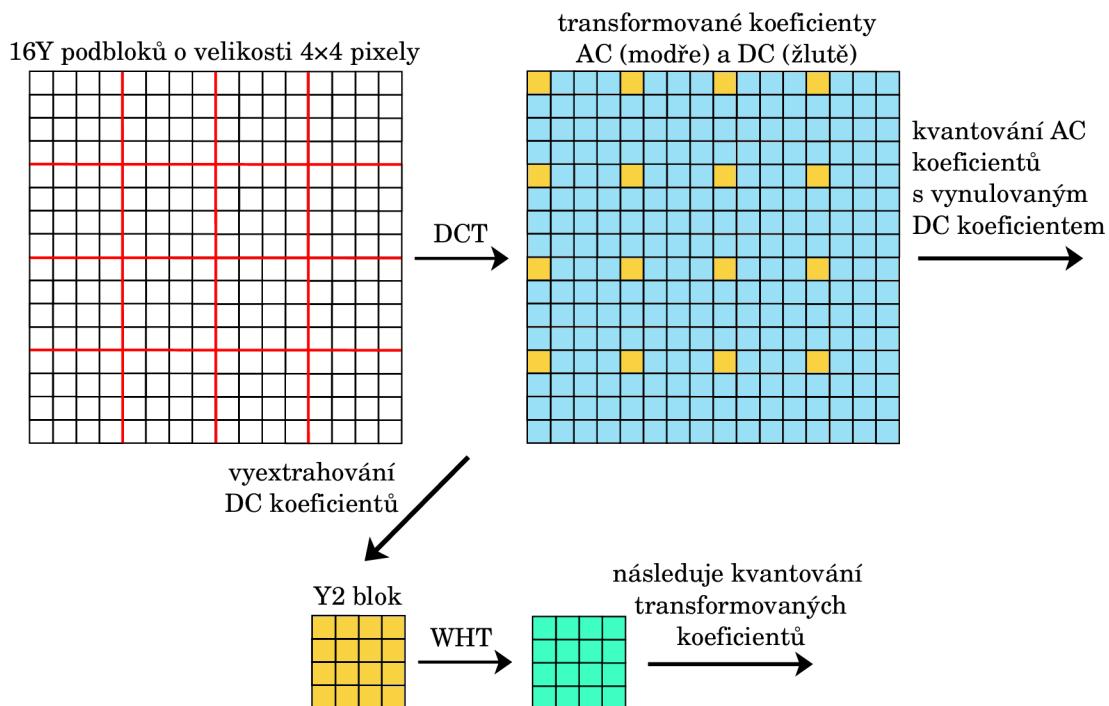
Po provedení DCT je nutné z každého z 16 jasových podbloků 4×4 vyextrahovat DC koeficient (na každý podblok připadá právě jeden), ze kterých se vytvoří nový virtuální blok Y2 o rozměrech 4×4, zatímco na koeficienty AC je již možné uplatnit kvantování, přičemž DC koeficienty neuvažujeme (nastaví se na hodnotu nula).

Nyní je nutné zpracovat blok Y2 (DC koeficienty z první transformace). Na tento blok se uplatní dopředná transformace WHT za účelem snížení redundance DC koeficientů, kdy poté získáváme 4×4 transformované koeficienty, které je nutné kvantovat. Úroveň kvantování určuje vhodné zvolení kvantovacího měřítka (quantization factor), které je různé

pro kvantování koeficientů v rovinách (Y, Y2, U a V) a pomocí kterého jsou kvantované koeficienty děleny. Pro zvolení vhodného měřítka lze použít kvantovací tabulky.



Obrázek 2.26: Detailnější pohled na transformaci a kvantování koeficientů. Nejprve se na 24 reálných podbloků (16 podbloků Y složky, 8 podbloků U a V složek) uplatní transformace DCT. Získáme koeficienty DC a AC, kdy AC koeficienty s vynulovaným DC koeficientem jsou přímo kvantovány a na DC koeficienty 4×4 (vyextrahované z 16 Y podbloků) se uplatní transformace WHT a následně kvantování. Grafické znázornění zpracování 16 Y podbloků je zobrazeno na Obrázku 2.27.



Obrázek 2.27: Zde je graficky znázorněno zpracování 16 Y podbloků při transformaci. Nejprve se zvlášť transformují všechny podbloky o velikosti 4×4 pixely pomocí DCT. Následně získáme 16 DC a 240 AC transformovaných koeficientů. Vyextrahujeme DC koeficienty a umístíme je do bloku Y2 o rozměrech 4×4 . AC koeficienty získané z DCT transformace spolu s vynulovanými DC koeficienty je nyní možné kvantovat. Vytvořený Y2 blok transformujeme pomocí WHT za účelem snížení redundance DC koeficientů a následně tyto transformované koeficienty kvantujeme.

Další část kompresního procesu se stará o potlačení blokových artefaktů použitím loop filtru (loop filter). Existují dva typy tohoto filtru:

- Normální filtr,
- Jednoduchý filtr.

Pokud vyžadujeme rychlé dekódování obrazu, je možné filtr úplně přeskočit ovšem za významného snížení kvality obrazu. Dále je možné si zvolit, jak moc efektivní filtr bude na snímky aplikován. Normální filtr je vhodné uplatnit na intra-snímky a na inter-snímky lze použít jednoduchý filtr, který je méně kvalitní a aplikuje se pouze na jasové složky.

Stejně tak lze nastavit několik úrovní filtrování na specifické oblasti obrazu. Podobně jako je tomu u ROI oblastí ve formátu JPEG 2000 lze nastavit, že na oblasti, kde se nachází tváře osob, bude uplatněn silnější filtr než na oblasti s okolním prostředím. Je nutné rozumně vybírat, které oblasti budou filtrovány, protože se jedná o výpočetně nejnáročnější aspekt formátu VP8.

Nyní se dostáváme k finálnímu procesu komprese a tím je Booleovské entropické kódování (boolean entropy encoding), které využívá aritmetický kódér. Tento aritmetický kódér kóduje vždy jednu logickou hodnotu (0 nebo 1) ve stejném čase za účelem bezztrátové komprese s využitím pravděpodobnosti. Zdali kódovaná hodnota bude nabývat hodnoty 0 nebo 1, je možné odhadnout např. pomocí konstant anebo dřívějšího chování kódovaných dat.

Pro snazší binární zakódování symbolů se využívá abstraktní Huffmanův strom. Lze jej uplatnit na libovolnou hodnotu (vektory pohybu, kvantované a transformované koeficienty, typy makrobloku atd.). Zakódování se provádí bitovou sekvencí, která se generuje od kořenového uzlu k odpovídajícímu listovému uzlu, čímž data zůstávají konzistentní (v procesu komprese i dekomprese).

Všechna data jsou nakonec rozdělena do několika oddílů, které je možné efektivně zpracovávat paralelně a využít tak výpočetní výkon vícejádrových procesorů (především se jedná o dekódování). Dohromady tyto oddíly tvoří komprimovaný datový tok, který je finálním výstupem entropického kódování.

Kapitola 3

Srovnání formátů

V této kapitole budou popsány metriky, které se využívají pro měření kvality komprimovaného obrazu. Dále zde bude provedeno teoretické a praktické srovnání ztrátových formátů popsaných v kapitole [Formáty ztrátové komprese obrazu](#).

3.1 Metriky srovnání

Ztrátové formáty dat snižují kvalitu obrazu, protože využívají techniky komprimace dat, které jsou ztrátové a tudíž je nelze zpětně zrekonstruovat. Z tohoto důvodu je nutné posuzovat výslednou kvalitu komprimovaného obrazu.

Lze tak učinit na základě subjektivního pozorování, tedy člověk subjektivně posoudí, jak se mu daný obraz líbí a zhodnotí jej. Druhou možností je zhodnotit kvalitu objektivně, tedy počítač na základě matematických metod určí kvalitu obrazu. Zástupci těchto pozorovacích metod jsou metody PSNR a SSIM.

3.1.1 PSNR

PSNR (peak signal-to-noise ratio) je poměr mezi maximální možnou hodnotou signálu a síly zkreslení hluku, který má vliv na kvalitu signálu. Kvůli velkému dynamickému rozsahu signálů se PSNR vyjadřuje v decibelech. V této sekci jsem čerpal informace z materiálu [13].

Jak již bylo nastíněno, zlepšování vizuální kvality digitálního obrazu často bývá subjektivní a každý člověk jej bude vnímat jinak. Z tohoto důvodu je nutné zavést měřítka kvality v podobě sady testů, které lze snadno uplatnit na libovolný obraz. Měření závisí na třech faktorech špička-signal-hluk (peak-signal-noise).

Pro následující příklad předpokládáme, že vstupní data, jak pro původní obraz, tak pro komprimovaný obraz, jsou ve tvaru dvourozměrné matice o totožných rozměrech. Matematické vyjádření PSNR pak vypadá následovně:

$$\text{PSNR} = 20 \log_{10} \left(\frac{MAX_f}{\sqrt{\text{MSE}}} \right), \quad (3.1)$$

kde MSE (Mean Squared Error) je:

$$\text{MSE} = \frac{1}{mn} \sum_0^{m-1} \sum_0^{n-1} \|f(i, j) - g(i, j)\|^2 \quad (3.2)$$

V rovnici 3.1 vyjadřuje hodnota f matici dat původního obrazu a hodnota g matici dat komprimovaného obrazu. Hodnota m reprezentuje počet řádků matice a i index řádků matice. Hodnoty n a j jsou analogické k m a n , vyjadřují počet sloupců a index sloupce. V rovnici 3.2 hodnota MAX_f značí maximální hodnotu signálu v původním obraze.

3.1.2 SSIM

SSIM (structural similarity index) je vylepšená metoda k ohodnocování kvality obrazu. Vytváří tři podmínky, kdy SSIM index odhaduje vizuální dopad posuvů v jas (luminiscence) obrazu, změny kontrastu (contrast) a ostatní chyby označované jako změny strukturální (structural). V této sekci jsem čerpal informace z materiálů [5] a [7].

Tato metrika je založena na předpokladu, že lidský zrakový systém HVS (human vision system) je adaptován na extrahování strukturální informace ze scény a tudíž by strukturální podobnost mohla být dobrou aproximací vnímané kvality obrazu.

Algoritmus SSIM posuzuje tři výše zmíněné podmínky mezi dvěma signály (původním a kódovaným) označenými x a y . Jedná se o jas $l(x, y)$, kontrast $c(x, y)$ a strukturu $s(x, y)$.

$$\begin{aligned} l(x, y) &= \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \\ c(x, y) &= \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \\ s(x, y) &= \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}, \end{aligned}$$

kde $C_1 = (K_1L)^2$, $C_2 = (K_2L)^2$ a $C_3 = C_2/2$. V těchto vyjádřeních hodnota L značí dynamický rozsah hodnot pixelu a K_1 spolu s K_2 vyjadřují skalární konstanty. Konstanty C_1 , C_2 a C_3 maskují vlastnosti (masking properties) a zároveň zajišťují stabilitu.

Výsledný SSIM index získáme kombinací výše zmíněných 3 podmínek:

$$\begin{aligned} SSIM(x, y) &= [l(x, y)] [c(x, y)] [s(x, y)] \\ SSIM(x, y) &= \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \end{aligned}$$

3.2 Vlastnosti formátů

V této sekci bude provedeno teoretické srovnání ztrátových formátů, které jsou popsány v kapitole **Formáty ztrátové komprese obrazu**.

Nejdříve se zaměříme na formát **JPEG**, který je již přes 20 let nejpoužívanějším ztrátovým formátem na světě. Jedním z důvodů je jistě ten, že je prakticky zdarma a podporuje jej snad každý grafický nástroj (GIMP, Adobe Photoshop, Corel atd.) a HW zařízení pracující s fotografiemi či jinou formou grafiky (fotoaparát, videokamera atd.). JPEG je vhodné použít pro ztrátovou kompresi fotografických snímků reálného okolí a obrázků s hladkými přechody barev. Naopak je velmi nevhodné jej použít na obrázky s ostrými hranami a případným textem. V takových oblastech je totiž náchylný ke vzniku artefaktů. JPEG tedy není vhodný pro komprimaci vektorové grafiky. Mezi klady jistě patří, že v základním režimu (baseline) ke komprimaci a dekomprimaci vyžaduje pouze minimum operační paměti, toho

se využívá u HW zařízení (fotoaparáty, pomalá paměťová média, tiskárny atd.) Mezi jeho zápory patří chybějící podpora alfa kanálu a na dnešní poměry nízká kvalita výsledného obrazu při použití vyšší komprese.

V roce 2000 měl být JPEG nahrazen novým formátem s názvem **JPEG 2000**. Ten přináší mnohá vylepšení oproti předchozímu formátu. Za zmínku stojí vyšší kompresní poměr při použití stejné kvality, vyšší odolnost proti chybám a podpora alfa kanálu. U JPEGu 2000 je datový tok rozdělen do vrstev, díky tomu je schopen při načítání obrázku (např. z internetu) postupně zvyšovat kvalitu s každou další dekódovanou vrstvou. Další jeho zajímavou funkcí je vyznačení oblasti zájmu (ROI). Lze nastavit, aby se např. obličej na fotografii dekodovaly přednostně vůči okolnímu prostředí (předsunutím magnitud koeficientů do popředí). JPEG 2000 má však i své záporné stránky, mezi ně se řadí vyšší výpočetní náročnost a s tím spojený delší kompresní čas. Při použití vysoké komprese dochází k velmi znatelnému rozmazání obrázku. To jsou nejspíš hlavní důvody, proč nebyl nikdy pořádně rozšířen.

Další formát, který se pokusil o nahrazení formátu JPEG, je formát **JPEG XR**. JPEG XR je schopen velmi efektivně využít komprimovaný snímek. Podporuje převod z mnoha barevných prostorů (RGB, CMYK, černobílá) o různé bitové hloubce. Stejně jako JPEG 2000 i tento formát obsahuje podporu pro bezztrátovou kompresi.

JPEG XR kombinuje výhody optimalizované kvality obrazu a efektivnosti komprese spolu s nízkou náročností požadavků na implementaci. Samozřejmostí jsou nové funkce jako vyšší kompresní schopnost komprese, nízké výpočetní a paměťové nároky, segmentace obrazu za účelem náhodného přístupu (u velkých obrazových formátů), podpora alfa kanálu a další funkce. Nově obsahuje i podporu pro aplikace s vysokým obrazovým rozsahem (HDR). JPEG XR je méně náchylný k vytváření artefaktů. Používá transformaci složenou ze 2 etap, ve které je možné uplatnit filtr na hranice mezi bloky a volitelně i mezi dlaždicemi. Tohle je blíže popsáno v sekci **Transformace**. Tyto funkce jsou paměťově náročné. I přes všechna tato vylepšení se ale ani tomuto formátu prozatím nepodařilo nahradit formát JPEG.

Posledním porovnávaným ztrátovým formátem je formát **WebP** vyvíjený firmou Google, který je založený na kompresním formátu videa (VP8). WebP podobně jako minulé dva formáty se snaží o nahrazení formátu JPEG ale s tím rozdílem, že se snaží prosadit na poli různých webových služeb. WebP podporuje bezztrátovou kompresi, ale mezi jeho největší výhody patří vysoká kompresní efektivnost s nízkou dekódovací složitostí. Jeho úspěch tkví především ve zvolení vhodné metody predikce.

WebP slibuje snížení paměťové náročnosti při použití stejné kvality o více než dvakrát oproti formátu JPEG a řada testů již tuto skutečnost potvrdila. V tuto chvíli WebP obsahuje podporu i pro alfa kanál. Formát WebP podporují téměř všechny webové prohlížeče a některé sociální sítě jej už testují (Facebook).

V následující tabulce je zobrazen krátký přehled o porovnávaných formátech:

formát	standard	bezeztrátový	transformace	alfa k.	HDR
JFIF/JPEG	ITU-T T.81	ne	DCT	ne	ne
JPEG 2000	ITU-T T.800	ano	DWT	ano	ne
JPEG XR	ITU-T T.832	ano	LBT (OPF a FCT)	ano	ano
WebP	RFC 6386 (VP8)	ano	DCT a WHT	ano	ne

Tabulka 3.1: Tabulka srovnání formátů. Pro upřesnění, JPEG obsahuje specifikaci pro bezeztrátovou kompresi, která se ale nevyužívá a místo toho byla nahrazena formátem JPEG-LS (rovněž se nepoužívá). V tabulce je uveden u formátu JPEG i JFIF, jelikož JPEG je pouze specifikací pro kompresi složky 2D obrazu a neudává význam složek (JFIF ano). HDR je vysoký obrazový rozsah (high dynamic range).

3.3 Kvalita komprese

V této sekci je popsáno praktické srovnání ztrátových formátů pomocí metriky SSIM. Nejprve bylo nutné vybrat vhodné obrázky, na kterých bude srovnání provedeno. V této práci je uvedeno srovnání na dvou rastrových obrázcích a jednom vektorovém obrázku. Rastrové obrázky byly vhodně zvoleny tak, aby neobsahovaly mnoho ostrých hran, které jsou nežádoucí pro ztrátovou kompresi. Na vektorovém obrázku bylo otestováno, jak si ztrátové formáty poradí s ostrými hranami a kontrastními přechody barev (černá a bílá).

Poté bylo nutné vyhledat nástroje pro kompresi obrazu do jednotlivých ztrátových formátů, které jsou blíže popsány v kapitole **Formáty ztrátové komprese obrazu**. Samotná komprese byla provedena z bezeztrátového formátu do příslušného ztrátového formátu pro 25 různých kvalit. Pro každou kvalitu bylo provedeno srovnání s původním obrázkem pomocí metriky SSIM. Výsledky byly vykresleny ve formě grafů, kde osa x znázorňuje datový tok v bitech na pixel a osa y vyjadřuje index kvality SSIM v logaritmickém měřítku.

V grafech na Obrázku 3.2 byly použity následující nástroje: Pro JPEG byla použita otevřená knihovna libjpeg (parametr `-baseline -quality`). Pro JPEG 2000 byla použita demoverze Kakadu (parametr `-rate`). Pro JPEG XR byl použit MS DPKv1.0 (parametr `-q`). Pro WebP byla použita otevřená knihovna libwebp verze 0.4.0 (parametr `-q`).

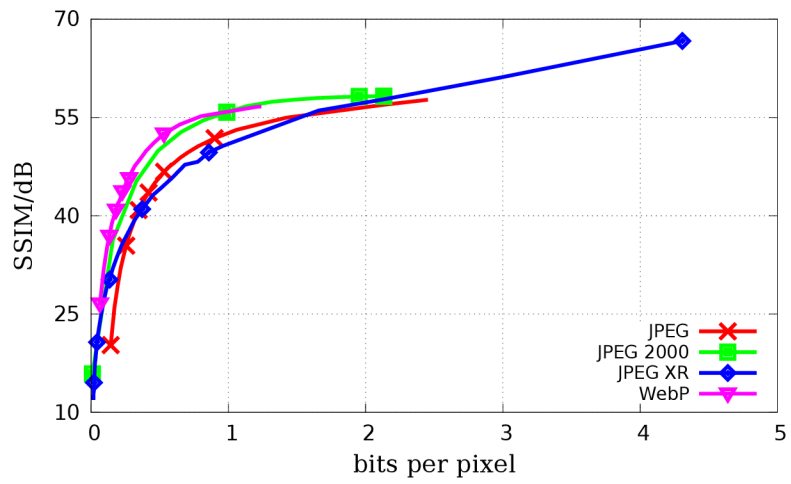


Obrázek 3.1: Uvedené obrázky byly vybrány pro porovnání ztrátových formátů metrikou SSIM.

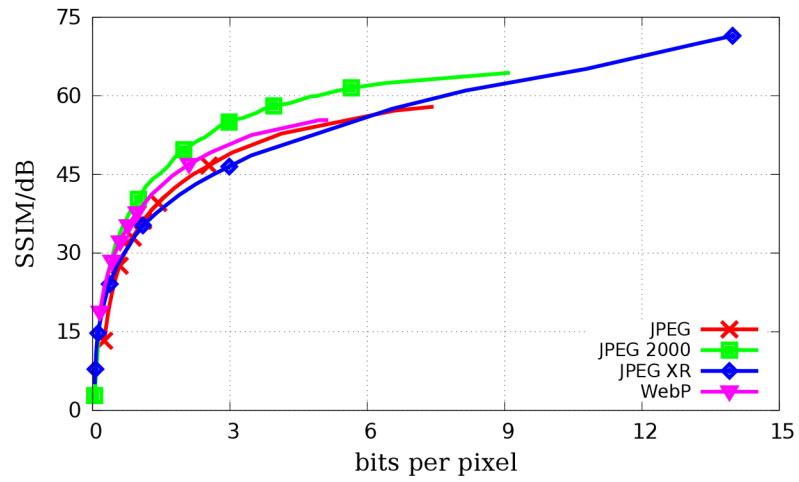
Z grafů 3.2a a 3.2b, které byly vytvořeny pomocí rastrových obrázků 3.1a a 3.1b, je patrné, že téměř všechny ztrátové formáty dosahují vysoké kvality indexu SSIM. Nejlepším kompresním poměrem při zachování vysoké kvality obrazu disponují ztrátové formáty JPEG 2000 a WebP. V nižších datových tocích má lepší kompresní poměr formát WebP a naopak ve vyšších datových tocích formát JPEG 2000. Formát JPEG XR zaznamenal propad, jelikož jeho komprimované soubory obsahují mnoho artefaktů.

Na vektorovém Obrázku 3.1c se s ostrými hranami nejlépe vypořádal formát WebP, který dosahuje vynikajícího kompresního poměru při zachování vysoké kvality obrazu. Velmi dobré výsledky zde podaly i formáty JPEG XR a JPEG. Všechny tyto formáty mají společné to, že jejich transformace alespoň v první etapě využívá transformaci DCT anebo její obdobnou verzi. Propad zde zaznamenal formát JPEG 2000, který využívá vlnkovou transformaci DWT.

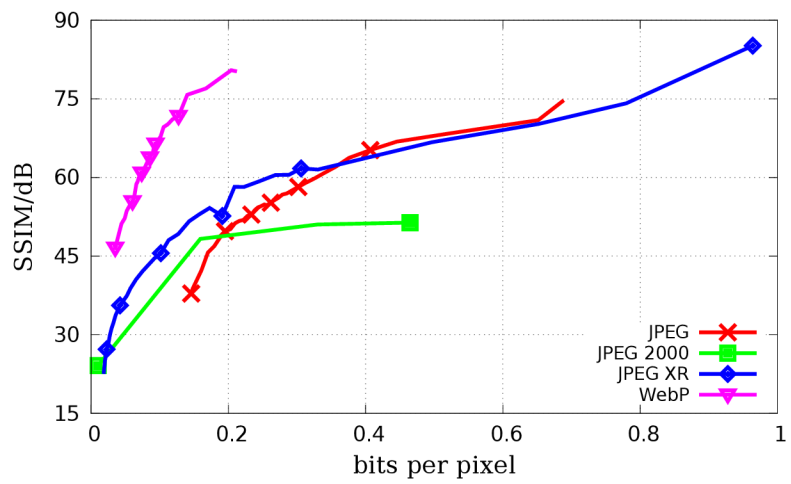
Další porovnání kompresního poměru je zobrazeno v příloze B, kde na Obrázku B.1 jsou uvedeny obrázky, na kterých bylo provedeno srovnání metrikou SSIM a na Obrázku B.2 jsou již zobrazeny výsledné grafy.



(a) Artificial



(b) Lenna



(c) Runner

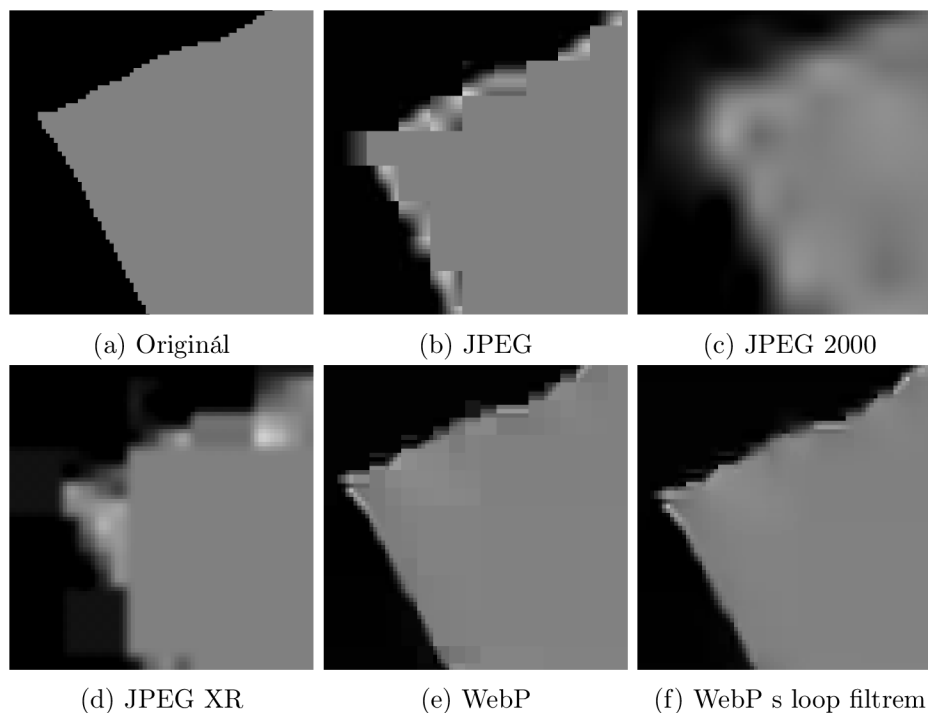
Obrázek 3.2: Srovnání kompresního poměru ztrátových formátů metrikou SSIM na Obrázcích 3.1a, 3.1b a 3.1c.

V nižších datových tocích vznikají v komprimovaných souborech různé typy artefaktů. Formát JPEG je nejvíce náchylný ke vzniku blokového efektu (blocking). Dochází k němu kvůli zpracování obrazu po blocích o velikosti 8×8 pixelů. Na Obrázku 3.3b jsou na hranách jasně vidět čtvercové oblasti o velikosti 8×8 pixelů, na kterých se objevují různě velké dvourozměrné kosinusoidy. Artefaktům, které vznikají na ostrých hranách, se říká prstencování (ringing).

Formát JPEG 2000 je nejvíce náchylný ke vzniku artefaktů rozmazání (blurring). K rozmazání dochází při použití vysoké komprese obrazu. Tento efekt je možné vidět na Obrázku 3.3c spolu s prstencováním na ostrých hranách, které se projevuje na pohled rozmazanými vlnkami. U komprimovaných obrázků může docházet i k blokovému efektu, který vzniká mezi dlaždicemi.

Formát JPEG XR je náchylný ke tvorbě blokových artefaktů, které vznikají mezi makrobloky o velikosti 16×16 pixelů a mohou vznikat i mezi dlaždicemi. Tyto artefakty jsou vidět na Obrázku 3.3d. JPEG XR se snaží s tímto problémem vypořádat pomocí transformace rozdělené do dvou etap, kdy každá etapa se skládá z transformace FCT (obdoba DCT) a z volitelné operace předfiltrování OPF. Právě tato operace předfiltrování cílí na hranice mezi makrobloky a volitelně i na oblast mezi dlaždicemi za účelem potlačení artefaktů ovšem za ceny vyšších výpočetních nároků.

Komprimované obrázky do formátu WebP s použitím loop filtru dosahují vysoké kvality i v nižších datových tocích. Při použití velmi vysoké komprese je tento formát náchylný k rozmazání obrazu. Výřez z komprimovaného obrázku formátem WebP s použitým loop filtrem je možné vidět na Obrázku 3.3f. Aby bylo možné detekovat bloky pixelů ve vysoce komprimovaném obrázku, byla navíc provedena komprese do formátu WebP s parametrem -nofilter, který při dekompresi obrazu nepoužívá loop filtr k vyhlazování hran. Obrázek 3.3e demonstruje tento případ a zobrazuje zde podbloky o velikosti 4×4 pixely, na kterých se objevují různě velké dvourozměrné kosinusoidy.



Obrázek 3.3: Na těchto obrázcích jsou demonstrovány artefakty jednotlivých ztrátových formátů na výřezu vektorového obrázku Runner. U formátu WebP byl při dekompresi použit parametr `-nofilter`, aby byly při přiblížení viditelné bloky pixelů.

3.4 Odolnost proti chybám

V této sekci bude demonstrována odolnost formátů vůči chybám na obrázku Lenna. Obrázek v bezztrátovém formátu byl nejprve komprimován s kvalitou cca 70% do ztrátového formátu a následně bylo zhruba uprostřed obrázku nahrazeno 72 bajtů hodnotou nula. Takto upravený obrázek byl poté převeden zpět do bezztrátového formátu.

Úmyslné zanesení jiných hodnot do obrázku mělo za následek různě značné znehodnocení výsledného obrazu. U formátu JPEG je výsledné znehodnocení zobrazeno na Obrázku 3.4a, ze kterého je patrné, že od místa znehodnocení se v obrázku objevuje několik bloků o velikosti 8×8 pixelů, které jsou velmi poškozené a několik dalších řádků bloků je posunuto kvůli DC složce, která se kóduje diferencially a obsahuje zkreslené barvy. Jelikož byly při kompresi přidány restartovací značky, tak se obrázek od určitého řádku nakonec chytl a až do konce souboru se již zobrazuje korektně.

U formátu JPEG 2000 je znehodnocení demonstrováno na Obrázku 3.4b. Tento formát se se ztrátou dat vyrovnal nejlépe ze všech formátů. Obraz se zde dekóduje celý po vrstvách, kde s každou další dekódovanou vrstvou dochází k postupnému zlepšování kvality. Kvůli chybějícím datům se po celém obrázku objevují artefakty v podobě rozmazaných obdelníků, což je způsobeno přičtením nějaké vlnky s jinou vahou než by měla mít, ale výsledek stále působí pro lidské oko dobrým dojmem.

U formátu JPEG XR je znehodnocení zobrazeno na Obrázku 3.4c. Z obrázku je zřejmé, že od místa znehodnocení došlo k velmi znatelnému poškození obrázku. Od tohoto místa působí obrázek velmi rozmazaným dojmem, ale stále lze odhadnout, o jaký obrázek se jednalo, jelikož JPEG XR podobně jako JPEG používá restartovací značky. V poškozené části

se náhodně v obraze objevují dvourozměrné kosinusoidy v různých barvách.

U formátu WebP je znehodnocení demonstrováno na Obrázku 3.4d. Tento formát se se ztrátou dat vyrovnal nejhůře. Na poškozeném obrázku se od místa znehodnocení objevují různě barevné rozmazané plochy obdelníkového tvaru, ze kterých není příliš patrné, o jaký obrázek se původně jednalo. Tento formát využívá k dekódování obrazu predikci bloků a je velmi náchylný ke ztrátě dat.

Další srovnání odolnosti ztrátových formátů vůči chybám je provedeno na dvou obrázcích v příloze A.



Obrázek 3.4: Na těchto obrázcích je demonstrována odolnost ztrátových formátů vůči chybám. Zhruba uprostřed komprimovaného obrázku bylo zaměněno 72 bajtů hodnotou nula. Ke kompresi byly využity stejné nástroje, které jsou popsány v sekci **Kvalita komprese**. Změna se týká pouze formátů JPEG, kde při kompresi byl použit parametr `-restart` pro použití restartovacích značek, a formátu JPEG XR, kde ke kompresi byl použit program GIMP s pluginem JPEG XR.

3.5 Interaktivní tutoriál

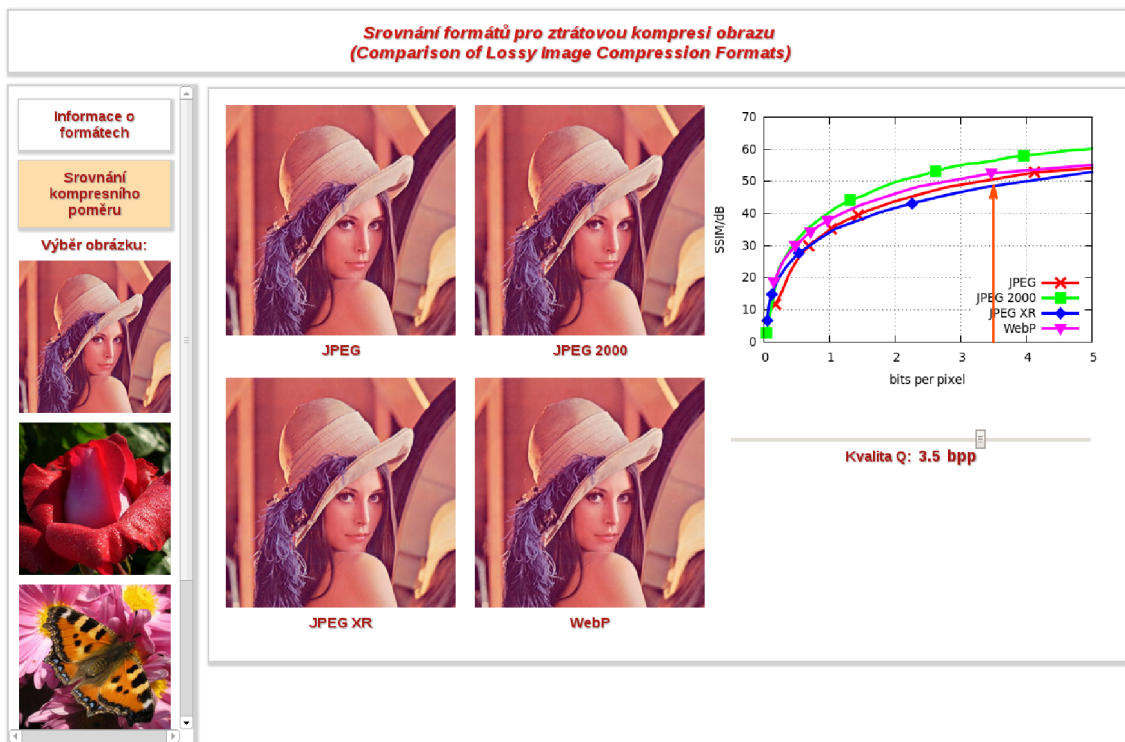
Praktická část bakalářské práce zahrnuje i implementaci výukové aplikace, která by měla sloužit k demonstraci kvality komprimovaných obrazů. Dále by měla poukázat na artefakty, které vznikají při vyšší kompresi obrazu. Různé artefakty, ke kterým jsou ztrátové formáty náchylné, jsou popsány v sekci **Kvalita komprese**. Výuková aplikace je umístěna na webových stránkách: http://www.stud.fit.vutbr.cz/~xkolac08/Vyukova_aplikace/VyukovaApp.html a <http://ztratoveformaty.ic.cz/index.html>.

Uživatel si bude moci pohodlně zvolit kvalitu komprese a porovnat si výsledné datové toky komprimovaných obrazů jednotlivých formátů. U každého obrazu navíc bude přítomen graf závislosti kvality SSIM na datovém toku.

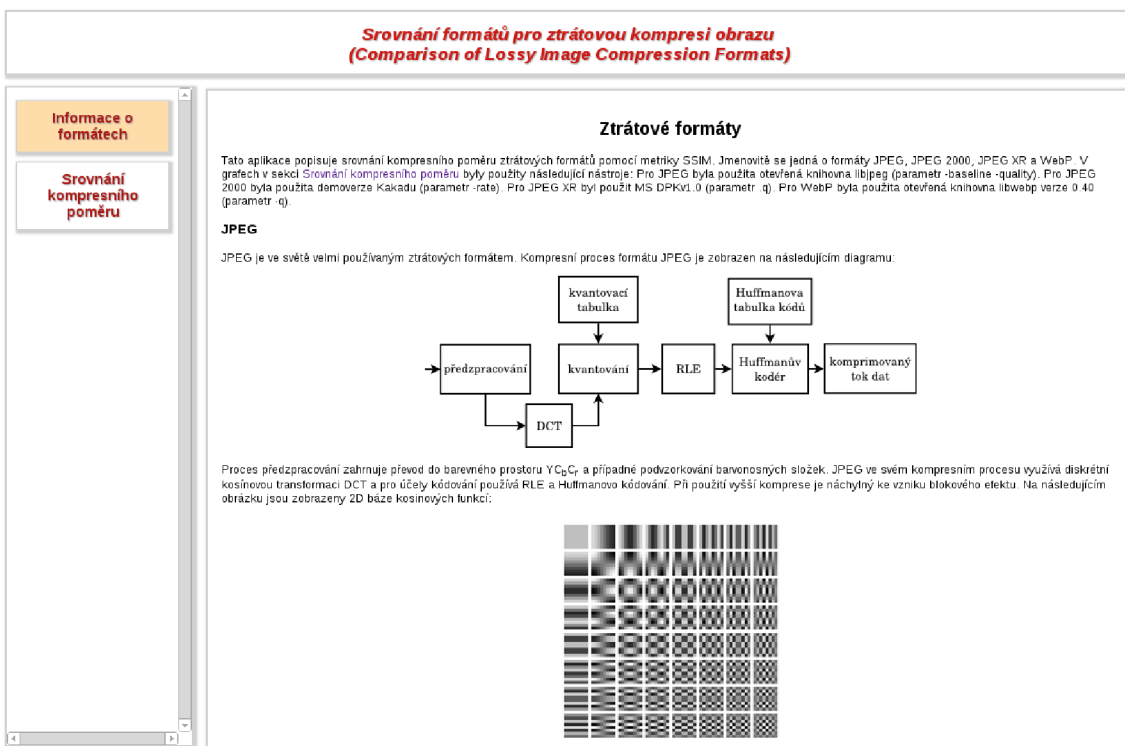
3.5.1 Návrh

Byl navrhnut interaktivní tutoriál, který demonstruje srovnání kompresního poměru ztrátových formátů JPEG, JPEG 2000, JPEG XR a WebP. Aplikace je rozdělena na menu v levé části, odkud si uživatel může zobrazit teoretické informace o ztrátových formátech anebo si rovnou může vybrat jeden z dostupných obrázků, na kterém bude provedeno srovnání kompresního poměru.

Dále se v aplikaci nachází hlavní okno, kde se zobrazují informace o ztrátových formátech anebo samotné porovnání formátů. Srovnání kompresního poměru je zde demonstrováno na 4 obrázcích, kde každý přísluší jednomu ze ztrátových formátů. Dále je zde uveden graf závislosti indexu kvality SSIM na datovém toku. Uživatel může pomocí posuvníku libovolně měnit datový tok, pro který chce dané obrázky zobrazit. Na změnu datového toku reaguje i šipka, která více upřesňuje, o jaký index kvality se právě jedná. Na následujících Obrázcích je zobrazena ukázka aplikace:



Obrázek 3.5: Srovnání kompresního poměru.



Obrázek 3.6: Informace o ztrátových formátech.

3.5.2 Implementace

Pro implementaci výukové aplikace byl použit poměrně nový značkovací jazyk HTML5, který byl navržen tak, aby obsahoval všechny důležité nástroje bez nutnosti přidávání dalších pluginů. Současná verze HTML5 podporuje animace, přidávání hudby, vytváření náročnějších webových aplikací a mnoho dalšího.

Moderní vzhled aplikace byl zajištěn pomocí kaskádových stylů CSS3, pomocí kterých bylo např. docíleno 3D efektu boxů. Dále byl použit efekt stínování textu a dalších objektů a jiné úpravy. Pro zajištění kompatibility u webových prohlížečů Google Chrome, Mozilla Firefox a Internet Explorer verze 10 a vyšší byla dále použita knihovna JavaScriptu jQuery. Tato knihovna byla přímo přidána do aplikace, a tím pádem není nutné, aby si ji uživatel musel doinstalovat k jejímu správnému spuštění.

Porovnávané obrázky, které byly použity v aplikaci, byly získány pomocí nástrojů popsaných v sekci **Kvalita komprese**. Z takto získaných obrázků bylo navíc důležité vybrat ty, které nejlépe vyhovují určitému datovému toku. Za účelem tohoto výběru byla implementována řada skriptů, které postupně vyhledaly obrázky s nejbližším datovým tokem. V aplikaci je navíc ke každému datovému toku zobrazen i graf, který pomocí šipky signalizuje index kvality SSIM pro právě zvolený datový tok. Tyto grafy byly vykresleny pomocí nástroje Gnuplot.

Kapitola 4

Závěr

V práci byly popsány ztrátové formáty, konkrétně JPEG, JPEG 2000, JPEG XR a WebP, spolu se základními principy komprese obrazu a metrikami pro výpočet kvality obrazu. Hlavním cílem této práce bylo nalézt ztrátový formát s nejlepším kompresním poměrem při zachování vysoké kvality obrazu. Za tímto účelem jsem provedl teoretické srovnání výše zmíněných ztrátových formátů, které napovědělo, že nejlepší kompresní poměr při zachování vysoké kvality obrazu bude mít formát JPEG 2000, anebo JPEG XR.

Aby bylo možné objektivně zhodnotit kvalitu ztrátových formátů, tak jsem dále provedl praktické srovnání formátů pomocí metriky SSIM na rastrových obrázcích. Výsledky potvrdily, že nejlepším kompresním poměrem při zachování vysoké kvality obrazu disponuje právě formát JPEG 2000. Velmi neočekávané byly výsledky formátu WebP, který v nižších datových tocích mnohdy dosahuje ještě lepších výsledků než formát JPEG 2000. Velmi dobré výsledky podal i formát JPEG. Naopak značný propad zaznamenal formát JPEG XR, kvůli četným artefaktům v komprimovaných obrázcích.

Poněkud jiné výsledky jsem získal při srovnání formátů na původně vektorovém obrázku s ostrými hranami. Nejhorší výsledky zde podal formát JPEG 2000 a nejlepší formát WebP. Velmi dobře se zde umístil formát JPEG XR. Na Obrázku 3.3 je zobrazen přiblížený výřez vektorového obrázku, který demonstruje artefakty ztrátových formátů.

Dále jsem rovněž naimplementoval **Interaktivní tutoriál**, na kterém uživatel může libovolně nastavovat datový tok, pro který chce zobrazit komprimované obrázky za účelem subjektivního porovnání ztrátových formátů.

Dalším možným navazujícím výzkumem by mohlo být porovnání časové náročnosti komprese výše zmíněných formátů.

Literatura

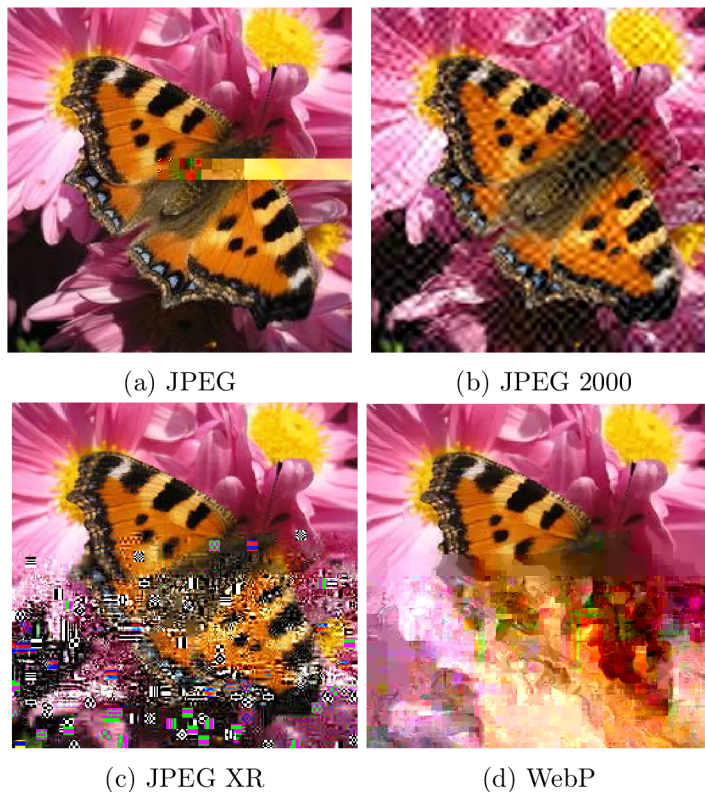
- [1] Bankoski, J.; Koleszar, J.; Quillio, L.; aj.: VP8 Data Format and Decoding Guide. *RFC 6386*, listopad 2011, ISSN 2070-1721.
- [2] Bankoski, J.; Wilkins, P.; Xu, Y.: Technical overview of VP8, an open source video codec for the web. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, červenec 2011, ISSN 1945-7871, s. 1–6, doi:10.1109/ICME.2011.6012227.
- [3] Bařina, D.; Zemčık, P.: *Multimédia - Studiní opora*. FIT VUT v Brně, 2013, 115 s.
- [4] Bařina, D.; Zemčık, P.; Beran, V.; aj.: *Zpracování obrazu - Studiní opora*. FIT VUT v Brně, 2011, 92 s.
- [5] Chen, M.-J.; Bovik, A. C.: Fast Structural Similarity Index Algorithm. *J. Real-Time Image Process.*, ročník 6, č. 4, Prosinec 2011: s. 281–287, ISSN 1861-8200, doi:10.1007/s11554-010-0170-9.
- [6] Dąbrowska-Boruch, A.; Wiatr, K.: Implementation of FCT Transformation in JPEG-XR Standard in Programmable Devices. *AUTOMATYKA*, 2011: s. 83–91, ISSN 1429-3447.
- [7] Dosselmann, R.; Yang, X.: A Formal Assessment of the Structural Similarity Index. Technická zpráva, University of Regina. Department of Computer Science, 2008.
- [8] ITU-T T.800 | ISO/IEC 15444-1: JPEG 2000 image coding system: Core coding system. 2000.
- [9] ITU-T T.81 | ISO/IEC 10918-1: Information technology - Digital compression and coding of continuous-tone still images: Requirements and guidelines. 1992.
- [10] ITU-T T.832 | ISO/IEC 29199-2: Information technology – JPEG XR image coding system – Image coding specification. 2012.
- [11] Lang, Y.: *Evaluating and Implementing JPEG XR Optimized for Video Surveillance*. Diplomová práce, Institute of Technology in Linköping University, S-581 83 Linköping, Sweden, 2010.
- [12] Lyons, R.: Frequency domain tutorial, part 2: complex signals and spectral diagrams. http://www.eetimes.com/document.asp?doc_id=1275580, 2008.
- [13] National Instruments: Peak Signal-to-Noise Ratio as an Image Quality Metric. <http://www.ni.com/white-paper/13306/en/>, 2013.

- [14] Muhammad Riaz ur Rehman, A. K. K., Gulistan Raja: Implementation of Lapped Biorthogonal Transform for JPEG-XR Image Coding. In *Advanced Video Coding for Next-Generation Multimedia Services*, 2012, ISBN 978-953-51-0929-7, doi:10.5772/53100.
- [15] Tišnovský, P.: Ztrátová komprese obrazových dat pomocí JPEG. *ROOT, informace nejen ze světa Linuxu*, 2006, ISSN 1212-8309.
- [16] Tišnovský, P.: Programujeme JPEG: diskrétní kosinová transformace (DCT). *ROOT, informace nejen ze světa Linuxu*, 2007, ISSN 1212-8309.

Příloha A

Odolnost vůči chybám

Na těchto obrázcích je demonstrována odolnost ztrátových formátů vůči chybám. Zhruba uprostřed komprimovaného obrázku bylo zaměněno 72 bajtů hodnotou nula. Ke kompresi byly využity stejné nástroje, které jsou popsány v sekci **Kvalita komprese**. Změna se týká pouze formátů JPEG, kde při kompresi byl použit parametr `-restart` pro použití restartovacích značek a formátu JPEG XR, kde ke kompresi byl použit program GIMP s pluginem JPEG XR.



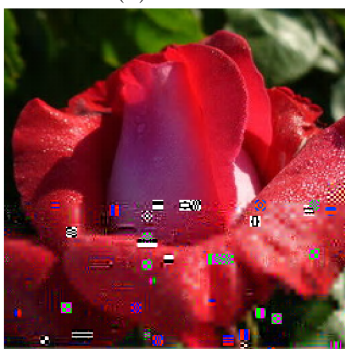
Obrázek A.1: Srovnání odolnosti ztrátových formátů vůči chybám.



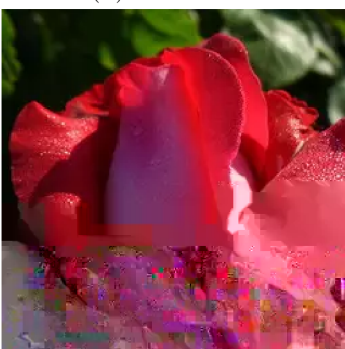
(a) JPEG



(b) JPEG 2000



(c) JPEG XR



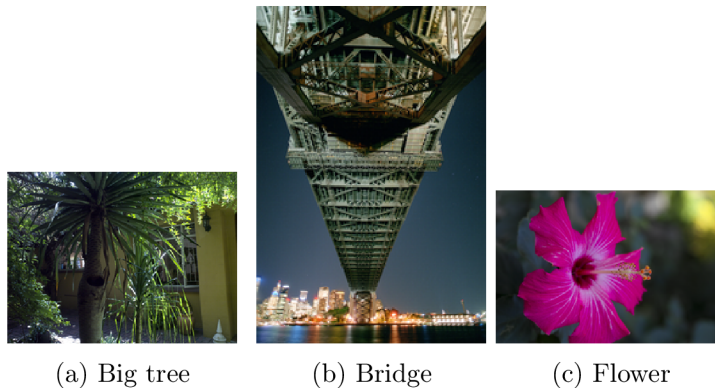
(d) WebP

Obrázek A.2: Srovnání odolnosti ztrátových formátů vůči chybám.

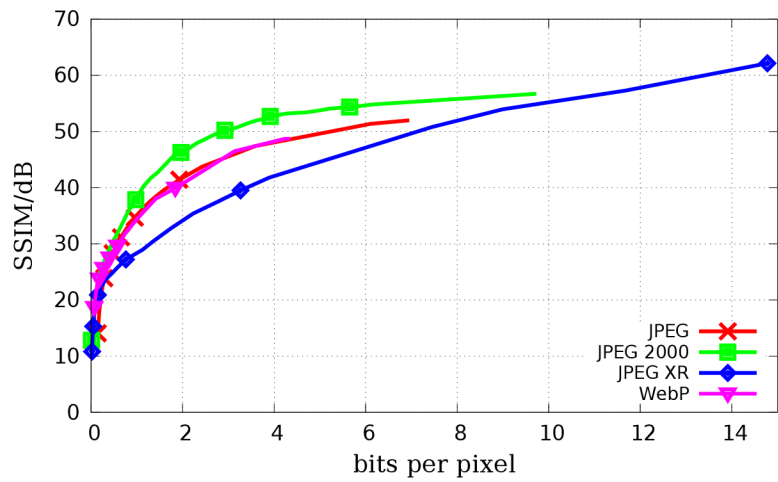
Příloha B

Srovnání kompresního poměru

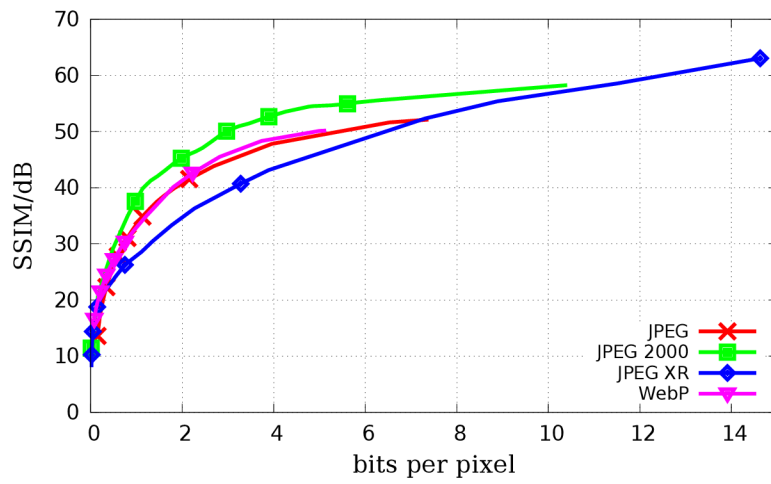
V této sekci se nachází srovnání kompresního poměru metrikou SSIM. K vytvoření grafů na Obrázku B.2 byly využity stejné nástroje, které jsou popsány v sekci **Kvalita komprese**. Výsledky srovnání kompresního poměru na Obrázcích B.1a a B.1b se velmi podobají výsledkům v kapitole **Kvalita komprese**. Nejlepšího kompresního poměru při zachování vysoké kvality obrazu dosahuje formát JPEG 2000 a velmi dobré výsledky podaly i formáty WebP a JPEG. U Obrázku B.1c jsou výsledky velmi vyrovnané, ale i zde nejlepším kompresním poměrem disponuje formát JPEG 2000.



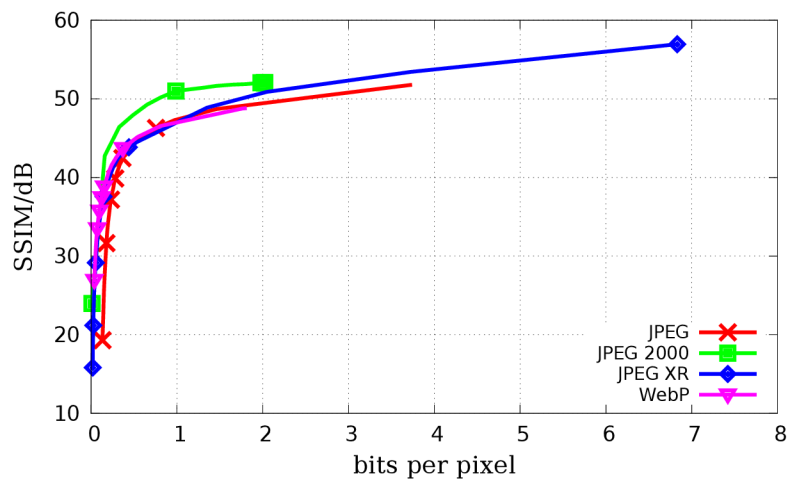
Obrázek B.1: Uvedené obrázky byly vybrány pro porovnání ztrátových formátů metrikou SSIM.



(a) Big tree



(b) Bridge



(c) Flower

Obrázek B.2: Srovnání kompresního poměru ztrátových formátů metrikou SSIM na Obrázcích B.1a, B.1b a B.1c.