

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

MODERNÍ KOMPRESNÍ FORMÁTY DIGITÁLNÍHO OBRAZU

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

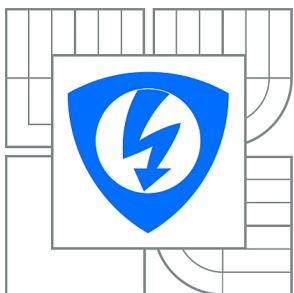
RADEK EDER

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

# MODERNÍ KOMPRESNÍ FORMÁTY DIGITÁLNÍHO OBRAZU

MODERN COMPRESSION FORMATS OF DIGITAL IMAGE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

RADEK EDER

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETR ČÍKA, Ph.D.

BRNO 2014



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Bakalářská práce

bakalářský studijní obor  
Teleinformatika

**Student:** Radek Eder

**ID:** 145934

**Ročník:** 3

**Akademický rok:** 2013/2014

## NÁZEV TÉMATU:

**Moderní kompresní formáty digitálního obrazu**

## POKYNY PRO VYPRACOVÁNÍ:

Prostudujte kompresní formáty využívané pro kompresi digitálního obrazu, zejména JPEG, JPEG-2000, PNG a jejich možné implementace. Prostudujte způsoby, jakými lze objektivně i subjektivně vyhodnotit kvalitu komprimovaného obrazu. Navrhněte aplikaci, která bude umožňovat kompresi vloženého nekomprimovaného digitálního obrazu do výše zmíněných formátů a následně otestuje kvalitu komprese.

## DOPORUČENÁ LITERATURA:

[1] TAUBMAN, David S. JPEG2000: Image Compression Fundamentals, Standards and Practice. New York: Springer, 2002, xix, 773 s. ISBN 07-923-7519-X.

[2] ISO/IEC 15948:2003. Portable Network Graphics (PNG) Specification (Second Edition). Oxford Brookes University: W3C, 2003. Dostupné z: <http://www.w3.org/TR/PNG/>

**Termín zadání:** 10.2.2014

**Termín odevzdání:** 4.6.2014

**Vedoucí práce:** Ing. Petr Číka, Ph.D.

**Konzultanti bakalářské práce:**

**doc. Ing. Jiří Mišurec, CSc.**

*Předseda oborové rady*

## UPOZORNĚNÍ:

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

## **ABSTRAKT**

V této bakalářské práci je nastíněna problematika komprese digitálního obrazu. Jsou zde popsány základní barevné modely RGB a YCbCr a převod mezi nimi. Popsány jsou současně používané kodeky digitálního obrazu. Detailněji je rozebrán kompresní formát JPEG, který v současné době je stále nejrozšířenějším formátem pro digitální fotografii. Popsán je postup při digitalizaci obrazu pomocí standardu JPEG, kam patří vzorkování, kvantování a kódování. Dále rozebrány formáty JPEG2000, PNG a WebP. Závěrem práce jsou rozebrány metody pro subjektivní a objektivní vyhodnocení kvality komprese digitálního obrazu. Mezi subjektivní metody patří Double Stimulus Impairment Scale (DSIS) a Double Stimulus Continuous Quality Scale (DSCQS), mezi objektivní pak Mean Squared Error (MSE), Peak Signal to Noise Ratio (PSNR) a Structural Similarity (SSIM). V rámci práce byla vytvořena i aplikace, která umožňuje komprimovat zdrojové obrazu do vybraných formátů. Dále umožňuje provést kvalitativní vyhodnocení komprese a uložit výsledné hodnoty do souboru.

## **KLÍČOVÁ SLOVA**

Komprese, PSNR, MSE, SSIM, JPEG, JPEG2000, PNG, WebP, Kvantování, Vzorkování

## **ABSTRACT**

This bachelor thesis deals with the problems of digital image compression. There are described basic color models RGB and YCbCr and conversion between them. Described are currently used codecs for digital images. JPEG compression format is discussed in more detail, which currently is still the most widely used format for digital photos. Described is a process of digitizing the image using the JPEG standard, which includes sampling, quantization and coding. Described are also formats JPEG2000, PNG and WebP. End of chapter are described methods for subjective and objective evaluation of the quality of digital image compression. Among the subjective methods include Double Stimulus Impairment Scale (DSIS) and Double Stimulus Continuous Quality Scale (DSCQS), then the objective methods include Mean Squared Error (MSE), Peak Signal to Noise Ratio (PSNR) a Structural Similarity (SSIM). Application was created as part of the work that allows you to compress the source image to the selected formats. It also allows a qualitative evaluation of compression and store the results to the file.

## **KEYWORDS**

Compression, PSNR, MSE, SSIM, JPEG, JPEG2000, PNG, WebP, Quantization, Sampling

EDER, Radek *Moderní kompresní formáty digitálního obrazu: bakalářská práce*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 64 s. Vedoucí práce byl Ing. Petr Číka, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Moderní kompresní formáty digitálního obrazu“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno .....

.....

(podpis autora)

## PODĚKOVÁNÍ

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

Brno .....

.....

(podpis autora)



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

## PODĚKOVÁNÍ

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

Brno .....

.....

(podpis autora)



EVROPSKÁ UNIE  
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ  
INVESTICE DO VAŠÍ BUDOUCNOSTI



# OBSAH

Úvod	11
<b>1 Reprezentace obrazu</b>	<b>12</b>
1.1 Barevné modely	12
1.1.1 Model RGB	12
1.1.2 Model YCbCr	12
1.1.3 Převod z RGB do YCbCr	13
1.2 Barevná hloubka	13
<b>2 Obrazové formáty</b>	<b>14</b>
2.1 BMP	14
2.2 GIF	14
2.3 PNG	14
2.4 TGA	14
2.5 TIFF	15
2.6 JPEG	15
2.7 JPEG2000	15
2.8 WebP	15
<b>3 Rozbor obrazových formátů</b>	<b>17</b>
3.1 Popis formátu JPEG	17
3.1.1 Transformace barev	17
3.1.2 Redukce barev	17
3.1.3 DCT (Diskrétní kosinová transformace)	17
3.1.4 Kvantizace	18
3.1.5 Kódování	19
3.1.6 VLC – Variable Length Coding	21
3.1.7 Dekódování	23
3.2 Popis formátu JPEG2000	24
3.2.1 DC Level Shifting – posun stejnosměrné složky	24
3.2.2 Transformace barev	24
3.2.3 DWT – Diskrétní vlnková transformace	25
3.2.4 Kvantizace	27
3.2.5 ROI – Region of Interest Coding	27
3.2.6 Entropické kódování	28
3.2.7 Komprese výstupního datového toku	28
3.3 Popis formátu PNG	29

3.3.1	Příprava PNG obrázku . . . . .	29
3.3.2	Prokládání pixelů . . . . .	30
3.3.3	Filtrování . . . . .	31
3.3.4	Komprese . . . . .	32
3.3.5	Chunking . . . . .	32
3.3.6	Popis jednotlivých chunků . . . . .	33
3.4	Popis formátu WebP . . . . .	36
3.4.1	Ztrátový WebP . . . . .	36
3.4.2	Bezeztrátový WebP . . . . .	37
<b>4</b>	<b>Hodnocení kvality komprese</b>	<b>39</b>
4.1	MSE – střední kvadratická chyba . . . . .	39
4.2	PSNR – špičkový poměr signálu k šumu . . . . .	40
4.3	SSIM – Structural Similarity . . . . .	40
4.4	Další neimplementované metody . . . . .	42
4.4.1	NAE – Normalizovaná absolutní chyba . . . . .	42
4.4.2	NC – Normalizovaná korelace . . . . .	42
4.4.3	AD – Průměrná odchylka . . . . .	42
4.4.4	MD – Maximální odchylka . . . . .	43
4.4.5	NMSE – Normalizovaná střední kvadratická chyba . . . . .	43
<b>5</b>	<b>Praktická realizace</b>	<b>44</b>
5.1	Analyzátor kvality komprese . . . . .	44
<b>6</b>	<b>Vyhodnocení</b>	<b>47</b>
<b>7</b>	<b>Závěr</b>	<b>58</b>
	<b>Literatura</b>	<b>59</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>61</b>
<b>A</b>	<b>Obsah přiloženého CD/DVD</b>	<b>64</b>

# SEZNAM OBRÁZKŮ

1.1	Barevný model RGB . . . . .	13
3.1	Rozdělení obrazu do bloků o velikosti $8 \times 8$ pixelů . . . . .	17
3.2	Blokový diagram baseline JPEG enkodéru . . . . .	20
3.3	Rozdílové kódování DC koeficientu . . . . .	20
3.4	čtení zig-zag AC koeficientů . . . . .	22
3.5	Dvourozměrná diskretní vlnková transformace . . . . .	26
3.6	Vlnková dekompozice druhé a třetí úrovně . . . . .	26
3.7	Souvislost mezi bitovým tokem a prostorovými daty . . . . .	29
3.8	Prokládání pixelů metodou Adam7 u formátu PNG . . . . .	31
3.9	Poloha bytů použitých při filtrování . . . . .	31
5.1	Hlavní okno programu . . . . .	44
5.2	Okno s výsledky testování . . . . .	45
5.3	Okno s parametry testování . . . . .	46
6.1	Vzorové obrázky použité k testování (Lena, Baboon, Buildings, Plane)	47
6.2	Porovnání kompresních formátů při 0,09 bpp (vlevo) a 0,132 bpp (vpravo), (a) a (b) JPEG, (c) a (d) JPEG2000, (e) a (f) WebP . . . .	49
6.3	Porovnání kompresních formátů při 0,23 bpp (vlevo) a 0,771 bpp (vpravo), (a) a (b) JPEG, (c) a (d) JPEG2000, (e) a (f) WebP . . . .	50
6.4	Porovnání kompresních formátů při 1,053 bpp (vlevo) a 1,511 bpp (vpravo), (a) a (b) JPEG, (c) a (d) JPEG2000, (e) a (f) WebP . . . .	51
6.5	Porovnání formátů měřením PSNR na obrázku Baboon . . . . .	52
6.6	Porovnání formátů měřením PSNR na obrázku Buildings . . . . .	52
6.7	Porovnání formátů měřením PSNR na obrázku Lena . . . . .	53
6.8	Porovnání formátů měřením PSNR na obrázku Plane . . . . .	53
6.9	Porovnání formátů měřením SSIM na obrázku Baboon . . . . .	54
6.10	Porovnání formátů měřením SSIM na obrázku Buildings . . . . .	54
6.11	Porovnání formátů měřením SSIM na obrázku Lena . . . . .	55
6.12	Porovnání formátů měřením SSIM na obrázku Plane . . . . .	55

## SEZNAM TABULEK

3.1	Kvantizační tabulka pro jasovou složku . . . . .	19
3.2	Kvantizační tabulka pro barevné složky . . . . .	19
3.3	kategorie (CAT) a DC Huffmanovi koeficienty pro jasovou složku . . .	21
3.4	Ukázka Huffmanova kódování . . . . .	23
3.5	Ukázka Aritmetického kódování . . . . .	23
3.6	Koeficienty CDF biortogonálních vlnkových filtrů . . . . .	27
3.7	Podporované kombinace typu barvy a bitové hloubky formátu PNG .	30
3.8	Význam velikosti znaků v názvu typu chunku . . . . .	33
3.9	Typy chunků uvedené v jakém pořadí by se měly nacházet . . . . .	34
6.1	Naměřené hodnoty na obrázku Baboon . . . . .	56
6.2	Naměřené hodnoty na obrázku Buildings . . . . .	56
6.3	Naměřené hodnoty na obrázku Lena . . . . .	57
6.4	Naměřené hodnoty na obrázku Plane . . . . .	57

# ÚVOD

S digitálním obrazem přišel do styku téměř každý. V posledním desetiletí se stal neodmyslitelnou součástí našich životů. Avšak málokdo o digitálním obrazu ví více, než že je pořízen digitálním fotoaparátem a uložen na paměťovém médiu. Mezi těmito krajními stavy probíhá spousta dalších úkonů, které běžný uživatel ani nepotřebuje znát. Nejprve je potřeba analogový obraz převést do digitálního světa, tento proces začíná u vzorkování, kvantování a kódování. Důležitým procesem je komprese, která značně ovlivňuje výslednou kvalitu obrazu.

Posouzení kvality komprese bude hlavním cílem této práce. Hlavním atributem komprese je dosažení co nejvyšší kvality obrazu, při co nejmenším objemu dat a zároveň i dosažení malé výpočetní náročnosti. Dle použití digitálního obrazu je potřeba volit určitý kompromis. Značný vliv má i zvolený kompresní formát.

Podrobněji zde bude popsán dnes nejvíce používaný formát JPEG (Joint Photographic Expert Group). Tento formát je již značně zastaralý, ale právě díky jeho rozšíření a jednoduchosti implementace, pravděpodobně jen tak nezmizí. Práce rozebírá i formát JPEG2000, dosahující větší komprese, za cenu výpočetní náročnosti. Dále formát PNG vhodný spíše na grafiku. A v neposlední řadě i formát WebP, jehož cílem je nahradit veškeré obrazové formáty.

K posouzení komprese vzniklo několik postupů, které objektivně posuzují kvalitu komprimovaného obrazu na základě porovnání s originálním (nekomprimovaným) obrazem. Práce se bude zabývat třemi metodami posouzení kvality komprese. Nejjednodušší metodou je MSE, tedy průměrná kvadratická chyba, na jejímž základě je postavena další metoda PSNR, špičkový poměr signálu k šumu. Současně nejobektivnější metoda je SSIM, jejíž výstup se nejvíce blíží subjektivnímu posouzení.

Úkolem této práce, bylo i vytvoření jednoduchého programu, který bude schopen 2 obrazy zvoleným kompresním formátem zkomprimovat a kvalitativními metodami posoudit. Samotným programem bylo otestováno několik vzorových obrazů a získány výsledky.

# 1 REPREZENTACE OBRAZU

Nejprve je potřeba získat teoretické znalosti o reprezentaci obrazu v počítači. V této kapitole budou stručně popsány barevné modely a převody mezi nimi, potřebné k pochopení principu metod posouzení kvality obrazu. Popsány budou i základy převodu obrazu do digitální podoby a závěrem i konkrétní rozbor obrazového formátu JPEG.

## 1.1 Barevné modely

Obraz lze vyjádřit s použitím několika barevných modelů. Pro oblast komprese obrazu jsou důležité zejména modely RGB a YCbCr. Další modely nebudou v práci více rozebírány. Pro televizní signál je používán model YUV. Pro tisk model CMYK a další používané modely jsou např. HSL a HSV, též známý jako HSB.

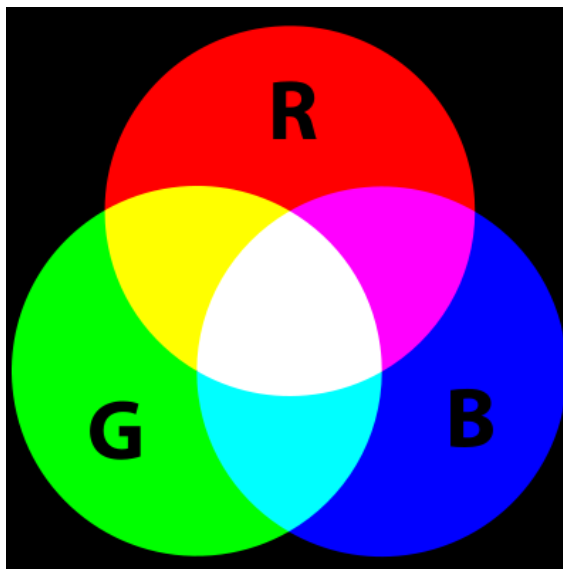
### 1.1.1 Model RGB

Barevný model RGB používá základních barev červená (Red), zelená (Green) a modrá (Blue). Další barvy vznikají mísením těchto 3 základních barev. Model je tzv. aditivní, založený na principu mísení světla. Jednotlivé barevné složky si můžeme představit jako reflektory v temné místnosti, vyzařující světlo o určité vlnové délce, pokud je smícháme, stává se výsledný odstín světlejší. Pokud smícháme všechny složky, vznikne barva bílá. Z tohoto důvodu, je model vhodný pro použití u zařízení, které vyzařují nebo zachytávají světlo. Tedy např. LCD monitory, televize, skenery, digitální fotoaparáty. Ukázka modelu je na obr. 1.1.

V digitální podobě v kombinaci s 24bitovou barevnou hloubkou (8 bitů pro každou barevnou složku), reprezentuje každou barevnou složku číslo o maximální hodnotě 255. Barvu tedy můžeme zapsat pomocí trojrozměrné soustavy třemi hodnotami. RGB barva (255, 255, 255) značí bílou, naproti tomu (0, 0, 0) černou.

### 1.1.2 Model YCbCr

Model YCbCr vychází z modelu YUV. Model YUV vznikl při použití s televizní technikou a to z důvodu oddělení jasové složky (Y) od barevných složek (UV), pro snadnější přenos signálu a přechod z černobílého na barevné vysílání. Model YCbCr není používán k zobrazování, vznikl pouze k zjednodušení kódování a komprese obrazu. Je snazší a vhodnější kódovat jednotlivé složky zvlášť. Jasová složka (Y) je stejně jako u modelu YUV oddělena od obrazových informací (CbCr). Cb a Cr značí modrou, respektive červenou chrominanční složku. Pokud zobrazíme pouze jasovou složku barevného obrazu, vznikne obraz vyjádřený ve stupních šedi.



Obr. 1.1: Barevný model RGB

### 1.1.3 Převod z RGB do YCbCr

Pro potřeby komprese, je potřeba obraz převést z modelu RGB do modelu YCbCr. Konkrétně pro obrazový formát JPEG/JFIF je převod pro 8bitové barvy definován vzorcem

$$\begin{pmatrix} Y' \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} + \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,168736 & -0,331264 & 0,5 \\ 0,5 & -0,418688 & -0,081312 \end{pmatrix} \begin{pmatrix} R' \\ G' \\ B' \end{pmatrix}. \quad (1.1)$$

## 1.2 Barevná hloubka

Každý pixel obrazu je reprezentován určitou hodnotou. Pojmem barevná hloubka vyjadřujeme maximální možnou hodnotu tohoto pixelu. Obraz ve stupních šedi je zpravidla 8bitový (pixel může nabývat hodnot 0–255), barevný obraz je zpravidla 24bitový (8 bitů pro každou barevnou složku) nebo může být i 32bitový (barevné složky jsou doplněny o Alfa kanál – průhlednost). Hodnotu barevné hloubky označujeme zkratkou bpp (bits per pixel).

## 2 OBRAZOVÉ FORMÁTY

Tato kapitola má za úkol stručně představit současné formáty digitálního obrazu, vyzdvihnout jejich výhody a nevýhody.

### 2.1 BMP

Formát BMP (Microsoft Windows Bitmap) byl vytvořen v roce 1988 firmou Microsoft Corporation. Výhodami tohoto formátu jsou jeho jednoduchost, dobrá dokumentovanost a absence patentové ochrany. Téměř každý grafický editor podporuje tento formát. Podporuje 1, 4, 8, 16, 24 bpp barevné obrázky, včetně 8 bpp obrázků ve stupních šedi. Nevýhodou je absence komprese (existuje varianta s kompresí RLE) soubory BMP tedy bývají mnohem větší než soubory ostatních formátů.

### 2.2 GIF

Formát GIF (Graphics Interchange Format) byl vytvořen v roce 1987 firmou CompuServe. Využívá bezztrátovou kompresi LZW, umožňuje vytváření animací. Nevýhodou je podpora maximálně 8 bpp, ale s podporou barevných palet (ty mohou být pro každý obrázek v animaci různé). Z výše uvedených vlastností vyplývá, že jeho použití je vhodné zejména pro ukládání grafiky (loga, nadpisy), zejména pak při použití na Internetu.

### 2.3 PNG

Formát PNG (Portable Network Graphics) byl představen konzorciem W3C v roce 1996. Využívá bezztrátovou kompresi Deflate. Oproti formátu GIF nabízí plné barvy až 48 bpp a 16bitový kanál pro průhlednost. Velkou výhodou je přenositelnost formátu, formát je hardwarově i softwarově nezávislý na platformě. Neumožňuje animace, ty přináší až formáty APNG a MNG. Formát je otevřený a nejsou s ním spojeny žádné licenční poplatky. Velké využití nachází na internetu.

### 2.4 TGA

Formát TGA (Targa) byl vytvořen firmou TrueVision Inc. v roce 1984. Jako první umožnil použití 24 bpp. Dovoluje použití 8bitového kanálu pro průhlednost. Může být nekomprimovaný, případně využívá bezztrátové kódování RLE v kombinaci s Huffmanovým kódem. Využíván zejména programátory pro ukládání textur.

## 2.5 TIFF

Formát TIFF (Tag Image File Format) byl vytvořen firmou Aldus v roce 1986. Umožňuje použití 24 bpp. Nabízí různé druhy komprese, např. RLE, LZW, CCITT, ZIP, JPEG. Umožňuje uložit více obrázků do jednoho souboru, proto nachází uplatnění při skenování a faxování.

## 2.6 JPEG

Standard JPEG vytvořila skupina *Joint Photographic Experts Group* (populární název pro pracovní skupinu WG1 z ISO/IEC Joint Technical Committee 1, Study Committee 9 ve zkratce ISO/IEC JTC1/SC29/WG1). Do standardu JPEG patří několik kompresních algoritmů, ztrátových i bezztrátových. Bezeztrátová verze se neujala, navíc stejná skupina později vytvořila nový bezztrátový kompresní standard JPEG-LS (Lossless). Ztrátová verze JPEG byla poprvé uvedena roku 1991 a během následujících let si tento standard získal dominantní pozici pro ukládání a přenos komprimovaných obrazů.

Je vhodný zejména pro fotografie, u kterých dosahuje vysoké komprese. Základem je předpoklad, že sousední pixely mají obvykle podobnou barvu. Tento předpoklad zpravidla platí pro fotografii, ale ne pro uměle vytvořenou grafiku. Ta obsahuje zejména ostré přechody mezi barvami a velké souvislé jednobarevné plochy. Vznikají tak i při malé kompresi viditelné artefakty. Jejich vznik je patrný z principu komprese (viz část 3.1.3).

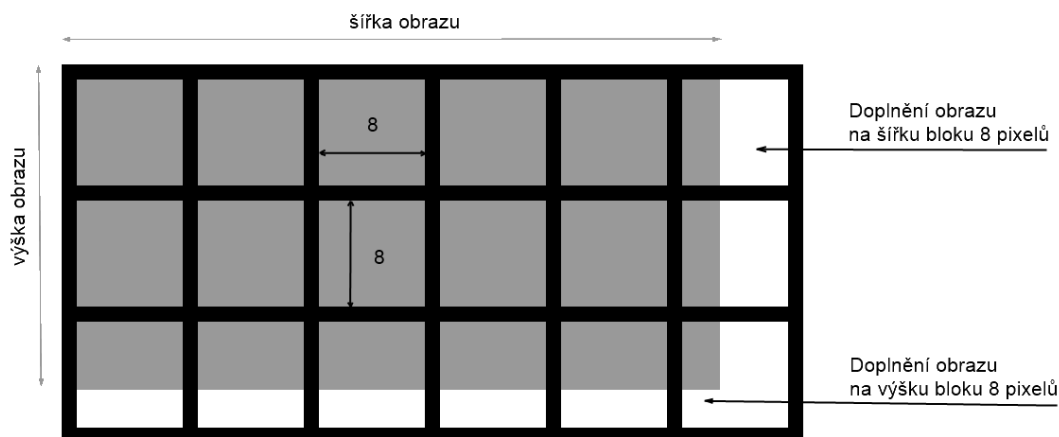
## 2.7 JPEG2000

Stejná skupina vytvořila v roce 2000 i kompresní standard JPEG-2000, založený na diskrétní vlnkové transformaci (DWT). Jeho cílem bylo zejména nabídnout vysoký kompresní poměr při nízkých bitových rychlostech, což se podařilo. Ale vzhledem k již značnému rozšíření standardu JPEG a nejspíš i díky složitosti a výpočetní náročnosti algoritmu standardu JPEG-2000 se neujal.

## 2.8 WebP

WebP je nejnovější formát popsáný v této práci. Vyvinula jej firma Google Inc. v roce 2010. Nabízí ztrátovou i bezztrátovou kompresi rastrových obrázků. V porovnání s PNG nabízí dle [13] až o 26% menší velikost souboru v případě bezztrátové komprese a v porovnání s formátem JPEG o 25-34% menší velikost v případě ztrátové

komprese. Podporuje jak průhlednost tak i animace. Díky dobré kompresi je vhodný k nahrazení ostatních grafických formátů. Nevýhodou je jeho rozšíření, ne všechny grafické programy tento formát implementovaly, nicméně tato situace se každým dnem zlepšuje.



Obr. 3.1: Rozdělení obrazu do bloků o velikosti  $8 \times 8$  pixelů

## 3 ROZBOR OBRAZOVÝCH FORMÁTŮ

Následující kapitoly podrobněji rozeberou dnes nepoužívanější obrazové formáty.

### 3.1 Popis formátu JPEG

#### 3.1.1 Transformace barev

Základní algoritmus komprese JPEG pracuje v barevném prostoru YCbCr (viz část 1.1.2), proto je prvním krokem algoritmu převod do tohoto modelu. Pokud je zdrojový obraz v barevném modelu RGB, je použito k přepočtu vztahu (1.1). Dále jsou jednotlivé složky obrazu zpracovávány samostatně.

#### 3.1.2 Redukce barev

Jelikož je lidské oko daleko citlivější na změny jasu než barev, je možné barevnou složku redukovat a docílit tak značnou úsporu velikosti obrazu. Nejčastěji se používá model 4:2:0 (YUY12), který zachová jasovou složku beze změny a u barevných složek je každému superpixelu  $2 \times 2$  přiřazena průměrná hodnota barvy těchto 4 pixelů. Tímto procesem dojde k redukci šířky pásma na polovinu při minimálním dopadu na lidské vnímání obrazu. Je možné použít i modely 4:4:4, 4:2:2 a 4:1:1.

#### 3.1.3 DCT (Diskrétní kosinová transformace)

Jasová složka obrazu je rozdělena do bloků o velikosti  $8 \times 8$  pixelů. Stejně tak barevná složka je rozdělena do bloků o velikosti  $8 \times 8$  pixelů (vzhledem k předchozí

transformaci reprezentují blok o velikosti  $16 \times 16$  pixelů původního obrazu). Pokud rozměry obrazu nejsou přesné násobky 8, bloky na spodním a pravém okraji nebudou plně obsazeny. V tomto případě musí být neúplné bloky doplněny na velikost  $8 \times 8$  pixelů (obr. 3.1) a následně jsou zpracovány jako bloky ostatní. A na těchto blocích je následně provedena dvojrozměrná diskretní kosinová transformace (2D-DCT). Výsledkem transformace je převedení obrazu z prostorové oblasti do oblasti frekvenční. Výsledný blok sestává ze stejnosměrné složky (DC) v levém horním rohu a 63 střídavých složek (AC). Pro výpočet 2D-DCT lze užít vztahu

$$S_{uv} = \frac{1}{4} C_u C_v \sum_{i=0}^7 \sum_{j=0}^7 s_{ij} \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16}, \quad (3.1)$$

kde

$$C_u = C_v = \frac{1}{\sqrt{2}} \text{ pro } u = 0 \text{ a } v = 0$$

$$C_u = C_v = 1 \text{ pro } u \neq 0 \text{ a } v \neq 0$$

a  $S_{uv}$  je frekvenční koeficient v transformovaném bloku na pozici  $uv$ .

### 3.1.4 Kvantizace

Získané transformované bloky jsou následně kvantovány, což je proces, při kterém jsou DCT koeficienty vyděleny kvantizační maticí. Uživatel má možnost zvolit kvalitu kvantizace zvolením hodnoty koeficientu  $q\_JPEG$ , který nabývá hodnot od 1 do 100, přičemž 100 znamená nejvyšší kvalitu. Dle zvoleného koeficientu se ke kvantizaci použije příslušná kvantizační matice. Kvantizační matice jsou odlišené pro jasovou a barevnou složku. Kvantování probíhá pomocí vzorce

$$\mathbf{F}^q(u, v) = \left[ \begin{array}{c} \mathbf{F}(u, v) \\ \mathbf{Q}(u, v) \end{array} \right], \quad (3.2)$$

kde

$u$  a  $v$  značí horizontální respektive vertikální frekvenci,

$\mathbf{Q}(u, v)$  reprezentuje prvky kvantizačních tabulek pro jasovou a barevné složky,

$\mathbf{F}(u, v)$  reprezentuje koeficienty před kvantizací

a výsledné hodnoty jsou zaokrouhleny k nejbližšímu celému číslu.

Tab. 3.1: Kvantizační tabulka pro jasovou složku

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Tab. 3.2: Kvantizační tabulka pro barevné složky

17	18	24	47	99	99	99	99
18	24	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	24	47	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Standardní matice při nastavení  $q\_JPEG = 50$  jsou v uvedeny tabulkách 3.1 a 3.2. Pro jiné nastavení kvality, jsou hodnoty matice vynásobeny kompresním faktorem  $\alpha$ , definovaným jako

$$\alpha = \frac{50}{q\_JPEG} \quad \text{pro } 1 \leq q\_JPEG \leq 50, \quad (3.3)$$

$$\alpha = 2 - \frac{2 \times q\_JPEG}{100} \quad \text{pro } 50 \leq q\_JPEG \leq 99.$$

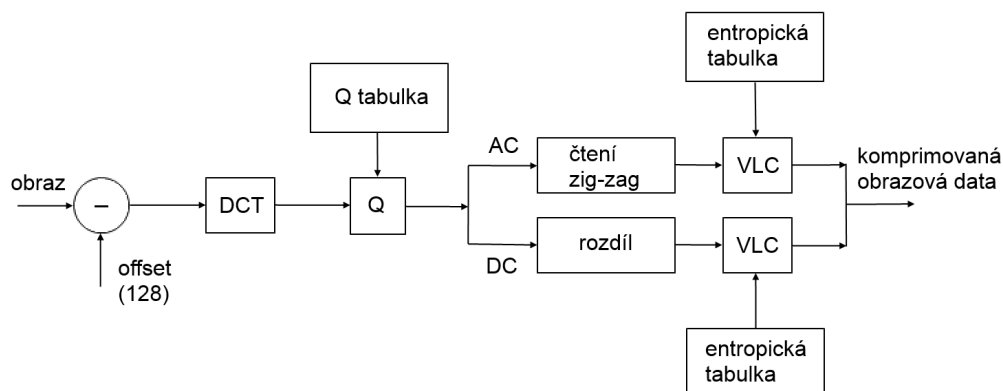
Při nastavení  $q\_JPEG = 100$  jsou hodnoty v kvantizačních tabulkách rovny 1.

### 3.1.5 Kódování

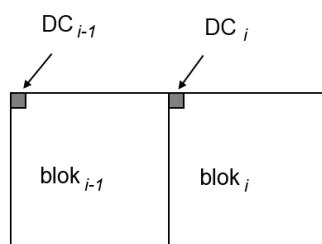
Po kvantování jsou stejnosměrný koeficient (DC) a 63 střídavých koeficientů (AC) kódovány samostatně jak je znázorněno na obr. 3.2.

#### Kódování DC koeficientu

Hodnota stejnosměrného koeficientu je získána z rozdílu DC koeficientu současného a předešlého bloku, jak je znázorněno na obr. 3.3, viz  $DIFF = DC_i - DC_{i-1}$ . Samostatný přístup ke kódování je zvolen z toho důvodu, že DC koeficient zpravidla



Obr. 3.2: Blokový diagram baseline JPEG enkodéru



Obr. 3.3: Rozdílové kódování DC koeficientu

nese největší část obrazové energie a tímto postupem je zjištěna vzájemná korelace DC koeficientů, která umožní efektivnější kódování. Pro kódování je následně zvolen VLC (Variable Length Codeword – Kód s proměnlivou délkou kódového slova) kód, např. Huffmanův kód. Není však zvoleno kódové slovo pro každou hodnotu  $DIFF$ , ale tyto hodnoty jsou rozřazeny do kategorií (CAT) a zakódována je až tato hodnota. Rozdělení do kategorií a k nim přiřazený Huffmanův kód je znázorněno v tabulce 3.3. CAT hodnota 0 znamená konec bloku (EOB – End of Block).

Samotné kódování je složeno ze seřazeného páru  $(c, u)$ ,  $c$  je kategorie CAT,  $u$  obsahuje přesně zbylých  $c$  bitů potřebných k zakódování symbolu. Pokud je  $c \neq 0$ ,  $u$  musí nést informaci o znaménku, v opačném případě je kódovaný symbol nulový a není třeba žádných dalších bitů. Pokud je  $s > 0$ ,  $u$  obsahuje 1 následovanou  $c - 1$  nejméně významnými bity (LSB – Least significant bits) absolutní hodnoty kódovaného symbolu. Pokud je  $s < 0$ ,  $u$  obsahuje 0 následovanou jednotkovým doplňkem  $c - 1$  nejméně významnými bity absolutní hodnoty kódovaného symbolu.

### Kódování AC koeficientu

Pro kódování střídavého (AC) koeficientu je zvolena metoda vyčítání *zig-zag*, jak je znázorněno na obr. 3.4. Koeficienty jsou seřazeny vzestupně podle frekvencí, vyšší

Tab. 3.3: kategorie (CAT) a DC Huffmanovi koeficienty pro jasovou složku

CAT	Rozsah	Kódové slovo
0	-	00
1	-1, 1	010
2	-3, -2, 2, 3	011
3	-7, ..., -4, 4, ..., 7	100
4	-15, ..., -8, 8, ..., 15	101
5	-31, ..., -16, 16, ..., 31	110
6	-63, ..., -32, 32, ..., 63	1110
7	-127, ..., -64, 64, ..., 124	11110
8	-255, ..., -128, 128, ..., 255	111110
9	-511, ..., -256, 256, ..., 511	1111110
10	-1023, ..., -512, 512, ..., 1023	11111110
11	-2047, ..., -1024, 1024, ..., 2047	111111110

frekvence bývají zpravidla nulové. Vznikne sekvence 63 po sobě jdoucích indexů, které označíme  $q[0] = AC[0, 1]$  až  $q[63] = AC[7, 7]$ . Na sekvenci je použito RLE (Run-length encoding), čímž vznikne sekvence o proměnné délce obsahující páry  $(r[i], v[i])$ , kde  $i$ -tý pár značí sled  $r[i]$  nul, následovanými hodnotou  $v[i]$ . Sled je omezen  $0 \leq r[i] \leq 15$ , takže sled 16 po sobě jdoucích nul bude reprezentován párem  $(15, 0)$ . Páry kde  $v[i] = 0$  a  $r[i] < 15$  jsou neplatné, s výjimkou páru  $(0, 0)$ , který značí EOB (Konec bloku). EOB vyjadřuje, že zbývající koeficienty jsou nulové.

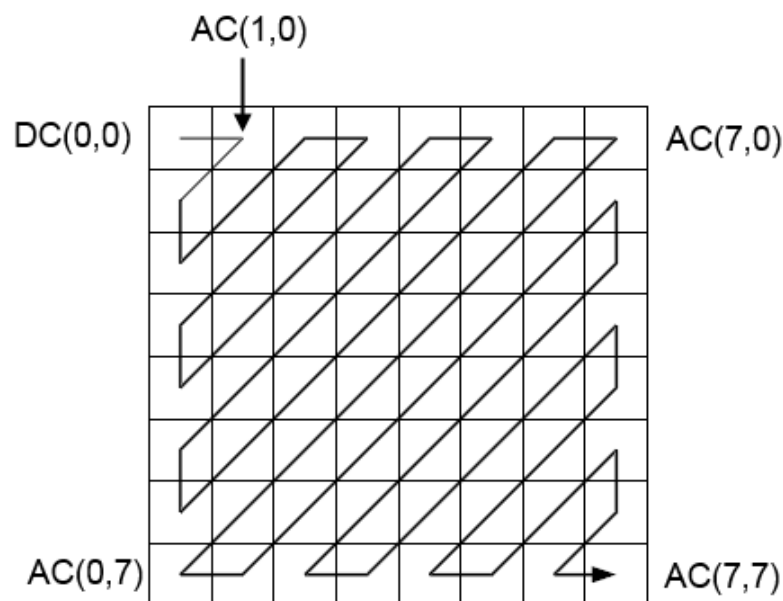
Páry jsou následně rozděleny do trojic  $(r[i], c[i], u[i])$ , kde  $(c[i], u[i])$  značí kód kategorie, podobně jako tomu bylo u DC koeficientu.  $r[i]$  i  $c[i]$  jsou reprezentovány 4 bity, mohou být tedy sloučeny do 8bitového slova, které je podrobena VLC (Variable Length Coding) kódování. Dle [9] je při aplikování na 8bitové obrazy výsledný počet kódových slov až 162, pro 12bitové obrazy až 226.

### 3.1.6 VLC – Variable Length Coding

Následuje entropické bezztrátové kódování, RLE kódování bylo již použito na DC i AC koeficienty. Dále lze volit mezi použitím Huffmanova kódování nebo aritmetického kódování.

#### RLE – Run-Length Encoding

Nejprve krátké doplnění principu RLE. Pracuje s délkou posloupnosti a její hodnotou. Vstupní data jsou kódována do dvojic (četnost, znak). Účinnost kódování



Obr. 3.4: čtení zig-zag AC koeficientů

je tedy velmi závislá na výskytu sekvencí stejných znaků. Tato metoda tedy není vhodná jako primární kódování pro JPEG a digitální fotografii. Používá se pouze jako pomocná metoda, před Huffmanovým nebo aritmetickým kódováním.

### Huffmanovo kódování

Huffmanův kód je velmi oblíbený a je považován za nejkratší kód pro minimalizaci nadbytečnosti. Kód patří do skupiny kódů s proměnlivou délkou kódového slova, kódové slovo může být i 1 bit. Znaky které se v kódované zprávě vyskytují nejčastěji se kódují s co nejmenším počtem znaků. Naopak znaky, které se vyskytují méně často, jsou zakódovány s větším počtem znaků.

Prvním krokem algoritmu je zjištění četnosti znaků. Za pomoci těchto informací, vytvoří kodér binární strom. Nejprve se vyhledají 2 znaky s nejvyšší četností, jejich pravděpodobnosti se sečtou a vznikne tak znak nový. K původním znakům je doplněna 0 a 1 tak, že k symbolu s menší pravděpodobností zapíšeme 0 a druhému 1 (případně je lze označit i opačně, jen je nutné postupovat stejným způsobem na celý strom). Pro snazší pochopení je kódování znázorněno v tabulce 3.4, kde  $s_i$  je kódovaný znak a  $p_i$  je četnost výskytu daného znaku. Takto je vytvořena tabulka s kódovými slovy a je přiložena do JPEG souboru pro zpětné dekódování. Standard JPEG nabízí připravené Huffmanovy tabulky, zmíněné v [3], ačkoliv kodér si může vytvořit své vlastní, efektivnější pro konkrétní obrázek.

Tab. 3.4: Ukázka Huffmanova kódování

$s_i$	$p_i$	I	$s_i$	$p_i$	II	$s_i$	$p_i$	III	$s_i$	$p_i$	IV
$s_4$	0,45		$s_4$	0,45		$s_4$	0,45		$s_4$	0,45	0
$s_1$	0,35		$s_1$	0,35		$s_1$	0,35	1	$s_{1235}$	0,55	1
$s_5$	0,1		$s_5$	0,1	1	$s_{235}$	0,2	0			
$s_2$	0,07	1	$s_{23}$	0,1	0						
$s_3$	0,03	0									

$$s_1 \rightarrow 11 \quad s_2 \rightarrow 1001 \quad s_3 \rightarrow 1000 \quad s_4 \rightarrow 0 \quad s_5 \rightarrow 101$$

Tab. 3.5: Ukázka Aritmetického kódování

$s_i$	$p_i$	$a_i$	$1 - \log_2 p_i$	$n$	$2^n a_i$	$c$	kód
$s_1$	0,55	0	1,862	2	0	1	01
$s_2$	0,3	0,55	2,737	3	4,4	5	101
$s_3$	0,15	0,85	3,737	4	13,6	14	1110

### Aritmetické kódování

Aritmetické kódování se snaží stejně jako Huffmanovo, zakódovat četné znaky s co nejmenším počtem znaků. Rozdílný je přístup k návrhu značky, který je individuální. To je výhodné pro případy, kdy se změní její výskyt, lze modifikovat část kódu, aniž by bylo potřeba přepočítat všechny značky znovu. Díky tomu je často mnohem efektivnější než Huffmanův kód. Vstupní slovo je zakódováno do reálného čísla v rozsahu  $(0, 1)$ .

**Postup aritmetického kódování dle [6]:**

1. Pro každé  $i$  určíme  $n$  tak, aby bylo nejbližším vyšším celým číslem:  $n \geq 1 - \log_2 p_i$ .
2. Určíme celé číslo  $c$ :  $c - 1 \leq 2^n a_i < c$ .
3. Číslo  $c$  převedeme na kódové slovo.

### 3.1.7 Dekódování

Dekódování formátu JPEG pracuje v naprosto opačném pořadí jako kódování. Nejprve bezztrátové dekodování za pomoci Huffmanových tabulek, dále RLE dekodování AC a DC koeficientů, vynásobení koeficientů příslušnou kvantizační maticí, zpětná kosinová diskretní transformace, nadzvorkování barvonosných složek a nakonec převod z YCbCr barevného modelu do námi požadovaného modelu pro zobrazení, obvykle RGB.

## 3.2 Popis formátu JPEG2000

Kroky kódování formátu JPEG lze rozdělit na předzpracování, kompresi a kompresi výstupního datového toku. Samotný formát JPEG2000 se dělí do mnoha částí, z toho pouze první část je přístupná bezplatně a zahrnuje základní práci s formátem. Další části jsou využívány komerčně k velmi specifickým úkonům a mnoho těchto částí je stále ve vývoji. Práce se zabývá pouze částí 1 – Core Coding System publikované v mezinárodním standardu ISO/IEC 15444-1:2000.

### 3.2.1 DC Level Shifting – posun stejnosměrné složky

Pokud jsou obrazová data uložena v nezáporných hodnotách (unsigned integer), je nejprve třeba převést je na hodnoty soustředěné kolem nuly (signed integer). Pokud máme obraz  $\mathbf{I}(x, y)$ , potom od každého vzorku odečteme stejnou hodnotu dle vzorce

$$\mathbf{I}'(x, y) = \mathbf{I}(x, y) - (2^{p-1}) \quad (3.4)$$

kde  $p$  značí počet bitů na kanál.

### 3.2.2 Transformace barev

Stejně jako u formátu JPEG i tento formát vyžaduje převod z barevného prostoru RGB do jiného vhodnějšího prostoru. Nabízí se dvě možnosti, rozhoduje jestli použijeme ztrátovou nebo bezztrátovou kompresi.

#### RCT – Vratná barevná transformace

Vratnou barevnou transformaci lze použít na ztrátovou i bezztrátovou kompresi, zejména z důvodu použití celočíselných koeficientů transformace a tedy i možností při inverzní RCT rekonstruovat obraz zpět. Z tohoto důvodu je využíván zejména pro bezztrátovou kompresi. Pro transformaci se používá vzorců:

$$\begin{aligned} Y_r &= \left\lfloor \frac{R + 2G + B}{4} \right\rfloor, \\ U_r &= B - G, \\ V_r &= R - G, \end{aligned} \quad (3.5)$$

$$\begin{aligned} G &= Y_r + \left\lfloor \frac{U_r - V_r}{4} \right\rfloor, \\ R &= V_r + G, \\ B &= U_r + G. \end{aligned} \quad (3.6)$$

## ICT – Nevratná barevná transformace

Nevratnou barevnou transformaci jak je již z názvu patrné, lze použít pouze na ztrátovou kompresi. Koeficienty transformace jsou reálná čísla, nelze tak zpětně získat přesně rekonstruovaný původní obraz, ale dochází k menším chybám. Převod je velmi podobný jako u formátu JPEG, pouze s tím rozdílem, že převod stejnosměrné složky byl již proveden, takže se ve vzorci vynechá:

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,168736 & -0,331264 & 0,5 \\ 0,5 & -0,418688 & -0,081312 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}, \quad (3.7)$$

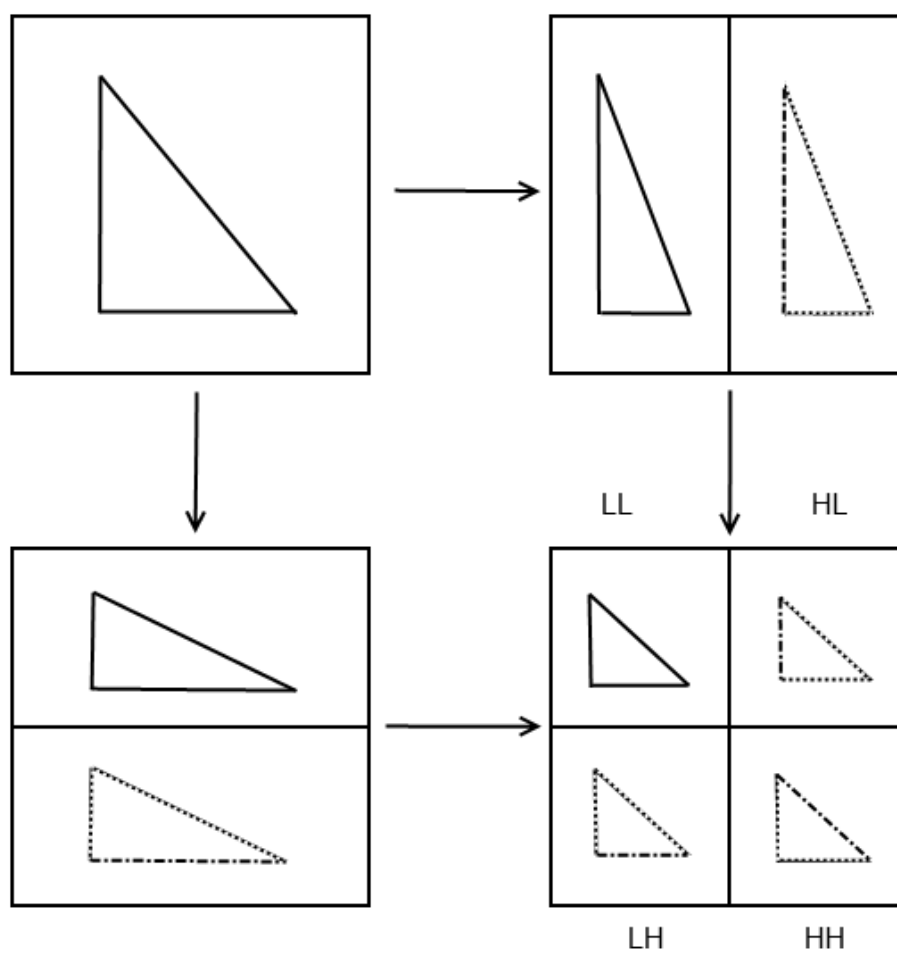
$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1,0 & 0,0 & 1,402 \\ 1,0 & -0,344136 & -0,714136 \\ 1,0 & 1,772 & 0,0 \end{pmatrix} \begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix}. \quad (3.8)$$

### 3.2.3 DWT – Diskrétní vlnková transformace

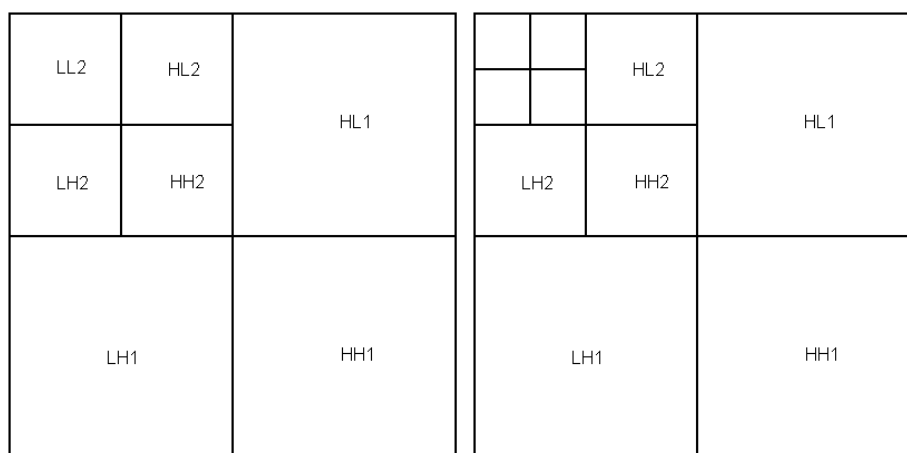
Kompresi obrazu probíhá také ve třech krocích, prvním krokem je diskrétní vlnková transformace. Je to klíčový rozdíl mezi formátem JPEG. Výhodou DWT je impulsní charakter a tak lépe zaznamená hrany a kontrasty v obrazu. DWT analyzuje jednotlivé dlaždice obrazu a rozloží je do několika subpásem s různými úrovněmi rozlišení. Na obrazy se používá 2D-DWT (dvourozměrná diskrétní vlnková transformace), princip je patrný z obr. 3.5. Zjednodušeně probíhá jeden krok 2D-DWT dle [1] následovně:

1. Nejprve aplikujeme 1D-DWT po řádcích.
2. Vzniknou 2 obrazy o poloviční šířce než původní obraz. Jeden je složený z vysokofrekvenčních složek (H – obsahuje zejména hrany), druhý z nízkofrekvenčních složek (L).
3. Na vzniklé obrazy následně zvlášť aplikujeme 1D-DWT tentokrát po sloupcích.
4. Z každého obrazu opět vzniknou 2 obrazy o poloviční výšce, rozdělené na H a L složky.
5. Výsledkem celého kroku jsou pak 4 obrazy (LL, LH, HL, HH) a každý obsahuje přesně čtvrtinu koeficientů původního obrazu.

Krok lze provést i v opačném pořadí, nejprve filtrovat po sloupcích a až poté po řádcích, výsledek bude ale totožný. Rozložení obrazu druhé a třetí úrovně je zobrazeno na obr. 3.6. Subpásma LL1 je hrubší verze původního obrazu (zmenšeného v poměru 2:1) a subpásma LH1, HL1, HH1 obsahují rozdílové složky, potřebné k rekonstrukci původního obrazu. Je možné použít dekompozici až do úrovně 32, avšak část 1 standardu JPEG2000 podporuje maximálně druhou úroveň.



Obr. 3.5: Dvourozměrná diskretní vlnková transformace



Obr. 3.6: Vlnková dekompozice druhé a třetí úrovně

### CDF 9/7 (Cohen-Daubechies-Feauveau)

Pro ztrátovou kompresi je základním vlnkovým filtrem *Daubechies (9, 7) biorthogonal spline filter*. U filtru čísla (9, 7) značí počet dekompozičních/rekonstrukčních koeficientů pro horní a dolní propust. Koeficienty jsou vyjádřeny reálnými čísly, jejich hodnoty zaokrouhleny na 6 desetinných míst zobrazuje tabulka 3.6.

Tab. 3.6: Koeficienty CDF biortogonálních vlnkových filtrů

Koeficienty	CDF(9/7)		CDF(5/3)	
	DP	HP	DP	HP
0	0.602949	1.115087	3/4	1
$\pm 1$	0.266864	-0.591272	1/4	-1/2
$\pm 2$	-0.078223	-0.057544	-1/8	
$\pm 3$	-0.016864	0.091272		
$\pm 4$	0.026729			

### CDF 5/3 (Cohen-Daubechies-Feauveau)

Pro bezztrátovou kompresi je základním vlnkovým filtrem *Daubechies (5, 3) bi-orthogonal spline filter*. Koeficienty jsou vyjádřeny celými čísly, jejich hodnoty zobrazuje tabulka 3.6.

#### 3.2.4 Kvantizace

U bezztrátové komprese je kvantizace vynechána, jelikož je to hlavní proces, při kterém dochází k nevratné ztrátě informací. U ztrátové komprese kvantizace nejvíce ovlivňuje kvalitu a velikost výsledného obrázku. Používá se uniformní kvantizace, kde  $\Delta b$  značí kvantovací krok a  $y_b(i, j)$  jsou DWT koeficienty subpásma  $b$ . Pomocí vzorce

$$q_b(i, j) = \text{sign}(y_b(i, j)) \left\lfloor \frac{|y_b(i, j)|}{\Delta b} \right\rfloor \quad (3.9)$$

získáme kvantované DWT koeficienty  $q_b(i, j)$ , které jsou celočíselné se znaménkem. Je možné zvolit rozdílný kvantizační krok pro každé subpásma.

#### 3.2.5 ROI – Region of Interest Coding

Region zájmu umožňuje zakódovat různé oblasti obrazu odlišnou kvalitou. Tyto oblasti mohou mít libovolný tvar a také je lze spojovat mezi sebou. Je tak možné určit hlavní objekt v obrazu, ten zakódovat v nejlepší kvalitě. Na zbylé pozadí je pak

použita větší komprese a dochází k úspoře velikosti obrázku při zachování důležitých částí v potřebné kvalitě. Tato technika se nazývá *Maxshift*.

### 3.2.6 Entropické kódování

Závěrečným krokem komprese je entropické kódování, zvoleno bylo kódování EBCOT (Embedded Block Coding with Optimised Truncation). Každé subpásmo obrazu je rozděleno do menších obdélníkových bloků, které jsou kódovány zvlášť. Typické velikosti bloku jsou  $32 \times 32$  a  $64 \times 64$ . Výsledný počet koeficientů nesmí přesáhnout 4096, přičemž výška i šířka musí být minimálně 4, maximálně 1024 a jsou voleny z mocnin dvou. Bloky koeficientů jsou kódovány bit po bitu, nejprve od nejvýznamnějšího bitu (MSB – Most Significant Bit) po nejméně významný bit (LSB – Least Significant Bit). Každá bitová rovina je kódována třemi průchody, mezi ně patří *significant propagation pass*, *magnitude refinement pass* a *cleanup pass*.

#### Significance Propagation Pass (SPP)

Zakódovány jsou bitové pozice, které obsahují jedničku v aktuální bitové rovině.

#### Magnitude Refinement Pass (MRP)

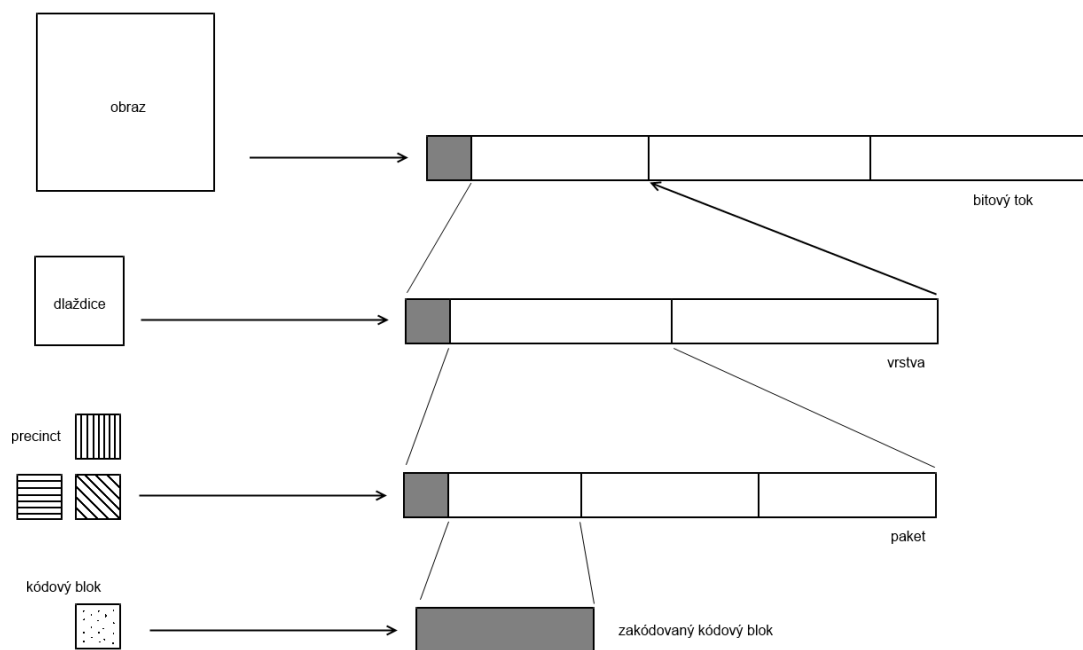
Zakódovány jsou bitové pozice, které nebyly kódovány v SPP, ale v předchozích rovinách obsahovaly jedničku.

#### Cleanup Pass (CUP)

Zakódovány jsou zbylé bitové pozice, které nebyly kódovány v SPP ani v MRP.

### 3.2.7 Komprese výstupního datového toku

Jakmile je celý obraz zkomprimován dochází k formování bitového toku. Nejprve jsou tři prostorově konzistentní kódové bloky (jeden z každého subpásma pro každé rozlišení) složeny do tzn. precinctu. Data ze tří kódových bloků tvořících precinct vytvoří paket s hlavičkou, která adresuje pozici precinctu v obraze. Pakety se dále sdružují do vrstev, která je také označena hlavičkou a adresována. Několik vrstev následně formuje bitový tok. Vrstva představuje celý obraz v určitém rozlišení nebo kvalitě, kdežto paket představuje jen část obrazu v určitém rozlišení nebo kvalitě. Každá vrstva tak zlepšuje výsledný obraz, lze tak dosáhnout progresivního zobrazování obrazu. Hierarchie datového toku je zobrazena na obr. 3.7.



Obr. 3.7: Souvislost mezi bitovým tokem a prostorovými daty

## 3.3 Popis formátu PNG

### 3.3.1 Příprava PNG obrázku

Na zdrojový obraz je nejprve aplikováno několik transformací, z nichž některé kroky jsou volitelné.

#### Alpha separation – oddělení alfa kanálu

Oddělení alfa kanálu je volitelný krok. Pokud veškeré složky alfa kanálu obsahují maximální hodnotu, může být alfa kanál z obrazu vynechán.

#### Indexování barev

Pokud obraz s paletou barev truecolor (24 nebo 32bitový obraz) využívá maximálně 256 barev, je možné použít indexování barev. Je vytvořena paleta barev, v které každá kombinace RGB složek je označena indexem. Jednotlivé vzorky v obraze potom odkazují na daný index barvy v paletě. Dojde tím k značné úspoře, jelikož každý vzorek je reprezentován pouze jedním bytem, namísto tří, respektive čtyř bytů pokud obraz obsahuje i průhlednost. Paleta je často seřazena vzestupně podle hodnot alfa kanálu, je tak možné vynechat maximální hodnotu alfa kanálu a uspořít další cenné byty.

Volitelný krok místo indexování je *RGB sloučení (merging)*. Pokud jednotlivé barevné kanály mají stejnou bitovou hloubku a pro každý pixel jsou hodnoty červené, modré i zelené složky shodné, pak je možné barevné kanály sloučit do jednoho kanálu se stupni šedi.

Další volitelný krok, použitelný pro neindexované obrazy se nazývá *Alpha compaction*. Pokud obraz obsahuje alfa kanál a pro jednu kombinaci RGB platí, že veškeré pixely jsou průhledné a naopak pro ostatní plně viditelné, potom je možné alfa kanál vynechat a zapsat pouze informaci o kombinaci RGB která je průhledná.

### Sample depth scaling – Změna bitové hloubky

Formát PNG podporuje pouze některé bitové hloubky obrazu. Kompletní seznam podporovaných kombinací zobrazuje tabulka 3.7. Pokud referenční obraz je v jiné bitové hloubce, zvolí se nejbližší vyšší bitová hloubka a hodnoty jsou lineárně přepočteny do nových hodnot bitové hloubky.

Tab. 3.7: Podporované kombinace typu barvy a bitové hloubky formátu PNG

Typ PNG obrazu	Typ barvy	Podporované bitové hloubky	Interpretace
Stupně šedi	0	1, 2, 4, 8, 16	Každý pixel je vzorek stupňů šedi
Truecolour	2	8, 16	Každý pixel obsahuje trojici hodnot RGB
Indexované barvy	3	1, 2, 4, 8	Každý pixel je zastoupen indexem paletové barvy
Stupně šedi s průhledností	4	8, 16	Každý pixel je vzorek stupňů šedi a hodnoty alfa kanálu
Truecolour s průhledností	6	8, 16	Každý pixel obsahuje trojici hodnot RGB a hodnotu alfa kanálu

### 3.3.2 Prokládání pixelů

PNG umožňuje dvě varianty prokládání. První je bez prokládání, tedy pixely jsou z obrázku vyčítány nejprve po řádcích zleva doprava. Výslednému souboru bytů se říká *scanline*, ta je pak vyčítána od horní ke spodní. Pokud je obraz uložen v 16bitové hloubce na kanál, pixel každého kanálu je zaznamenán 2 byty v pořadí *big endian* (nejprve MSB, pak LSB).

Druhá metoda, nazývaná *Adam7* vyčítá pixely obrazu v sedmi průchodech. První průchod vytvoří nejhrubší náhled obrazu a každý další jej zjemňuje. Přesný sled

1	6	4	6	2	6	4	6
7	7	7	7	7	7	7	7
5	6	5	6	5	6	5	6
7	7	7	7	7	7	7	7
3	6	4	6	3	6	4	6
7	7	7	7	7	7	7	7
5	6	5	6	5	6	5	6
7	7	7	7	7	7	7	7

Obr. 3.8: Prokládání pixelů metodou Adam7 u formátu PNG

c	b
a	x

Obr. 3.9: Poloha bytů použitých při filtrování

pixelů na oblasti  $8 \times 8$  pixelů ukazuje obr. 3.8. Tímto způsobem lze během přenosu již s  $1/64$  dat (po kompletním přenosu prvního průchodu prokládání) zobrazit hrubý náhled obrázku.

### 3.3.3 Filtrování

Před samotnou kompresí, je ještě provedeno filtrování. Snahou je připravit obraz tak, aby výsledná komprese byla lepší. PNG formát specifikuje 5 různých filtrů, je možné použít rozdílné filtry na každý řádek redukovaného obrazu (obraz vzniklý průchodem prokládání řádků). Nicméně na výsledný obraz by měl být použit pouze jeden filtr. Filtry již operují přímo s byty, nikoliv pixely obrázku. Označení pozic bytů použitých při filtrování je zobrazeno na obr. 3.9. Pokud byte  $x$  je na prvním pixelu *scanline*, pak byty  $a$  a  $c$  mají hodnotu nula. Stejně tak pokud se byte  $x$  nachází v první *scanline*, jsou byty  $b$  a  $c$  nulové.

#### Filtr typu 0 – None

Prvním filtrem, je vlastně *žádný*, byty jsou ze vstupu přesně zkopírovány na výstup a nedochází k žádné změně. Možná je to nepochopitelné, ale pro některé obrázky je to nejvýhodnější a je tak dosaženo největší komprese.

### Filtr typu 1 – Sub

Filtr typu 1 od aktuálního bytu odečítá byte nalevo od něj. Tedy  $x = x - a$ . Při rekonstrukci obrazu se rozdíl nahradí součtem, tedy  $x = x + a$ .

### Filtr typu 2 – Up

Filtr typu 2 od aktuálního bytu odečítá byte nad ním. Tedy  $x = x - b$ . Při rekonstrukci obrazu se rozdíl nahradí součtem, tedy  $x = x + b$ .

### Filtr typu 3 – Average

Filtr typu 3 kombinuje dva předchozí filtry a odečítá průměrnou hodnotu bytů nalevo a nad aktuálním bytem. Tedy  $x = x - \lfloor (a - b)/2 \rfloor$  a zaokrouhuje se na dolní celou část. Při rekonstrukci obrazu se rozdíl nahradí součtem, tedy  $x = x + \lfloor (a - b)/2 \rfloor$ .

### Filtr typu 4 – Paethův

Poslední z řady filtrů je nejsložitější, přináší to zvýšení výpočetní náročnosti a tedy i délku výpočtu. Porovnává absolutní hodnoty získané dle (3.10) a volí z  $pa$ ,  $pb$  a  $pc$  tu nejnižší. Ta je následně odečtena od původního bytu. Při rekonstrukci obrazu se opět rozdíl nahradí součtem.

$$\begin{aligned} p &= a + b - c \\ pa &= |(p - a)| \\ pb &= |(p - b)| \\ pc &= |(p - c)|. \end{aligned} \tag{3.10}$$

## 3.3.4 Kompresce

Ke kompresi se používá algoritmus *Deflate*, pro dekompresi *Inflate*. Algoritmus kombinuje kompresi LZ77 a Huffmanovo kódování 1.1.2. Kompresce LZ77 hledá v bloku dat ke kompresi duplikovanou sérii bytů. Jakmile nějakou nalezne, je nahrazena značkou obsahující délku série a vzdáleností k předchozí shodě. Maximální možná vzdálenost shodných sérií je 32768 bytů. Následuje Huffmanovo kódování, které dále sníží počet bytů. Výsledný datový tok je uložen ve formátu *zlib*.

## 3.3.5 Chunking

Formát PNG uchovává data v souboru rozdělených do částí, nazývané *chunky*. Ještě před samotnými chunky je v souboru obsažena hlavička, ta slouží k identifikaci

formátu a detekci nejběžnějších chyb při přenosu. Hlavička se skládá z následujících dekadických hodnot: *137 80 78 71 13 10 26 10*. První byte obsahuje hodnotu, která v nejvyšším bitu má jedničku, slouží k detekci chyb. Následují tři byty obsahují ASCII znaky *PNG*, dále symboly CR, LF (Cursor Return, Line Feed), dále speciální znak pro ukončení textového výpisu a opět znak LF.

Tab. 3.8: Význam velikosti znaků v názvu typu chunku

Byte	Hodnota bitu 5	Význam
1	0	povinný
	1	volitelný
2	0	veřejný
	1	privátní
3	0	odpovídá verzi PNG
	1	rezervován pro možné rozšíření
4	0	unsafe to copy
	1	safe to copy

Chunk se skládá ze čtyř částí. V první části o velikosti 4 byty je uložena celková délka datové části chunku. V druhé části chunku je uložen jeho typ. Ten je identifikován přesně čtyřmi znaky anglické abecedy v ASCII formátu, přičemž se rozlišují velká a malá písmena. Právě z velikosti znaku lze určit určité parametry, jak je vidět v tabulce 3.8. Jelikož je použit formát ASCII, kde velká písmena mají dekadický význam 65–90, malá písmena 97–122, stačí ke zjištění velikost písmene pouze přečtení hodnoty pátého bitu daného bytu. Třetí částí chunku jsou samotná data. Poslední částí chunku je cyklický redundantní součet (CRC), obsahující 4 byty. Je vypočítán na základě části s typem chunku a ze samotných dat. K jeho výpočtu je použit polynom  $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ .

### 3.3.6 Popis jednotlivých chunků

Možné typy chunků, jejich počet v souboru a pořadí popisuje tabulka 3.9.

#### **IHDR – Image header**

Hlavička obrázku, je odlišná od hlavičky souboru. Obsahuje základní informace o obrázku: rozměry obrázku, bitovou hloubku, typ barvy dle 3.7, metoda komprese, metoda filtrace, prokládání. V současném standardu musí být metoda filtrace i metoda komprimace nastavena na nulu. Musí být umístěna ihned za hlavičkou souboru.

Tab. 3.9: Typy chunků uvedené v jakém pořadí by se měly nacházet

Název chunku	Počet	Důležitost	Poznámka
IHDR	1	ano	
cHRM	0-1	ne	
qAMA	0-1	ne	
iCCP	0-1	ne	Pouze není-li sRGB
sBIT	0-1	ne	
sRGB	0-1	ne	Pouze není-li iCCP
PLTE	1	ano	Pouze obsahuje-li obrázek barevnou paletu
bKGD	0-1	ne	
hIST	0-1	ne	
tRNS	0-1	ne	
pHYs	0-1	ne	
sPLT	0 a více	ne	
IDAT	1 a více	ano	Pokud jich je více, musí být řazeny za sebou
tIME	0-1	ne	
iTXt	0 a více	ne	
tEXt	0 a více	ne	
zTXt	0 a více	ne	
IEND	1	ano	

### **PLTE – Palette**

Pokud obrázek obsahuje paletu barev, je uložena v tomto chunku.

### **IDAT – Image data**

Samotný obrázek je uložen zde, soubor musí obsahovat minimálně jeden IDAT chunk.

### **IEND – Image Trailer**

Označuje konec PNG souboru.

### **tRNS – Transparency**

Tento chunk obsahuje údaje o průhlednosti obrázku. V případě paletového obrázku obsahuje tabulku s hodnotami alfa kanálu pro dané indexy palety.

### **cHRM – Primary chromaticities and white point**

Obsahuje informace o pozicích primárních složek RGB a referenčního bílého bodu v obrázku.

### **gAMA – Image gamma**

Obsahuje poměr mezi vzorky obrázku a výstupního jasů.

### **iCCP – Embedded ICC profile**

Zde je obsažen barevný profil ICC, Pokud je tento chunk v souboru obsažen, musí mít alespoň jeden profil.

### **sBIT – Significant bits**

Slouží pro obnovu plného rozsahu bitové hloubky, pokud došlo k její změně.

### **sRGB – Standard RGB colour space**

Udává použitý sRGB barevný profil.

### **tEXt, iTXt a zTXt – Textual information**

Umožňuje zapsat do souboru různé textové informace.

### **bKGD – Background colour**

Udává základní barvu pozadí obrázku.

### **hIST – Image histogram**

Histogram obrázku, je možný pouze pro indexované obrázky.

### **pHYs – Physical pixel dimensions**

Udává fyzické rozměry pixelů. Vhodné například při tisku.

### sPLT – Suggested palette

Obsahuje informace o vhodných paletách, doporučených pro zobrazení obrázku.

### tIME – Image last-modification time

Časový údaj poslední změny obrazových dat.

## 3.4 Popis formátu WebP

Moderní formát WebP stále bojuje s větším rozšířením, stejně jako tomu je i u formátu JPEG2000. Velkou výhodou je fakt, že jeho algoritmus není tak složitý a výpočetně náročný jako v případě JPEG2000. A tak je možné, že v budoucnu se stane nástupcem formátů JPEG, GIF a PNG. Vše se bude odvíjet od zavedení podpory tohoto formátu velkými výrobci obrazové techniky.

Pro ukládání souborů formátu WebP byl zvolen formát RIFF (Resource Interchange File Format) a podobně jako u formátu PNG, rozděluje data do *chunků*. Prvním elementem v souboru je hlavička, v které první uloženou informací je identifikace dokumentu RIFF v ASCII formátu. Dále následuje 32bitové číslo udávající celkovou velikost souboru a poslední částí je identifikace formátu WEBP v ASCII formátu. Dále již následují jednotlivé chunky identifikované znaky *VP8*, respektive *VP8L* v případě bezztrátové komprese.

### 3.4.1 Ztrátový WebP

Základem ztrátové komprese formátu WebP je predikce bloků, obraz je tedy rozdělen do menších částí nazývaných makrobloky. V každém makrobloku, je pak hledána nadbytečná barevná a pohybová informace od předchozích makrobloků. Tato nadbytečná data jsou následně od bloku odečtena a zůstanou pouze rezidua, na ně je následně aplikována diskretní kosínová transformace. Vznikne spousta nulových koeficientů, které se lépe komprimují. Data jsou kvantována a entropicky zakódována. Jediný ztrátový krok v případě WebP je kvantizace.

#### Intra-frame Prediction

Predikce bloků pracuje se třemi druhy makrobloků:  $4 \times 4$  jas,  $16 \times 16$  jas,  $8 \times 8$  barvonosné složky. Barvonosné složky jsou redukovány stejně jako u formátu JPEG, jedinou možností je model 4:2:0 (viz část 3.1.2). Velikost bloků s jasovou složkou je volena na základě množství detailů ve scéně, na detailní části se použije  $4 \times 4$  makrobloků.

Samotná predikce nabízí 4 režimy:

- H\_PRED (Horizontální predikce) – každý sloupec bloku naplní sloupcem, který sousedí s blokem nalevo
- V\_PRED (Vertikální predikce) – každý řádek bloku naplní řádkem, který se nachází nad blokem
- DC\_PRED (DC predikce) – celý blok zaplní jednou barvou, kterou získá na základě průměru sloupce sousedícího nalevo a řádku nad blokem
- TM\_PRED (TrueMotion predikce) – Používá predikci pohybu na základě řádku nad blokem, sloupce vlevo a pixelu vlevo nad blokem.

U makrobloků  $4 \times 4$  je možné H\_PRED i V\_PRED rozšířit do dalších 6 směrů. Vytvořený predikční snímek je odečten od skutečné části a je uložen pouze rozdíl, který podléhá dalším procesům.

## DCT/WHT

Na vzniklá residua je většinou aplikována diskrétní kosinová transformace (viz část 3.1.3), výjimečně se použije Walsh–Hadamardova transformace. Obě transformace jsou prováděny s pevnou desetinou čárkou, aby byla minimalizována ztráta informací.

## Adaptivní kvantizace

Pro zlepšení výsledné kvality obrazu, umožňuje formát WebP použít adaptivní kvantizaci. Obraz je rozdělen do vizuálně podobných segmentů a každý tento segment může být komprimován s odlišnými parametry. Povoleny jsou maximálně 4 segmenty v obraze.

## Entropy coding

Jako entropické kódování bylo zvoleno aritmetické kódování, jeho princip byl nastíněn v kapitole 3.1.6.

## 3.4.2 Bezeztrátový WebP

Základem bezeztrátového kódování jsou transformace obrazu s použitím různých technik. Zakódovány jsou pak zejména parametry těchto technik s transformovanými obrazovými daty.

### **Spatial prediction transform**

Transformace s prostorovou predikcí využívá faktu, že sousedící pixely spolu často korelují. Hodnota aktuálního pixelu je predikovaná na základě již zakódovaných pixelů a je zakódován pouze rozdíl mezi aktuálním a predikovaným pixelem. Obraz je rozdělen do několika regionů a v rámci regionu lze použít pouze jeden druh predikce. Druhy predikce se rozdělují podle směru predikce, případně pokud používají zprůměrovanou hodnotu.

### **Color space transform**

Transformace barev dekoreluje RGB hodnoty každého pixelu. Zelená složka je ponechána v původní podobě, červená je transformována na základě zelené a modrá je transformována na základě zelené a pak červené.

### **Subtract green transform**

Tato transformace odečte hodnotu zelené složky od červené i modré složky.

### **Color Indexing Transform**

Technika naprosto shodná s indexováním u formátu PNG (viz část 3.3.1). Pokud obraz používá méně než 256 barev, je vytvořena paleta a vzorek odkazuje na index v paletě.

### **Kódování**

Po aplikaci transformací dochází k samotnému kódování. Ke kódování se používá varianty LZ77 a Huffmanova kódování (viz část 3.3.4).

## 4 HODNOCENÍ KVALITY KOMPRESY

Kvalitu komprimovaného obrazu můžeme posuzovat z objektivního nebo subjektivního pohledu. Subjektivní metody posouzení kvality popisuje [5]. Jedná se zejména o metody DSIS (Double Stimulus Impairment Scale), DSCQS (Double Stimulus Continuous Quality Scale) a MOS (Mean Opinion Score).

DSIS je metoda vhodná, pokud disponujeme referenčním obrazem, na základě kterého chceme porovnat jiné, méně kvalitní obrazy. Během hodnocení má před sebou hodnotící zároveň referenční i posuzovaný obraz a je mu známo, který obraz je referenční (metoda předpokládá, že posuzovaný obraz je méně kvalitní). Hodnotící vybírá na 5stupňové škále.

DSCQS je metoda relativní. Je vhodná pro případ, kdy nemáme k dispozici referenční obraz. Hodnotící má před sebou opět 2 obrazy, ale tentokrát neví, který je referenční. Hodnotí tedy kvalitu obou obrazů.

MOS metoda vyžaduje více pozorovatelů, kteří opět hodnotí kvalitu obrazu na 5stupňové škále. MOS vyjadřuje průměr těchto hodnocení za předpokladu, že podmínky hodnocení se nijak nemění.

Nevýhodou subjektivních metod je jejich časová náročnost a potřeba mít vyčvičené pozorovatele pro zajištění spolehlivosti výsledků. Objektivní metody tyto problémy eliminují, jejich popisu budou věnovány následující sekce.

### 4.1 MSE – střední kvadratická chyba

Střední kvadratická chyba je využívána několik desetiletí v oboru zpracování a analýzy signálů, jako kvalitativní prostředek. Základní výhodou MSE je její jednoduchost. Nevyžaduje žádné parametry, nepotřebuje paměť na výpočty, každý vzorek může být vyhodnocen nezávisle a zabere málo výpočetního času. I když tato metoda může být při posouzení kvality obrazového signálu přesná, interpretace obrazového signálu je od tohoto schématu odlišná. Nedokáže posoudit např. náhodné přemístění pixelů v obrazu, což může způsobit obrovské změny ve vnímání obrazu, který tak může být přeměněn až v šum.

Pokud máme originální obraz  $\mathbf{x} \equiv [n_1, n_2]$  a komprimovaný obraz  $\hat{\mathbf{x}} \equiv [\hat{n}_1, \hat{n}_2]$ , oba o shodné šířce  $N_1$  a výšce  $N_2$ , pak můžeme pomocí vzorce

$$\text{MSE} = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (x[n_1, n_2] - \hat{x}[n_1, n_2])^2 \quad (4.1)$$

získat hodnotu střední kvadratické chyby komprimovaného obrazu. Pokud jsou oba obrazy shodné, výsledkem je nula, naopak čím vyšší výsledek vyjde, tím jsou obrazy odlišnější. Pokud posuzujeme barevný obraz v RGB, vypočteme MSE pro všechny 3

barvy, sečteme a podělíme 3. V barevném prostoru YCbCr posuzujeme každý kanál samostatně. V příkladech budeme posuzovat pouze jasovou složku.

## 4.2 PSNR – špičkový poměr signálu k šumu

Špičkový poměr signálu k šumu vyjadřuje poměr mezi maximální možnou energií signálu a energií šumu. Vychází z MSE, pouze upravuje měřítko do reálnějšího výsledku. Jelikož obrazy mohou mít široké dynamické spektrum, je výsledek vyjádřen na logaritmickém měřítku. Obvyklá hodnota PSNR pro 24bitové obrazy je mezi 30 a 50 dB, vyšší hodnota znamená kvalitnější obraz. Pokud jsou obrazy shodné, není PSNR definované, jelikož MSE je nulové. PSNR můžeme vypočítat pomocí vztahu

$$\text{PSNR} = 10 \log_{10} \frac{(2^B - 1)^2}{\text{MSE}}, \quad (4.2)$$

kde  $B$  značí dynamický rozsah, který je při 8 bitech na kanál 255.

## 4.3 SSIM – Structural Similarity

Obě předešlé metody nerespektují psychovizuální model lidského vidění. Tuto problematiku bere v potaz právě metoda SSIM a snaží se vytvořit index, který se bude nejvíce blížit lidskému pozorování. Metoda jak je popsáno v [11] vychází z předpokladu, že přirozené obrazy jsou vysoce strukturované, pixely nesou mezi sebou silnou závislost, zvláště pokud jsou blízko u sebe a tato závislost nese důležitou informaci o struktuře objektů na scéně. Také bylo potřeba oddělit tyto strukturní informace od jasu, jelikož struktura objektů je na jasů nezávislá. Metoda tedy rozděluje měření do 3 srovnání: jas (luminance), kontrast a struktura.

Nejprve je porovnána hodnota jasu obou signálů. Předpokládejme diskrétní signály, potom to můžeme odhadnout pomocí střední intenzity:

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i. \quad (4.3)$$

Funkce porovnání jasu  $l(\mathbf{x}, \mathbf{y})$  je potom funkcí  $\mu_x$  a  $\mu_y$ .

Když teď známe hodnotu jasu signálu, můžeme jej ze signálu odstranit. K odhadnutí kontrastu využijeme směrodatnou odchylku (kvadratický průměr odchylky od aritmetického průměru). Odhad lze získat pomocí vzorce

$$\sigma_x = \left( \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right)^{\frac{1}{2}}. \quad (4.4)$$

Porovnání kontrastu  $c(\mathbf{x}, \mathbf{y})$  je tedy porovnáním  $\sigma_x$  a  $\sigma_y$ .

Následně je signál normalizován o svou vlastní směrodatnou odchylku, tak může být porovnána struktura obrazů. Porovnání struktury  $s(\mathbf{x}, \mathbf{y})$  je realizováno na základě těchto normalizovaných signálů  $\frac{(\mathbf{x}-\mu_x)}{\sigma_x}$  a  $\frac{(\mathbf{y}-\mu_y)}{\sigma_y}$ .

Nakonec získáme celkové srovnání zkombinováním všech 3 složek

$$S(\mathbf{x}, \mathbf{y}) = f(l(\mathbf{x}, \mathbf{y}), c(\mathbf{x}, \mathbf{y}), s(\mathbf{x}, \mathbf{y})). \quad (4.5)$$

Dále pro porovnání jasu je definován vzorec

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \quad (4.6)$$

kde konstanta  $C_1$  je přidána pro zamezení nestability, kdy se  $\mu_x^2 + \mu_y^2$  blíží k nule. Konstantu získáme pomocí vzorce

$$C_1 = (K_1L)^2, \quad (4.7)$$

kde  $L$  značí dynamický rozsah a je obvykle 255 pro 8 bitů na kanál,  $K_1$  je zpravidla rovna 0,01.

Podobně získáme i vzorec pro porovnání kontrastu

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \quad (4.8)$$

kde  $C_2 = (K_2L)^2$  a  $K_2$  je zpravidla 0,03.

Porovnání strukturních informací je provedeno po odečtení jasu a normalizaci signálu, jejich korelací získáme vzorec

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}, \quad (4.9)$$

samotnou korelaci získáme pomocí vzorce

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y). \quad (4.10)$$

Nyní můžeme všechny 3 složky zkombinovat a získáme konečný vzorec pro výpočet SSIM indexu

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}. \quad (4.11)$$

Pro vyhodnocení kvality obrazu, neporovnáváme SSIM index pro celý obraz, ale počítáme ho zpravidla pro okno o rozměrech  $8 \times 8$  pixelů. Je to hlavně z důvodu výpočetní náročnosti při velkých rozměrech obrazu. Čtvercové okno často způsobuje nežádoucí blokové artefakty, je možno použít jiný tvar okna. Toto okno následně posouváme po obraze po jednom pixelu. Ze všech dílčích výsledků následně uděláme

průměr a získáme celkový SSIM index obrazu, též nazýván jako MSSIM (mean SSIM – průměrný SSIM)

$$\text{MSSIM}(\mathbf{X}, \mathbf{Y}) = \frac{1}{M} \sum_{j=1}^M \text{SSIM}(\mathbf{x}_j, \mathbf{y}_j), \quad (4.12)$$

kde  $\mathbf{X}$  a  $\mathbf{Y}$  symbolizují referenční, respektive komprimovaný obraz,  $\mathbf{x}_j$  a  $\mathbf{y}_j$  obsah  $j$ -tého okna obrazu.

## 4.4 Další neimplementované metody

Následující metody nejsou již příliš používány v praxi a tak nejsou ani v testovacím programu implementovány.

### 4.4.1 NAE – Normalizovaná absolutní chyba

Pokud máme originální obraz  $\mathbf{x} \equiv [n_1, n_2]$  a komprimovaný obraz  $\hat{\mathbf{x}} \equiv [n_1, n_2]$ , oba o shodné šířce  $N_1$  a výšce  $N_2$ , pak můžeme pomocí vzorce

$$\text{NAE} = \frac{\sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (x[n_1, n_2] - \hat{x}[n_1, n_2])}{\sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (x[n_1, n_2])} \quad (4.13)$$

získat hodnotu normalizované absolutní chyby. Vyjadřuje poměr rozdílu pixelů dvou obrázků k referenčnímu obrázku.

### 4.4.2 NC – Normalizovaná korelace

Stejně definované proměnné můžeme použít ve vzorci

$$\text{NC} = \frac{\sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (x[n_1, n_2] - \hat{x}[n_1, n_2])}{\sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (x[n_1, n_2])^2} \quad (4.14)$$

a získáme hodnotu normalizované korelace. Vyjadřuje poměr rozdílu pixelů dvou obrázků k druhé mocnině referenčního obrázku.

### 4.4.3 AD – Průměrná odchylka

Tato metoda je téměř shodná s metodou MSE, nepočítá se druhá mocnina rozdílu dvou obrázků, ale pouze rozdíl:

$$\text{AD} = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (x[n_1, n_2] - \hat{x}[n_1, n_2]). \quad (4.15)$$

#### 4.4.4 MD – Maximální odchylka

Tato metoda pouze zjistí největší rozdíl mezi pixely dvou obrázků:

$$\text{MD} = \text{Max}(|x[n_1, n_2] - \hat{x}[n_1, n_2]|). \quad (4.16)$$

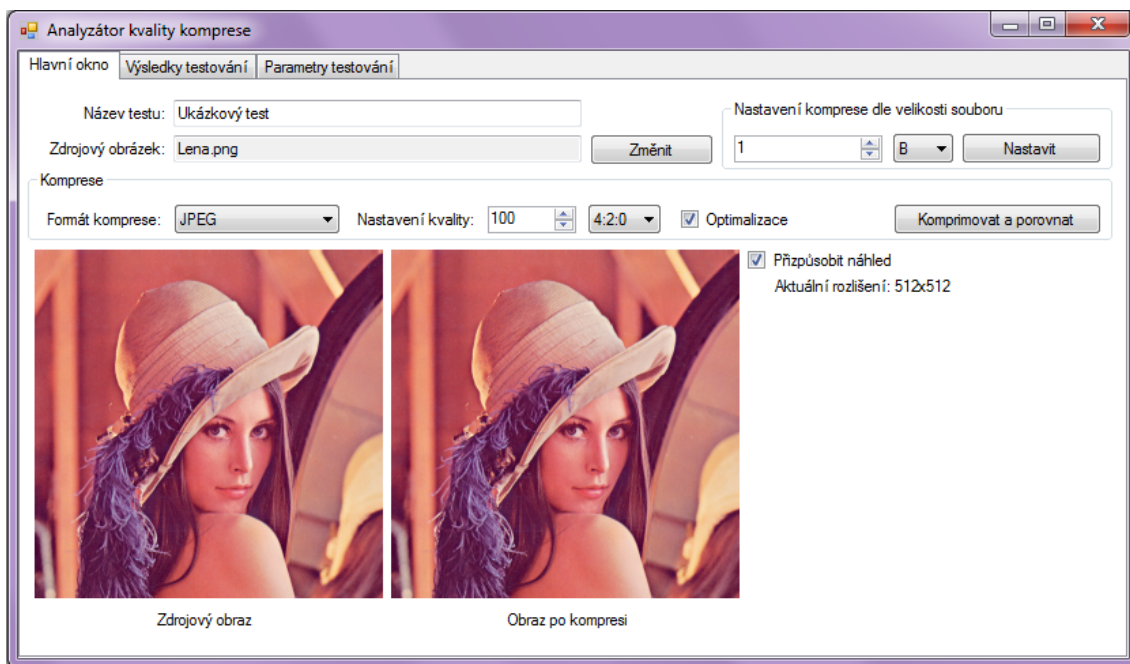
#### 4.4.5 NMSE – Normalizovaná střední kvadratická chyba

Pro tuto metodu je nejdříve nutné získat střední hodnotu obrázku  $\mu_x$  a jeho rozptyl  $\sigma_x$ , dále je shodná s metodou MSE, pouze se podělí rozptylem:

$$\mu_x = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x[n_1, n_2], \quad (4.17)$$

$$\sigma_x = \left( \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (x[n_1, n_2] - \mu_x)^2 \right)^{\frac{1}{2}}, \quad (4.18)$$

$$\text{NMSE} = \frac{1}{N_1 N_2 \sigma_x^2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (x[n_1, n_2] - \hat{x}[n_1, n_2])^2. \quad (4.19)$$



Obr. 5.1: Hlavní okno programu

## 5 PRAKTICKÁ REALIZACE

### 5.1 Analyzátor kvality komprese

Cílem práce bylo vytvořit program, který umožní načíst nekomprimovaný obraz, zkomprimovat jej zvoleným kompresním formátem a provést porovnání kvality komprese mezi původním obrazem a komprimovaným. V této části je popsán tento program, který byl vytvořen v programovacím jazyce C#, s využitím prostředí .NET. Projekt byl realizován v programu Microsoft Visual Studio 2012 ve verzi 11.0.61030.00 Update 4. Program je spustitelný v operačních systémech MS Windows, vyžaduje běhové prostředí .NET Framework verze 4.0.

Po spuštění programu je zobrazeno uživatelské prostředí a jeho hlavní okno (obr. 5.1). Do textového pole *název testu* je možné zadat libovolný text, kterým budou následně označeny uložené komprimované obrazy i výsledky testování, je možné pole ponechat prázdné. Tlačítkem *Změnit* se otevře výběr zdrojového obrázku na disku. Po vybrání platného obrázku dojde k jeho načtení do paměti a zobrazí se v levém spodním rohu programu. V textovém poli *Zdrojový obrázek* je zobrazen název souboru i s příponou aktuálně načteného obrázku.

V bloku *Komprese* je možné vybrat *Formát komprese*. Na výběr je ze čtyř formátů a to JPEG, JPEG2000, PNG a WebP. U všech formátů lze následně nastavit kvalitu komprese. U formátů JPEG a WebP je rozsah od 1 do 100, kde 100 značí

The screenshot shows a window titled 'Analyzátor kvality komprese' with three tabs: 'Hlavní okno', 'Výsledky testování', and 'Parametry testování'. The 'Výsledky testování' tab is active, displaying a table with the following data:

	Nazev testu	Zdrojovy obraz	Obraz po kompresi	MSE	PSNR	MSSIM
	Zkušební test 1	Lena.png	Testy/Zkušební test1_Komprimovany_Optimal_420...	64.77111533520...	30.01698985355...	0.815649004051...
▶*						

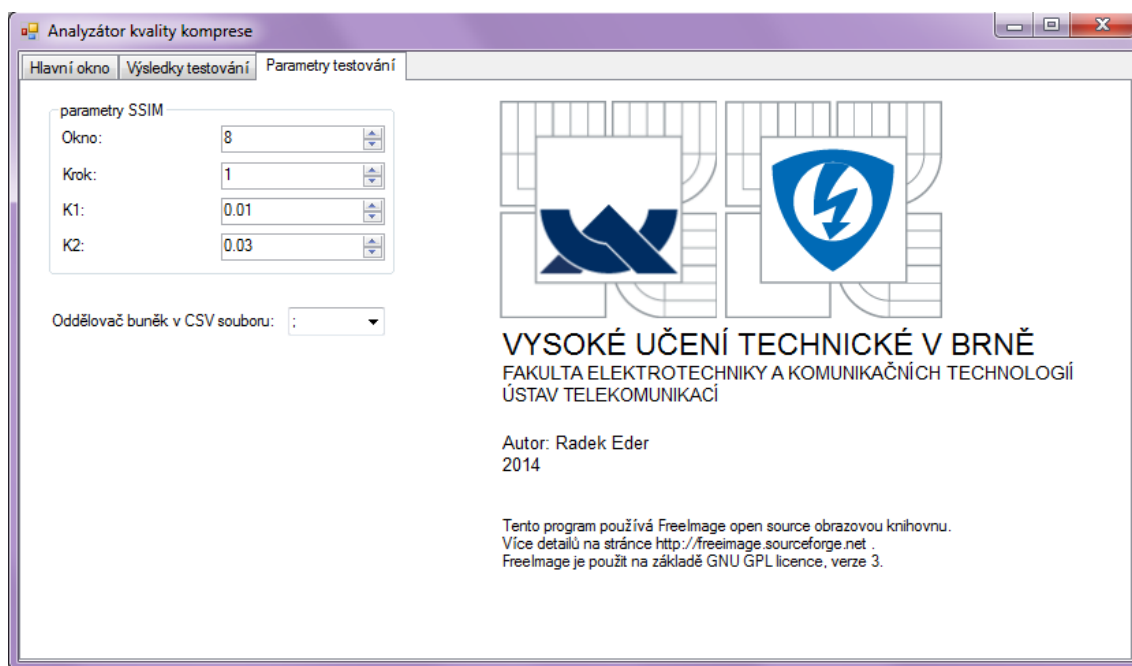
Below the table is a large greyed-out area and a button labeled 'Export do csv souboru'.

Obr. 5.2: Okno s výsledky testování

nejvyšší kvalitu obrázku. Formát PNG umožňuje nastavit rozsah od 0 do 9, kde 0 je nekomprimovaný obraz a 9 nejvyšší komprese. Formát JPEG2000 umožňuje rozsah kvality 1 až 512, kde 1 značí nejvyšší kvalitu obrázku. U formátu JPEG je navíc možné zvolit model redukce barev (viz část 3.1.2) a také pomocí zaškrtačacího políčka *Optimalizace* zvolit možnost vypočítat optimální Huffmanovy tabulky (viz část 3.1.6). Tlačítkem *Komprimovat a porovnat* se spustí proces komprimace, výsledný obrázek je uložen na disk a porovnán s původním obrázkem. Výsledky testu jsou zobrazeny v tabulce v okně *Výsledky testování* a aktuální obsah tabulky je uložen do CSV souboru na disku.

Parametr kvality komprese lze nechat určit programem, na základě zvolené výsledné velikosti souboru. Tato velikost se zadává do textového pole v bloku *Nastavení komprese dle velikosti souboru*. Tlačítkem *Nastavit* se spustí výpočet velikosti. Tento proces může chvíli trvat, dochází při něm k samotné kompresi a následnému měření velikosti komprimovaného souboru. Program to může několikrát zopakovat pro zjištění nejlepšího nastavení. Zjištěný parametr kvality komprese je nastaven a uživatel může spustit kompresi a porovnání.

V dolní části programu jsou vedle sebe zobrazeny původní nekomprimovaný obrázek a obrázek po kompresi. *Přizpůsobit náhled* slouží pouze k vizuální kontrole obrázku. Pokud je políčko zaškrtnuté, zmenší obrázek v náhledu tak, aby se vešel do okna 256 × 256 pixelů. V opačném případě je zobrazen v poměru 1:1 a části nad 256 pixelů nejsou zobrazeny. Tato volba nemá žádný vliv na kompresi nebo průběh



Obr. 5.3: Okno s parametry testování

testování.

V okně *Výsledky testování* (obr. 5.2), je zobrazena tabulka se všemi provedenými testy. Sloupec *Obraz po kompresi* obsahuje název komprimovaného obrázku a v jeho názvu jsou zapsány parametry komprese. Sloupce *MSE*, *PSNR* a *MSSIM* pak ukazují výsledky porovnání. Tlačítkem *Export do csv souboru* je obsah tabulky uložen do CSV souboru a je možné ho následně otevřít např. v programu Microsoft Excel a dále pracovat s daty.

V posledním okně s názvem *Parametry testování* (obr. 5.3), lze zvolit parametry pro metodu SSIM a dále oddělovač buněk formátu CSV, ten je totiž závislý od lokálního nastavení počítače a není nijak standardizován. Na běžném počítači s operačním systémem Microsoft Windows v českém jazyce jako oddělovač slouží středník, což je nastaveno při spuštění programu. Je možné jednak zvolit jiný oddělovač, nebo ho přímo zadat. Změny se projeví až při dalším exportu do souboru.



Obr. 6.1: Vzorové obrázky použité k testování (Lena, Baboon, Buildings, Plane)

## 6 VYHODNOCENÍ

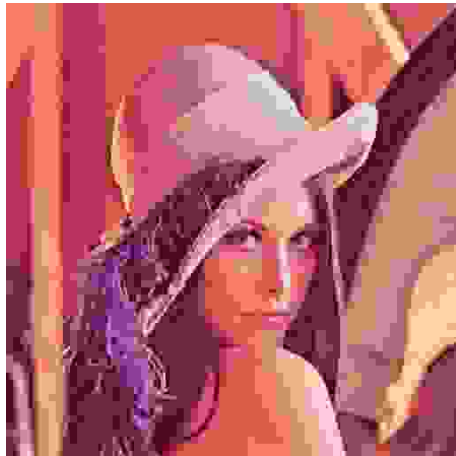
Tato kapitola vyhodnotí kodeky JPEG, JPEG2000 a WebP na referenčních obrázcích poskytnutých z databáze [8]. Jelikož formát PNG je bezztrátový, hodnocení kvality na něm nelze provést. Referenční obrázky jsou zobrazeny na obr. 6.1. Rozměry testovaných obrázků jsou  $512 \times 512$  a  $768 \times 512$  pixelů. Stejné rozměry byly zachovány i na samotné testování.

Testování proběhlo s parametry SSIM:

- Okno – strana čtverce testovaného okna, nastaveno na  $8 \times 8$  pixelů.
- Krok – udává o kolik pixelů se okno posune při další iteraci při výpočtu dle (4.12), nastaveno na 1 pixel.
- $K_1$  – koeficient ve vzorci (4.7), nastaven 0,01.
- $K_2$  – koeficient ve vzorci (4.7), nastaven 0,03.

Testování proběhlo při různé míře komprese, u obrázku *Lena* bylo otestováno chování kodeků při velmi malé velikosti souboru a tedy vysoké kompresi. U formátu JPEG při nastavení parametrů  $q\_JPEG = 1$  bylo dosaženo 0,09 bpp, dále  $q\_JPEG = 5$  bylo 0,132 bpp a nakonec při  $q\_JPEG = 10$  bylo dosaženo 0,230 bpp. U obrázku *Baboon* bylo otestováno chování kodeků při střední kompresi. U formátu JPEG při nastavení parametrů  $q\_JPEG = 20$  bylo dosaženo 0,771 bpp, dále  $q\_JPEG = 30$  bylo 1,053 bpp a nakonec při  $q\_JPEG = 50$  bylo dosaženo 1,511 bpp. Parametry komprese JPEG2000 a WebP byly zvoleny tak, aby bylo dosaženo shodné bpp. Formát WebP u obrázku *Lena* s nejvyšší kompresí nedokázal zajistit shodnou bpp, při jeho nejvyšší kompresi vychází 0,132 bpp. Na obr. 6.2, obr. 6.3 a 6.4 můžeme vidět srovnání kodeků JPEG, JPEG2000 a WebP. Také bylo provedeno komplexní porovnání formátů v celé škále nastavení kvality od  $q\_JPEG = 1$  po  $q\_JPEG = 100$  v inkrementech po 10 na všech čtyřech uvedených obrázcích. Parametry formátů WebP i JPEG2000 byly zvoleny opět tak, aby bylo dosaženo podobné velikosti souboru a tedy i bpp. Formátem WebP při nejvyšší kompresi se nepovedlo dosáhnout velikosti blízké formátu JPEG při nastavení kvality  $q\_JPEG = 1$ , z tohoto důvodu jsou následující grafy v tomto bodě u WebP zkráceny, jelikož výsledný soubor má rozdílnou bpp.

Ze získaných výsledků testů lze usoudit, že formát WebP skutečně dosahuje teoretických výkonů. Ve všech testech dosahuje v celém rozsahu nastavení kvality nejlepších výsledků jak v porovnání SSIM, tak i PSNR. Formát JPEG2000 dosahuje také výborných výsledků, zejména při nejvyšší kompresi. Předčil formát JPEG téměř ve všech testech, pouze u obrázku Baboon při nastavení  $q\_JPEG \geq 10$  dosáhl horších výsledků při testu SSIM, než formát JPEG. Dále u obrázku Lena se toto zhoršení projevilo při nastavení  $q\_JPEG \geq 40$ .



(a) PSNR = 24,4 dB, SSIM = 0,636



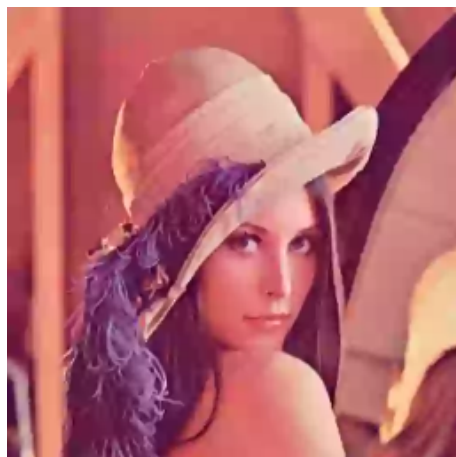
(b) PSNR = 27,412 dB, SSIM = 0,738



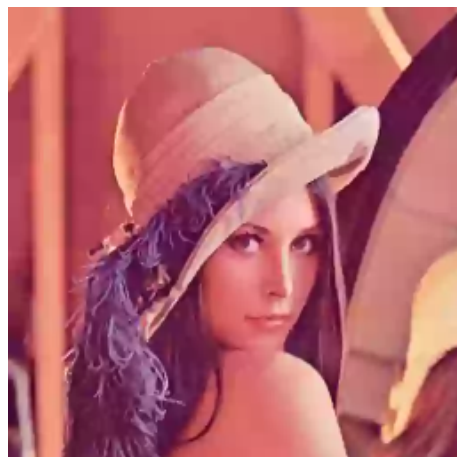
(c) PSNR = 27,841 dB, SSIM = 0,773



(d) PSNR = 29,527 dB, SSIM = 0,821



(e) PSNR = 29,660 dB, SSIM = 0,817

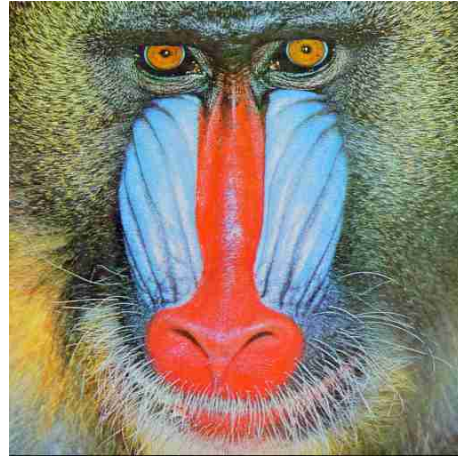


(f) PSNR = 30,077 dB, SSIM = 0,8272

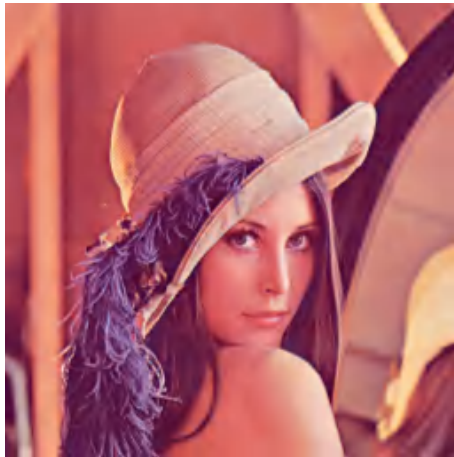
Obr. 6.2: Porovnání kompresních formátů při 0,09 bpp (vlevo) a 0,132 bpp (vpravo), (a) a (b) JPEG, (c) a (d) JPEG2000, (e) a (f) WebP



(a) PSNR = 30,472 dB, SSIM = 0,829



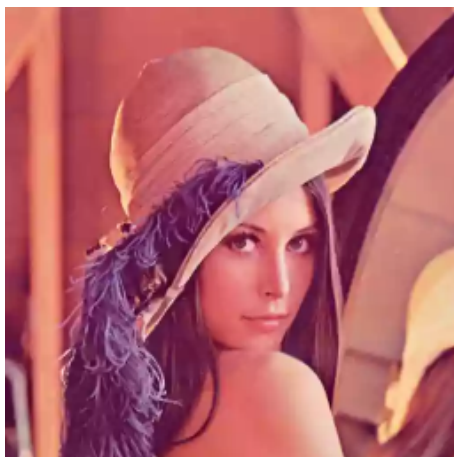
(b) PSNR = 25,28 dB, SSIM = 0,815



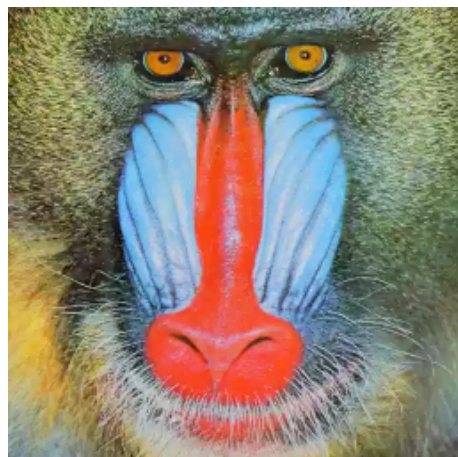
(c) PSNR = 31,608 dB, SSIM = 0,866



(d) PSNR = 26,42 dB, SSIM = 0,811



(e) PSNR = 32,588 dB, SSIM = 0,878

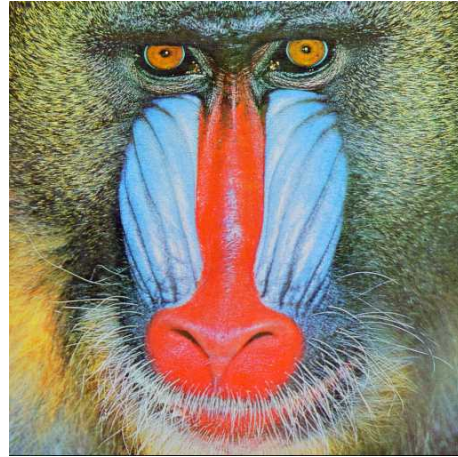


(f) PSNR = 26,54 dB, SSIM = 0,841

Obr. 6.3: Porovnání kompresních formátů při 0,23 bpp (vlevo) a 0,771 bpp (vpravo), (a) a (b) JPEG, (c) a (d) JPEG2000, (e) a (f) WebP



(a) PSNR = 26,462 dB, SSIM = 0,857



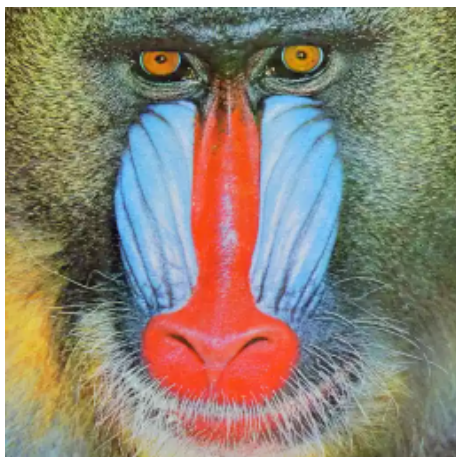
(b) PSNR = 28,241 dB, SSIM = 0,9



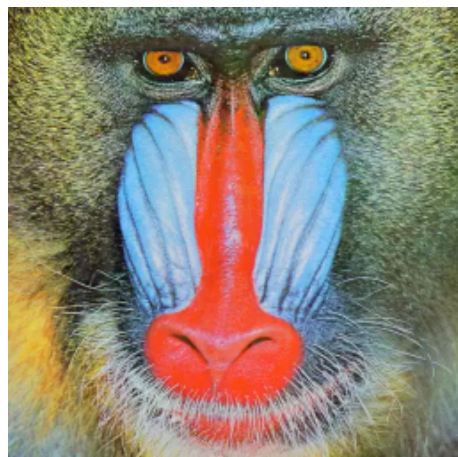
(c) PSNR = 27,721 dB, SSIM = 0,849



(d) PSNR = 29,608 dB, SSIM = 0,883

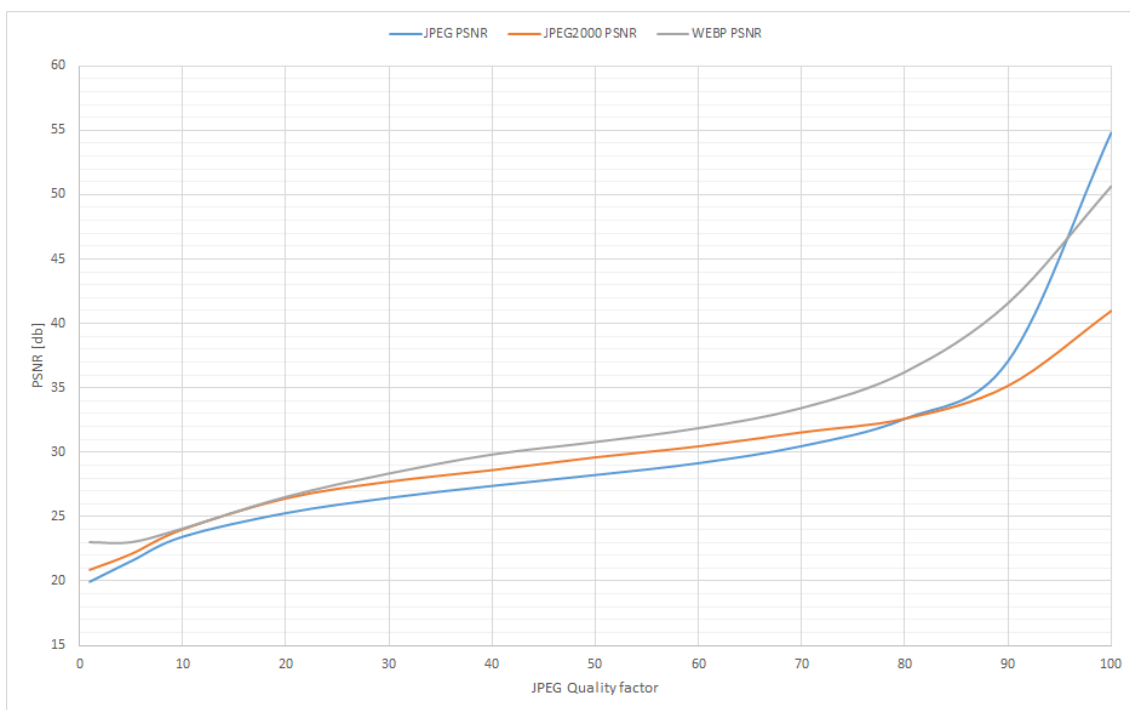


(e) PSNR = 28,356 dB, SSIM = 0,888

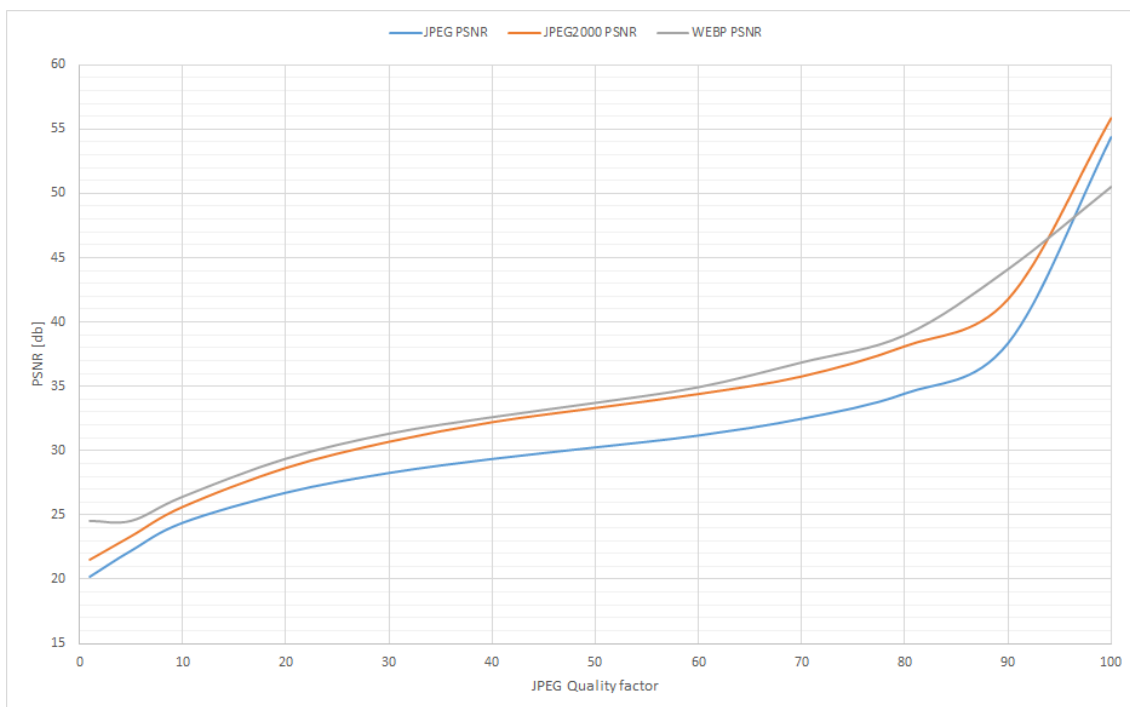


(f) PSNR = 30,796 dB, SSIM = 0,928

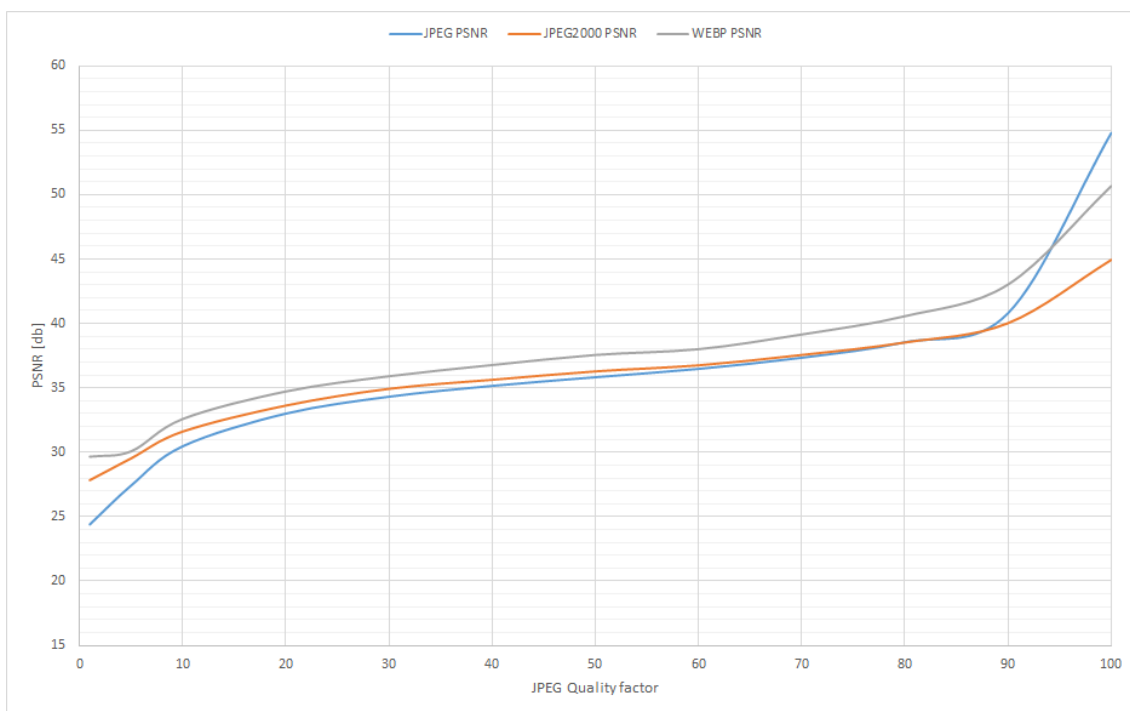
Obr. 6.4: Porovnání kompresních formátů při 1,053 bpp (vlevo) a 1,511 bpp (vpravo), (a) a (b) JPEG, (c) a (d) JPEG2000, (e) a (f) WebP



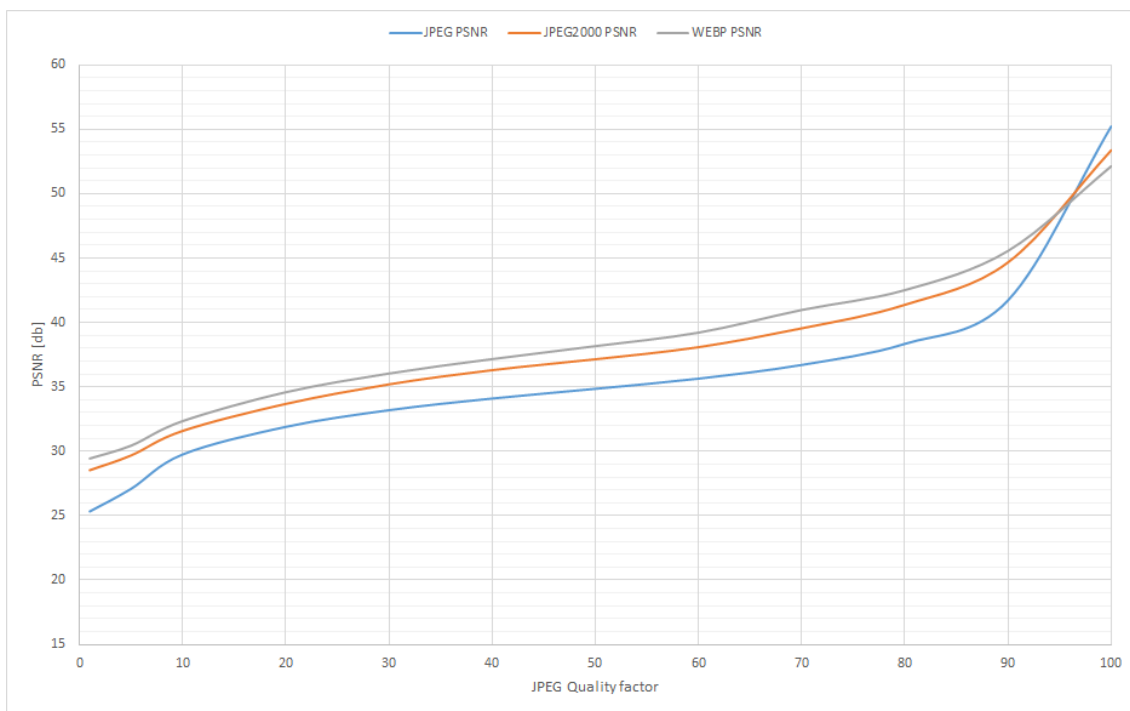
Obr. 6.5: Porovnání formátů měřením PSNR na obrázku Baboon



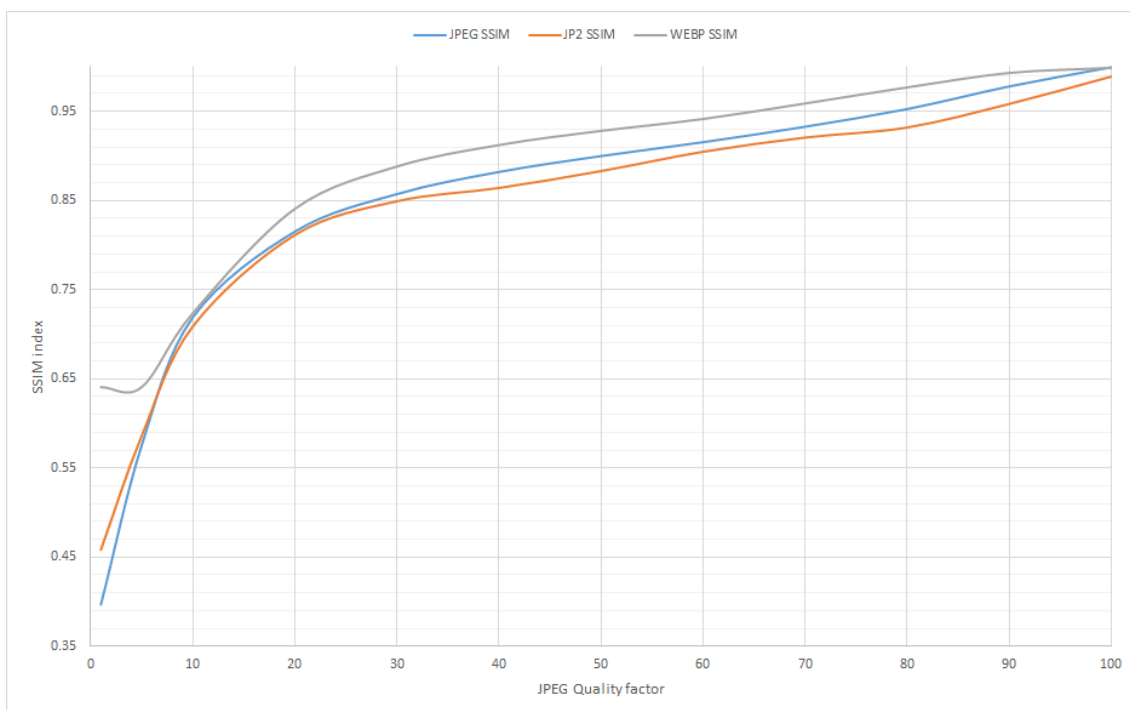
Obr. 6.6: Porovnání formátů měřením PSNR na obrázku Buildings



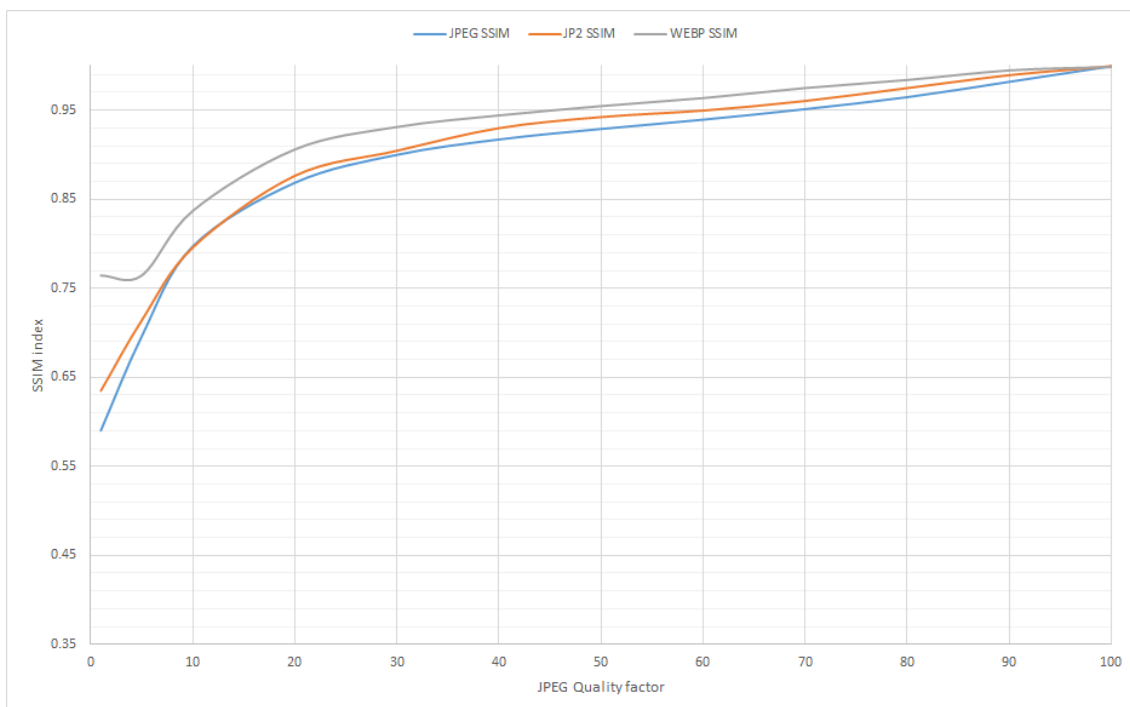
Obr. 6.7: Porovnání formátů měřením PSNR na obrázku Lena



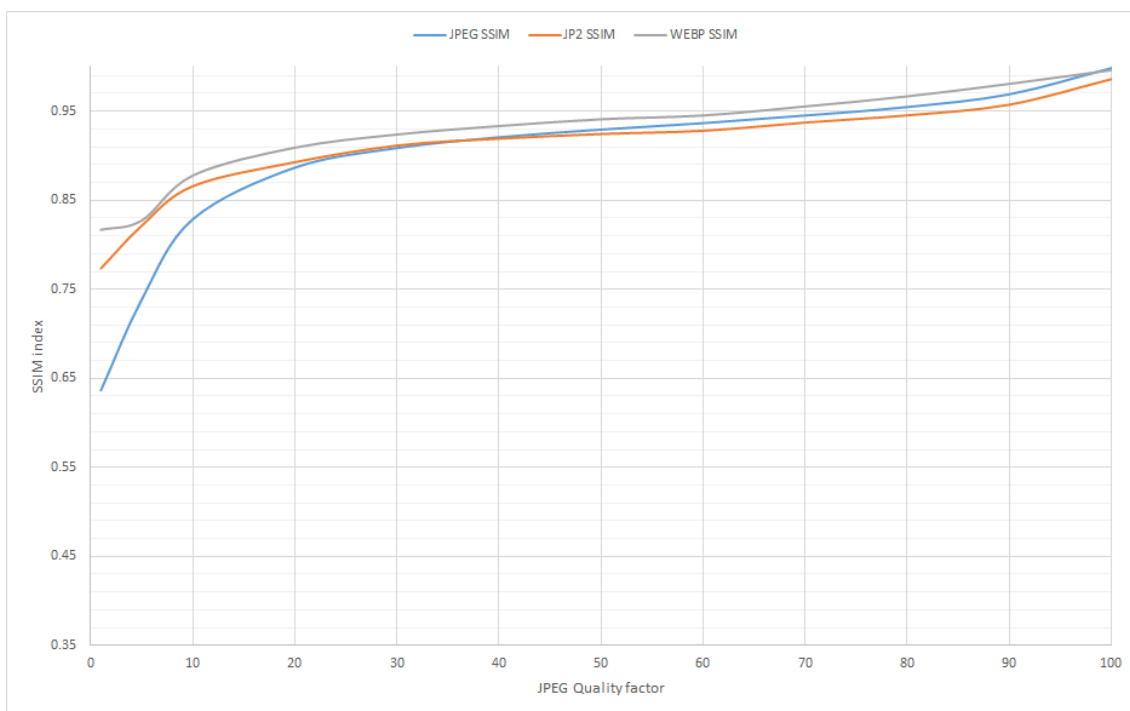
Obr. 6.8: Porovnání formátů měřením PSNR na obrázku Plane



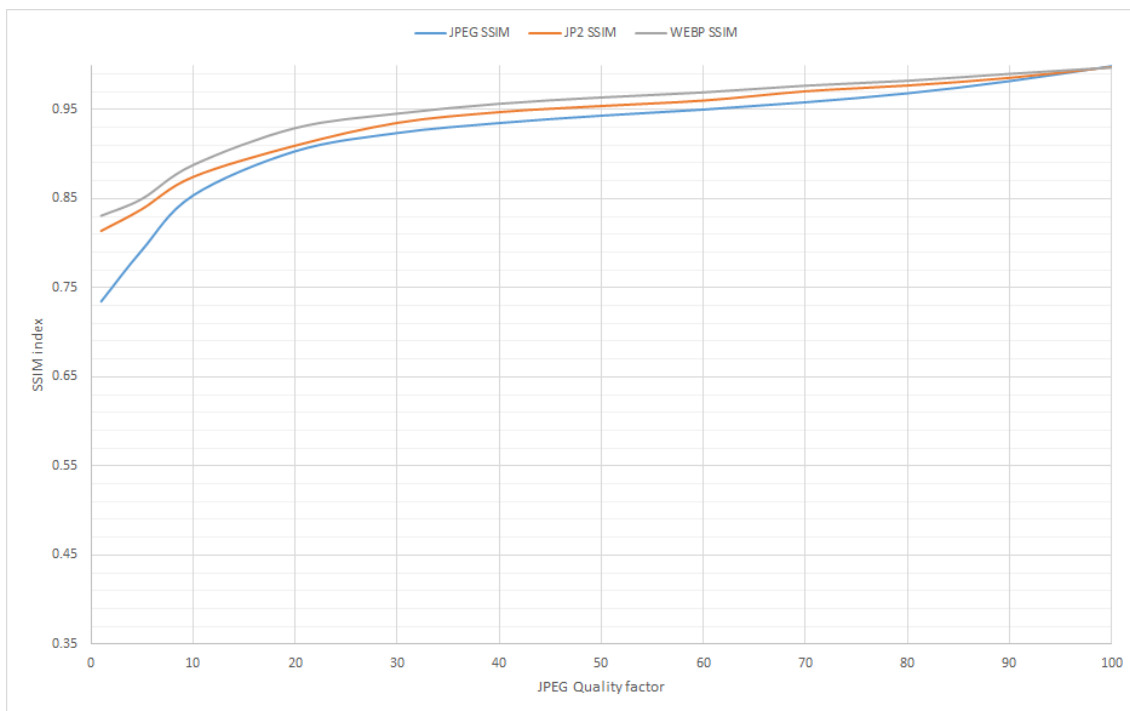
Obr. 6.9: Porovnání formátů měřením SSIM na obrázku Baboon



Obr. 6.10: Porovnání formátů měřením SSIM na obrázku Buildings



Obr. 6.11: Porovnání formátů měření SSIM na obrázku Lena



Obr. 6.12: Porovnání formátů měření SSIM na obrázku Plane

Tab. 6.1: Naměřené hodnoty na obrázku Baboon

Q	JPEG		JPEG2000		WEBP	
	PSNR[db]	SSIM[-]	PSNR[db]	SSIM[-]	PSNR[db]	SSIM[-]
1	19.952	0.397	20.874	0.458	23.018	0.641
5	21.541	0.575	22.100	0.586	23.018	0.641
10	23.444	0.719	24.011	0.709	24.086	0.724
20	25.280	0.815	26.420	0.811	26.539	0.841
30	26.462	0.857	27.721	0.849	28.356	0.888
40	27.393	0.882	28.617	0.864	29.826	0.912
50	28.241	0.900	29.608	0.883	30.796	0.928
60	29.159	0.916	30.464	0.905	31.873	0.942
70	30.481	0.933	31.552	0.921	33.439	0.959
80	32.597	0.953	32.613	0.932	36.226	0.977
90	37.074	0.978	35.160	0.958	41.573	0.993
100	54.803	1.000	40.972	0.989	50.642	0.999

Tab. 6.2: Naměřené hodnoty na obrázku Buildings

Q	JPEG		JPEG2000		WEBP	
	PSNR[db]	SSIM[-]	PSNR[db]	SSIM[-]	PSNR[db]	SSIM[-]
1	20.199	0.590	21.522	0.635	24.542	0.764
5	22.220	0.696	23.345	0.714	24.542	0.764
10	24.393	0.797	25.630	0.796	26.421	0.837
20	26.731	0.869	28.659	0.876	29.377	0.906
30	28.267	0.900	30.690	0.905	31.325	0.931
40	29.355	0.917	32.218	0.930	32.606	0.944
50	30.259	0.929	33.318	0.942	33.720	0.955
60	31.179	0.939	34.408	0.950	34.936	0.964
70	32.481	0.951	35.772	0.960	36.861	0.975
80	34.440	0.965	38.120	0.975	38.984	0.984
90	38.349	0.982	41.774	0.989	44.104	0.995
100	54.374	0.999	55.859	1.000	50.516	0.999

Tab. 6.3: Naměřené hodnoty na obrázku Lena

Q	JPEG		JPEG2000		WEBP	
	PSNR[db]	SSIM[-]	PSNR[db]	SSIM[-]	PSNR[db]	SSIM[-]
1	24.400	0.636	27.841	0.773	29.660	0.817
5	27.412	0.738	29.526	0.821	30.077	0.827
10	30.472	0.829	31.608	0.866	32.588	0.878
20	33.006	0.887	33.627	0.893	34.721	0.909
30	34.320	0.909	34.931	0.911	35.905	0.924
40	35.161	0.921	35.633	0.919	36.779	0.933
50	35.833	0.929	36.283	0.924	37.558	0.941
60	36.478	0.937	36.753	0.928	38.010	0.945
70	37.343	0.945	37.561	0.937	39.152	0.955
80	38.542	0.955	38.527	0.945	40.566	0.967
90	40.796	0.969	40.025	0.957	43.030	0.981
100	54.786	0.998	44.939	0.986	50.667	0.996

Tab. 6.4: Naměřené hodnoty na obrázku Plane

Q	JPEG		JPEG2000		WEBP	
	PSNR[db]	SSIM[-]	PSNR[db]	SSIM[-]	PSNR[db]	SSIM[-]
1	25.332	0.735	28.532	0.814	29.440	0.831
5	27.085	0.792	29.679	0.838	30.442	0.850
10	29.759	0.854	31.588	0.874	32.345	0.888
20	31.896	0.903	33.681	0.909	34.586	0.929
30	33.197	0.924	35.202	0.935	36.044	0.945
40	34.100	0.935	36.304	0.947	37.160	0.956
50	34.851	0.943	37.150	0.954	38.166	0.963
60	35.643	0.950	38.085	0.960	39.211	0.969
70	36.707	0.958	39.551	0.970	40.976	0.977
80	38.346	0.968	41.373	0.977	42.518	0.982
90	41.723	0.982	44.668	0.986	45.560	0.990
100	55.224	0.999	53.375	0.998	52.134	0.997

## 7 ZÁVĚR

V této práci byla objasněna reprezentace obrazu v počítači, vysvětleno co je to barevný model a představeny základní typy. Modely RGB a YCbCr byly blíže popsány, včetně konverze z RGB do YCbCr. Stručně bylo vysvětleno co je to pojem barevná hloubka.

V další kapitole byly popsány současné formáty digitálního obrazu. Mezi tyto formáty patří BMP, GIF, PNG, TGA, TIFF, JPEG, WebP a v neposlední řadě i JPEG2000.

Následující kapitola se zabývala podrobnějším popisem formátu JPEG. Byl objasněn postup převodu obrazu do tohoto formátu, který začíná u transformace z barevného prostoru RGB do YCbCr. Dále pokračuje redukcí barev, diskrétní kosinovou transformací, kvantizací, kódováním DC a AC koeficientů, Huffmanovým kódováním, případně aritmetickým kódováním. Mezi další podrobněji probrané formáty v této kapitole patří JPEG2000, PNG a WebP.

Vysvětleny byly možnosti posouzení kvality komprimovaného obrazu, stručně byly popsány subjektivní metody, které zahrnují DSIS a DSCQS. Objektivním metodám bylo věnováno více prostoru, dozvěděli jsme se výhody a nevýhody metod MSE a PSNR, postup jejich výpočtu. Podrobněji byla rozebrána metoda SSIM, důvod jejího vzniku a postup výpočtu.

Závěrem práce byly vyhodnoceny formáty JPEG, JPEG2000 a WebP metodami PSNR a SSIM. Byla získána představa o efektivitě jednotlivých kompresí, při různých nastaveních komprese. Z výsledků je patrné, že nejlepší poměr kvalita/velikost souboru má formát WebP. Dále při vysokých kompresních poměrech se k výsledkům formátu WebP blíží formát JPEG2000. U formátu JPEG je zejména při vysokých kompresních poměrech patrné značné zkrátení barev a výskyt bloků stejné barvy.

## LITERATURA

- [1] ACHARYA, T., TSAI, PING-SING: *JPEG2000 standard for image compression :concepts, algorithms and VLSI architectures..* Hoboken: John Wiley & Sons, 2005. 274 s. ISBN 0-471-48422-9.
- [2] ČÍKA, P.: *Multimediální služby*. Brno: Vysoké učení technické v Brně, 2012. s. 1-127. ISBN 978-80-214-4443-0.
- [3] GHANBARI, M.: *Standard Codecs: Image Compression to Advanced Video Coding*. Stevenage: The Institution of Electrical Engineers, 2003. 407 s. ISBN 0-85296-710-1.
- [4] ISO/IEC 15948:2003. *Portable Network Graphics (PNG) Specification (Second Edition)*. Oxford Brookes University: W3C, 2003, Dostupné z: <<http://www.w3.org/TR/PNG/>>.
- [5] ITU-R Recommendation BT.500-11, *Methodology for the subjective assessment of the quality of television pictures*, International Telecommunication Union, 2009. Dostupné z: <<http://www.itu.int/md/R07-WP6C-C-0161/en>>.
- [6] NĚMEC, K., ŠILHAVÝ, P.: *Datová komunikace*. Brno: Vysoké učení technické v Brně, 2012. s. 211. ISBN 978-80-214-4455-3.
- [7] RAJMIC, P.: *Základy počítačové sazby a grafiky*. Brno: Vysoké učení technické v Brně, 2012. s. 1-152. ISBN 978-80-214-4451-5.
- [8] SHEIKH, H., R., WANG, Z., CORMACK L., and BOVIK, A.: *LIVE Image Quality Assessment Database*. The University of Texas at Austin, 2003. Dostupné z: <<http://live.ece.utexas.edu/research/quality>>.
- [9] TAUBMAN, David S., MARCELLIN, Michael W.: *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002. 773 s. ISBN 0-7923-7519-X.
- [10] The FreeImage Project. *SourceForge* [online]. 23.3.2014 [cit. 2014-05-20]. Dostupné z: <<http://freeimage.sourceforge.net/>>.
- [11] WANG, Z., BOVIK, A., C., SHEIKH, H., R. and SIMONCELLI, E., P.: *Image quality assessment: From error visibility to structural similarity*. IEEE Transactions on Image Processing, vol. 13, no. 4, Apr. 2004. s. 600–612. Dostupné z: <<http://www.cns.nyu.edu/pub/eero/wang03-reprint.pdf>>.

- [12] WANG, Z., BOVIK, A. C.: *Mean squared error: love it or leave it? - A new look at signal fidelity measures*. IEEE Signal Processing Magazine, vol. 26, no. 1, Jan. 2009. s. 98-117. Dostupné z: <<https://ece.uwaterloo.ca/~z70wang/publications/SPM09.pdf>>.
- [13] WebP. *Google Developers* [online]. 21.3.2014 [cit. 2014-04-28]. Dostupné z: <<https://developers.google.com/speed/webp/>>.

# SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

- .NET aplikační běhové prostředí – Microsoft .NET Framework
- 1D-DWT jednorozměrná diskretní vlnková transformace – 1D Discrete Wavelet Transform
- 2D-DCT dvojrozměrná diskretní kosínova transformace – 2D Discrete Cosine Transform
- 2D-DWT dvojrozměrná diskretní vlnková transformace – 2D Discrete Wavelet Transform
- AC střídavý proud – Alternating Current
- AD průměrná odchylka – Average Difference
- APNG obrazový formát – Animated Portable Network Graphics
- ASCII znaková sada – American Standard Code for Information Interchange
- BMP obrazový formát – Windows Bitmap
- C# objektově orientovaný programovací jazyk
- CCITT mezinárodní telekomunikační unie – Comité Consultatif International Téléphonique et Télégraphique
- CDF biortogonální spline vlnky – Cohen-Daubechies-Feauveau
- CMYK barevný model – Cyan Magenta Yellow Key
- CR řídicí znak pro posun kurzoru na začátek řádku – Cursor/Carrier Return
- CRC cyklický redundantní součet – Cyclic Redundancy Check
- CSV hodnoty oddělené čárkami – Comma-Separated Values
- CUP krok při kódování JPEG2000 – Clean Up Pass
- DC stejnosměrný proud – Direct Current
- DCT diskretní kosinova transformace – Discrete Cosine Transform
- DSCQS testovací metoda – Double Stimulus Continuous Quality Scale
- DSIS testovací metoda – Double Stimulus Impairment Scale

DWT diskrétní vlnková transformace – Discrete Wavelet Transform

EBCOT kódování – Embedded Block Coding with Optimal Truncation

EOB konec bloku – End of Block

GIF obrazový formát – Graphics Interchange Format

HSB barevný model – Hue Saturation Brightness

HSL barevný model – Hue Saturation Lightness

HSV barevný model – Hue Saturation Value

ICT nevratná barevná transformace – Irreversible Color Transform

JFIF obrazový formát – JPEG File Interchange Format

JPEG kompresní standard – Joint Photographic Experts Group

JPEG2000 kompresní standard – Joint Photographic Experts Group 2000

JPEG-LS kompresní standard – Joint Photographic Experts Group Lossless

LCD zobrazovací zařízení – Liquid Crystal Display

LF posun o řádek – Line Feed

LSB nejméně významný bit – Least Significant Bit

LZ77 bezztrátový kompresní algoritmus – Lempel-Ziv 77

LZW bezztrátový kompresní algoritmus – Lempel-Ziv-Welch 84

MD maximální odchylka – Maximum Difference

MNG obrazový formát – Multiple-image Network Graphics

MRP krok při kódování JPEG2000 – Magnitude Refinement Pass

MSB nejvýznamnější bit – Most Significant Bit

MSE střední kvadratická chyba – Mean Squared Error

MOS srovnávací metoda – Mean Opinion Score

MSSIM srovnávací metoda – Mean Structural Similarity

NAE normalizovaná absolutní chyba – Normalized Absolute Error

NC normalizovaná korelace – Normalized Correlation

NMSE normalizovaná střední kvadratická chyba – Normalized Mean Squared Error

PNG obrazový formát – Portable Network Graphics

PSNR špičkový poměr signálu k šumu – Peak Signal-to-Noise Ratio

RCT vratná barevná transformace – Reversible Color Transform

RGB barevný model – Red Green Blue

RIFF souborový formát – Resource Interchange File Format

RLE bezztrátový kompresní algoritmus – Run Length Encoding

ROI oblast zájmu – Region Of Interest

SPP krok při kódování JPEG2000 – Significance Propagation Pass

SSIM srovnávací metoda – Structural Similarity

TGA obrazový formát – Targa

TIFF obrazový formát – Tag Image File Format

VLC kódování s proměnlivou délkou – Variable-Length Code

W3C mezinárodní konsorcium – World Wide Web Consortium

WHT Walsh-Hadamardova transformace – Walsh-Hadamard Transform

YCbCr barevný model používaný u digitální fotografie

YUV barevný model používaný v televizním přenosu

YUY12 formát vzorkování modelu YCbCr

ZIP bezztrátový kompresní algoritmus

## A OBSAH PŘILOŽENÉHO CD/DVD

Přiložené CD/DVD médium obsahuje elektronickou verzi bakalářské práce. Ve složce *Moderní Kompresní Formáty Digitálního obrazu - TeX source* jsou umístěny zdrojové texty elektronického dokumentu. Dále obsahuje ve složce *Source* zdrojové kódy programu pro analýzu kvality obrazu vytvořené v programu Visual Studio 2012. Dále instalační balíček pro aplikační běhové prostředí Microsoft .NET Framework ve verzi 4.0 (pro 32 i 64bitové operační systémy), nutný pro spuštění testovacího programu. Ve složce *Image Database* jsou uloženy referenční i komprimované obrazy použité k testování. Dále obsahuje samotný program pro testování kvality obrazu.

Obsah:

- Image Database – obsahuje 4 referenční obrázky *Baboon*, *Buildings*, *Lena* a *Plane*
  - JPEG2000 – obsahuje obrázky zkomprimované standardem JPEG 2000
  - JPEG – obsahuje obrázky zkomprimované standardem JPEG
  - WEBP – obsahuje obrázky zkomprimované standardem WebP
- Moderní Kompresní Formáty Digitálního obrazu - TeX source
- Source
- *Analyzator\_kvality\_komprese.zip* – testovací aplikace, před spuštěním rozbalte na disk
- *dotNetFx40\_Full\_x86\_x64.exe* – instalační balíček .NET Framework 4.0
- *Moderní Kompresní Formáty Digitálního obrazu.pdf*