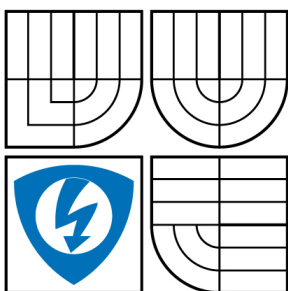


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

KOMPRESNÍ ALGORITMY PŘI ZPRACOVÁNÍ OBRAZU

COMPRESSION ALGORITHMS AND IMAGE PROCESSING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

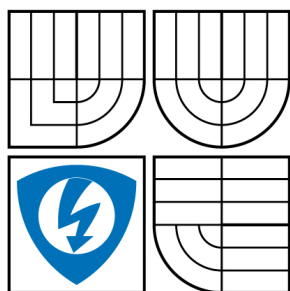
AUTOR PRÁCE
AUTHOR

JOSEF ŠIMEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. FRANTIŠEK KYSELÝ

BRNO 2008



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor

Teleinformatika

Student: Šimek Josef

ID: 78319

Ročník: 3

Akademický rok: 2007/2008

NÁZEV TÉMATU:

Kompresní algoritmy při zpracování obrazu

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte postup zpracování obrazových dat od snímačů, následného zpracování, komprese obrazu a formátu uložení dat. Zejména se zaměřte se na kompresní algoritmy, zjistěte jejich účinnost, vzájemně je porovnejte z hlediska objektivní a subjektivní kvality a vytvořte aplikaci, která umožní tyto parametry hodnotit. Při zpracování se zaměřte na JPEG a JPEG2000. Dále v prostředí Matlab realizujte obrazový kodér JPEG a vytvořený kodér se pokuste inovovat a kodér otestujete.

DOPORUČENÁ LITERATURA:

[1] HLAVÁČ V., Zpracování signálů a obrazů, 1. vyd., Praha Vydavatelství ČVUT 2000, 221 s., ISBN 9788001031100

[2] NOVÁK J., Digitální fotografie a video v praxi, 1. vyd., 159 s., ISBN 8024790718

Termín zadání: 11.2.2008

Termín odevzdání: 4.6.2008

Vedoucí práce: Ing. František Kyselý

prof. Ing. Kamil Vrba, 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 § 152 trestního zákona č. 140/1961 Sb.

LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Josef Šimek
Bytem: Valašská 1658, 75661, Rožnov pod Radhoštěm
Narozen/a (datum a místo): 8.10.1985, Valašské Meziříčí

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 244/53, 602 00, Brno
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Ing. Kamil Vrba, CSc.

(dále jen „nabyvatel“)

Čl. 1 Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- ☐ disertační práce
 - ☐ diplomová práce
 - ☐ bakalářská práce
 - ☐ jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP: Kompresní algoritmy při zpracování obrazu

Vedoucí/ školitel VŠKP: Ing. František Kyselý

Ústav: Ústav telekomunikací

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v * :

- ☐ tištěné formě – počet exemplářů 1
- ☐ elektronické formě – počet exemplářů 1

* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ☐ ihned po uzavření této smlouvy
 - ☐ 1 rok po uzavření této smlouvy
 - ☐ 3 roky po uzavření této smlouvy
 - ☐ 5 let po uzavření této smlouvy
 - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....
Nabyvatel

.....
Autor

ANOTACE

Způsob pořízení a zpracování obrazu má rozhodující vliv na výslednou podobu a velikost dat, které obraz reprezentují. Pro správnou interpretaci obrazu je nezbytné pochopit, jak tato data vznikají.

Bakalářská práce se zabývá problémem získání a úpravy obrazových dat. Jsou rozebrány obecné postupy založené na matematických a fyzikálních principech. Pozornost je pak blíže zaměřena na studium kompresních algoritmů JPEG a JPEG2000, jako účinných nástrojů ke snížení objemu obrazových dat. Účelem práce je vytvořit obrazový kodek JPEG a otestovat jeho účinnost při kompresi dat. Dále pak nalézt a implementovat metodu pro potlačení blokových artefaktů v komprimovaném obraze, které značně znehodnocují jeho kvalitu. K realizaci využijeme prostředí MATLAB, které nabízí širokou podporu pro práci s obrazovými daty.

V závěrečné části provedeme srovnání algoritmů JPEG a JPEG2000 ve vytvořené aplikaci. Zaměříme se především na hodnocení subjektivní a objektivní kvality komprimovaných obrazů. Pro práci s formátem JPEG2000 budou použity programy Kakadu a JPEG2000 Compressor.

Klíčová slova: blokový artefakt, JPEG, JPEG2000, kodek, komprese obrazu, MATLAB, zpracování obrazu

ABSTRACT

The method of image acquisition and processing has a decisive influence on the resulting form and size of the data, which represent the image. For the correct interpretation of the image, it is necessary to understand how the data are generated.

This bachelor's thesis deals with the problem of obtaining and modification of image data. The general procedures based on the mathematical and physical principles are analyzed. Attention is focused on closer study of JPEG and JPEG2000 compression algorithms, as effective tools for reducing the volume of image data. The purpose of this thesis is to create a JPEG video codec and to test its effectiveness in data compression. Furthermore, to find and implement a method for the suppression of blocking artifacts in compressed image, which greatly reduce its quality. The implementation will use MATLAB environment, which offers wide support for working with image data.

In the final part we will compare the algorithms of JPEG and JPEG2000 in the created application. We will focus on the subjective and objective evaluation of the quality of compressed images. To work with JPEG2000 format the programs Kakadu and JPEG2000 Compressor will be used.

Keywords: blocking artifact, JPEG, JPEG2000, codec, image compression, MATLAB, image processing

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „KOMPRESNÍ ALGORITMY PŘI ZPRACOVÁNÍ 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 jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....
(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu své bakalářské práce Ing. Františku Kyselému za velmi užitečnou metodickou pomoc a cenné rady při zpracování bakalářské práce.

V Brně dne

.....
(podpis autora)

PŘEHLED POUŽITÝCH ZKRATEK

2D FDWT – dvojrozměrná dopředná diskretní waveletová transformace
2D FDCT – dvojrozměrná dopředná diskretní kosinová transformace
2D IDCT – dvojrozměrná zpětná diskretní kosinová transformace
2D IDWT – dvojrozměrná zpětná diskretní waveletová transformace
AC – střídavá složka
AGC – Automatic Gain Control, automatické řízení zisku (zesílení)
AWB – Automatic White Balance, automatické vyvážení bílé
bpp – bits per pixel, počet bitů připadající na jeden pixel
BMP – Microsoft Windows Bitmap, nekomprimovaný formát obrazových dat
CCD – Charge-coupled device, technologie snímače obrazu
CMOS – Complementary metal-oxide-semiconductor, technologie snímače obrazu
CMYK – Cyan Agenta Yellow black, subtraktivní barevný model
CR – compress ratio, kompresní poměr
DSLR – Digital single-lens reflex, digitální zrcadlovka
DC – stejnosměrná složka
DCT – Discrete Cosine Transform, diskretní kosinová transformace
DFT – Discrete Fourier Transform, diskretní Fourierova transformace
DWT – Discrete Wavelet Transform, diskretní waveletová transformace
EBCOT – Embedded Block Coding with Optimized Truncation, entropické kódování
EOB – End of Block, konec kódovaného bloku
EZW – Embedded Zerotree Wavelet, kódovací algoritmus
FIT – Field Interline Transfer, druh plošného snímače CCD
fk – faktor kvality
GIF – Graphics Interchange Format, formát bezeztrátové komprese obrazových dat
ITU – International Telecommunication Union, Mezinárodní telekomunikační unie
JPEG – Joint Photographic Experts Group, obrazový formát (standard) ztrátové komprese
JPEG2000 – Joint Photographic Experts Group, obraz. formát (standard) ztrátové komprese
LCD – Liquid-Crystal Display, obrazovka z tekutých krystalů
MATLAB – MATrix LABoratory, skriptovací programovací jazyk
MPEG – Motion Picture Experts Group, formát komprese videa
MSE – Mean Squared Error, střední kvadratická odchylka
PNG – Portable Network Graphics, formát bezeztrátové komprese obrazových dat
PSNR – Peak Signal to Noise Ratio, špičkový odstup signál/šum
px – pixel, obrazový bod
RGB – Red Green Blue, aditivní barevný model
RLE – Run Length Encoding, způsob bezeztrátového kódování
SPIHT – Set Partitioning In Hierarchical Trees, metoda prahování koeficientů
YCbCr – barevný model s oddělenou jasovou a barevnými složkami
ZRL – Zero Run Length, speciální kódová značka

OBSAH

ÚVOD	12
1 POŘÍZENÍ OBRAZU	13
1.1 Obraz jako obrazová funkce	13
1.2 Barevný obraz	13
1.3 Fyzikální podstata snímání obrazu	13
1.3.1 Rovnice ozáření	14
1.3.2 Zobrazení geometrickou optikou	14
1.4 Snímače obrazu používané v kamerách	16
1.4.1 Snímače CMOS	16
1.4.2 Snímače CCD	17
1.5 Digitální kamery	17
1.5.1 Snímání barevného obrazu	18
1.5.2 Obvody zpracování signálu v digitálních kamerách	18
2 ÚPRAVA DIGITÁLNÍCH OBRAZOVÝCH DAT	20
2.1 Digitalizace	20
2.2 Korekce geometrického zkreslení	20
2.3 Korekce hodnot jasu	21
2.3.1 Jasové korekce	21
2.3.2 Transformace jasové stupnice	21
3 KOMPRESSE OBRAZOVÝCH DIGITÁLNÍCH DAT	23
3.1 Nekomprimovaný obraz	23
3.2 Bezeztrátová komprese	23
3.3 Ztrátová komprese	23
4 KOMPRESNÍ ALGORITMY	24
4.1 Algoritmus JPEG	24
4.1.1 Úprava vstupních dat	24
4.1.2 Diskrétní kosinová transformace	25
4.1.3 Kvantování koeficientů	26
4.1.4 Kódování dat	27
4.1.5 Uložení komprimovaných dat	27
4.2 Algoritmus JPEG2000	28
4.2.1 Úprava vstupních dat	28
4.2.2 Diskrétní vlnková transformace	28
4.2.3 Kvantování a kódování koeficientů dekompozičního obrazce	30
4.2.4 Uložení komprimovaných dat	31
5 KOMPRESNÍ ARTEFAKTY	32
5.1 Blokové artefakty	32
5.2 Redukce blokových artefaktů	32
5.2.1 Metoda prostorové filtrace	32
6 HODNOCENÍ EFEKTIVITY KOMPRESSE	34
6.1 Kompresní poměr	34
6.2 Kvalita obrázku	34

6.2.1	Subjektivní hodnocení obrázku	34
6.2.2	Objektivní hodnocení obrázku	34
6.3	Výpočetní náročnost kompresního algoritmu	35
7	REALIZACE KODEKU JPEG V PROSTŘEDÍ MATLAB.....	36
7.1	JPEG KoDeK	36
7.1.1	Algoritmus JPEG	37
7.1.2	Potlačení blokových artefaktů	41
7.1.3	Program Kakadu	42
8	TESTOVÁNÍ A HODNOCENÍ KOMPRESNÍCH ALGORITMŮ	43
8.1	Test obrazového kodeku JPEG	43
8.1.1	PSNR a CR.....	44
8.1.2	Vliv redukce blokových artefaktů	45
8.2	Srovnání kompresních algoritmů JPEG a JPEG2000	47
8.2.1	Objektivní hodnocení kvality obrázků	48
8.2.2	Subjektivní hodnocení kvality obrázků.....	49
8.2.3	Shrnutí.....	50
ZÁVĚR.....		51
SEZNAM POUŽITÉ LITERATURY		52

SEZNAM OBRÁZKŮ

Obr. 1.1: Spojná tenká čočka.....	15
Obr. 1.2: Blokové schéma digitální CCD kamery.....	20
Obr. 2.1: Histogram šedo-tónového obrazu pro 256 jasových úrovní.....	22
Obr. 4.1: Schéma komprese dat u algoritmu JPEG	24
Obr. 4.2: Blokové schéma komprese dat u algoritmu JPEG2000	28
Obr. 4.3: Modulové kmitočtové charakt. filtrů pro dyadickou dekompozici úrovně 3	29
Obr. 4.4: Vlnková transformace obrazu pro 1. úroveň	29
Obr. 4.5: Dekompoziční obrazec pro hloubku dekompozice $d = 3$	30
Obr. 5.1: Gaussovo dvojrozměrné rozdělení pravděpodobnosti	33
Obr. 7.1: Grafické rozhraní programu JPEG KoDeK.....	36
Obr. 7.2: Použité kvantizační tabulky.....	38
Obr. 7.3: Použité Huffmanovy tabulky pro kódování DC koeficientů	39
Obr. 7.4: Ukázka použitých Huffmanových tabulek pro kódování AC koeficientů	40
Obr. 7.5: Struktura dat pro uložení komprimovaného obrázku	40
Obr. 7.6: Princip redukce blokových artefaktů.....	42
Obr. 8.1: Původní (nahore) a komprimované obrázky, $fk = 50$	43
Obr. 8.2: Graf závislosti PSNR na faktoru kvality	44
Obr. 8.3: Graf závislosti PSNR na faktoru kvality pro různý typ použitého filtru.....	45
Obr. 8.4: Subjektivní posouzení kvality obrázku při redukci blokových artefaktů ($fk = 10$)..	46
Obr. 8.5: Testované obrázky	47
Obr. 8.6: Závislost PSNR na kompresním poměru CR (<i>Lena.bmp</i> , <i>Birds.bmp</i>).....	48
Obr. 8.7: Závislost PSNR na kompresním poměru CR (<i>Midnight.bmp</i> , <i>Texture.bmp</i>).....	49
Obr. 8.8: Porovnání subjektivní kvality obrázků (<i>Midnight.bmp</i> – nahore, <i>Texture.bmp</i>).....	50

SEZNAM TABULEK

Tab. 8.1: Naměřené hodnoty závislosti PSNR a CR na faktoru kvality	44
Tab. 8.2: Naměřené hodnoty PSNR pro různý typ použitého filtru	45
Tab. 8.3: Naměřené hodnoty PSNR pro měnící se kompresní poměr CR	48

ÚVOD

V dnešní době, kdy digitalizace proniká do různých oblastí lidské činnosti, stávají se obrazová data v digitální formě nedílnou součástí našeho života. S nástupem digitálního televizního vysílání je průběh tohoto procesu ještě umocněn. Digitalizace v užším slova smyslu značí převod analogového (spojitého) signálu na posloupnost číselných vzorků (nejčastěji binární čísla). Obecně takto získaná data jsou spolehlivějším zdrojem informace o daném signálu. Výhodou je rovněž nesrovnatelně větší životnost dat ve srovnání s analogovým médiem.

Z počátku se v digitální podobě vyskytovala pouze ta obrazová data, která byla vytvářena na počítačích. S postupným vývojem procesorů a grafických čipů a se zvyšováním kapacity přenosových médií se mohly začít digitalizovat i analogové signály nesoucí informace o skutečném obrazu. Zcela nový rozměr ve využití digitálních obrazových dat nastartoval velmi prudký rozvoj internetu a mobilních komunikací. Obraz je možno pořídit a ještě v téže okamžiku přenést teoreticky mezi libovolnými dvěma místy. S masivním užitím, ať už statických (obrázky, fotky) nebo dynamických (video) digitálních obrazových dat však vyvstává jeden značný problém: velikost přenášených dat. Množství informace o daném obrazu, které (zjednodušeně) přenášíme jako posloupnost 0 a 1, je přímo úměrné velikosti detailů v obraze. Jinak řečeno, vyžadujeme-li kvalitní snímek, který co nejlépe odpovídá skutečnosti, musíme počítat s velkým množstvím dat, která budou tento snímek reprezentovat. Náročnost na kapacitu přenosových médií může být tedy značná. Proto musel přijít na řadu nějaký mechanismus, který tento problém vyřeší. Tím mechanismem se stala komprese obrazové informace.

Komprese (nebo také komprimace) je postup, jehož úkolem je zmenšit datový tok při ukládání informace o obrazu. Obecně lze komprese rozdělit do dvou základních kategorií na bezztrátové a ztrátové. U bezztrátové komprese získáme po rekonstrukci data v původní podobě. Ztrátová komprese má za následek snížení nároků na přenosovou kapacitu na úkor nenávratné ztráty některé informace výsledného obrazu.

Cílem této bakalářské práce je objasnit postup při zpracování obrazových dat. Způsob získání obrazu pomocí obrazového snímače, převod obrazové funkce na obrazová digitální data a následné zpracování obrazových dat s využitím různých druhů transformací. Zejména se zaměříme na kompresi obrazových dat pomocí algoritmů JPEG a JPEG2000 a porovnáme jejich účinnost. Za tímto účelem bude vytvořena praktická aplikace v prostředí MATLAB, v rámci níž bude implementován obrazový kodek JPEG. Kodek se dále pokusíme rozšířit o možnost potlačení kompresních artefaktů, které se v obraze objevují při vysokých stupních komprese. Ve vytvořeném programu poté provedeme srovnání obrázků komprimovaných algoritmem JPEG2000 a námi realizovaným kodekem JPEG.

1 POŘÍZENÍ OBRAZU

1.1 Obraz jako obrazová funkce

Svět, ve kterém žijeme, je trojrozměrný, tedy i objekty, které jsou jeho součástí, vnímá naše oko jako vícerozměrný signál – obraz. Pokud bychom použili matematického modelu, může být obraz popsán pomocí spojitě skalární funkce f dvou nebo tří proměnných, kterou nazýváme *obrazová funkce*. V nejjednodušším případě je statický obraz vyjádřen jako obrazová funkce dvou proměnných $f(x, y)$, které udávají souřadnice bodu v rovině (např. na kulové ploše sítnice lidského oka). Jedná-li se o dynamické obrazy $f(x, y, t)$, tj. plošné obrazy měnící se v čase t , nebo o objemové obrazy $f(x, y, z)$, použijeme obrazovou funkci tří proměnných [9]. Hodnoty, které obrazová funkce nabývá, odpovídají některé měřené fyzikální veličině (např. jas, teplota atd.). Dvojměrný obraz v televizní kameře nebo na sítnici oka je výsledkem perspektivního zobrazení, nebo-li středového promítání, objektu z 3D scény. Toto zobrazení bude popsáno dále, v kapitole o optické části pořízení obrazu.

1.2 Barevný obraz

Máme-li v jednoduchém případě obraz monochromatický (pouze s jedinou barvou), je reprezentován jedinou obrazovou funkcí $f(x, y)$. V reálných situacích však většinou pracujeme se složitějším případem, tj. s barevným (multispektrálním) obrazem. Pro každou dvojici plošných souřadnic (x, y) , tak získáváme vektor hodnot, např. jasů jednotlivých barevných složek. Oblast vlnění se snažíme rozdělit do spektrálních pásem, jejichž vlnové délky nás budou při zpracování zajímat. Poté snímáme obraz několika senzory, z nichž každý má různou citlivost na část spektrálního pásma. Pro viditelné záření uvažujeme vlnové délky z rozsahu 400–700 nm. Výsledný barevný vjem je dán dominantní vlnovou délkou λ , spektrálním složením barvy $S(\lambda)$ a jasnem barvy L_f [20].

Při vytváření nových barev pomocí dvou nebo vícesložkových barev se využívá nedokonalosti lidského zraku. Na tomto principu jsou založeny různé barevné modely, které se používají především pro zjednodušení záznamu barevné informace. Např. kolorimetrická soustava RGB (angl. *Red, Green, Blue*) skládá barevný signál ze tří spektrálních složek (vlnová délka červené je 700 nm, zelené 546,1 nm a modré 435,8 nm). Každá ze složek je zpracovávána samostatně, výsledný barevný obraz vzniká smícháním složek. Jedná se tedy o tzv. aditivní model. V tiskových zařízeních se užívá subtraktivní model CMYK (angl. *Cyan, Magenta, Yellow, black*), který je naopak založen na odčítání jednotlivých barevných složek. Pro kompresi obrazu se využívají modely s oddělenou jasovou (Y) a barevnými složkami (U,V), jako např. YUV nebo YCbCr (viz lit [5], [16], [19]).

Při digitálním zpracováním multispektrálních obrazů se aplikují obdobné postupy jako při práci s obrazy monochromatickými, metody se použijí na jednotlivé spektrální složky.

1.3 Fyzikální podstata snímání obrazu

Při styku elektromagnetického záření s hmotou dochází k vzájemné reakci. Ta probíhá buď uvnitř objektů nebo pouze na jejich povrchu. Záření je z objektů uvolňováno vlivem tepelného pohybu molekul. Ten může být vyvolán vlastními molekulami nebo jako reakce na některou z vnějších stimulací (např. u odraženého záření). Záření je spojeno s přenosem informace, kterou můžeme využít pro bližší rozpoznání objektů a určení jejich některých

vlastností. Hlavními nositeli jsou zejména vlnová délka záření, což je jinak vyjádřena frekvence, a amplituda, která udává intenzitu záření.

Většina obrazových senzorů měří množství světelné energie dopadající na plochu senzoru. Takto získáme informaci o intenzitě světla, které častěji říkáme jas. Způsob vzniku obrazu nám umožňuje vysvětlit *radiometrie*. Je to část optiky, která se zabývá měřením energie přenášené elektromagnetickým zářením. Velmi blízkým vědním oborem k radiometrii je *fotometrie*. Ta zkoumá světlo z hlediska jeho působení na zrakový orgán člověka. Fotometrie popisuje podobné jevy jako radiometrie, avšak výsledné hodnoty veličin závisí na individuálních schopnostech pozorovatele [9].

1.3.1 Rovnice ozáření

Důležitým vztahem při popisu vzniku obrazu z elektromagnetického záření je tzv. *rovnice ozáření*, která dává do vztahu ozáření E dopadající na senzor v obrazové rovině v závislosti na záři L v pozorované scéně. Její odvození lze nalézt v lit.[9], výsledný tvar rovnice je

$$E = L \frac{\pi}{4} \left(\frac{d}{f} \right)^2 \cos^4 \alpha, \quad (1.1)$$

kde d představuje průměr čočky objektivu, f je ohnisková vzdálenost a α je úhel, pod kterým dopadá záření na senzor v obrazové rovině.

V rovnici se objevuje činitel d / f , jeho obrácená hodnota $N_f = f / d$ je důležitým parametrem objektivu a nazývá se *clonové číslo objektivu* nebo také *f-číslo*. Clona je přepážka s otvorem uprostřed, která je součástí objektivu. Velikost otvoru v cloně můžeme měnit a tím regulovat množství dopadajícího světla.

Další člen v rovnici, $\cos^4 \alpha$, ovlivňuje optickou vadu zvanou *přirozená vinětace*. Ta se projevuje tím, že střed obrázku je jasnější než jeho okraje. Paprsky které se v obrazové rovině lámou s větším úhlem α (dále od optické osy) jsou více zeslabovány. Jiným druhem je *vinětace optická*, která je způsobena nenulovou tloušťkou čoček objektivu. Velikost clonového otvoru je větší pro paprsky dopadající zešikma než pro paprsky dopadající ve směru optické osy. Přirozená i optická vinětace jsou vlastností všech objektivu.

1.3.2 Zobrazení geometrickou optikou

Zařízení pro snímání obrazu (digitální fotoaparát, videokamera) se skládá ze tří hlavních částí a to z objektivu, snímače obrazu a elektroniky, která slouží k převodu sejmutého obrazu pro další zpracování. K popisu funkce objektivu použijeme idealizovaný model dírkové komory, který všechny body snímaného objektu ležící v trojrozměrném prostoru zobrazí středovým promítáním (perspektivním zobrazením) do dvojrozměrného obrazu. Pro jednoduchost nebudeme uvažovat vlnový charakter světla zahrnující jevy jako polarizace, interference a difrakce světla [6].

Nejjednodušší objektiv je představován ideální tenkou čočkou, u které se paprsky promítají stejně jako u dírkové komory. Umístění obrazu libovolného bodu, který neleží na ose, můžeme nalézt pomocí tří speciálních paprsků vybraných ze všech paprsků, které vytvářejí obraz.

- Paprsek původně rovnoběžný s osou čočky se po průchodu čočkou láme do ohniska.
- Paprsek, který původně procházel ohniskem, vystupuje z čočky rovnoběžně s osou.
- Paprsek procházející středem čočky (hlavním bodem) se neodklání.

U správně zaostřené čočky se tyto paprsky protínají v jediném bodě a tvoří obraz bodu z předmětové roviny, viz **obr. 1.1**. Nevýhodou ideální tenké čočky je fakt, že soustřeďuje pouze paprsky vycházející z roviny kolmé k optické ose, které se nacházejí před předmětovým ohniskem.

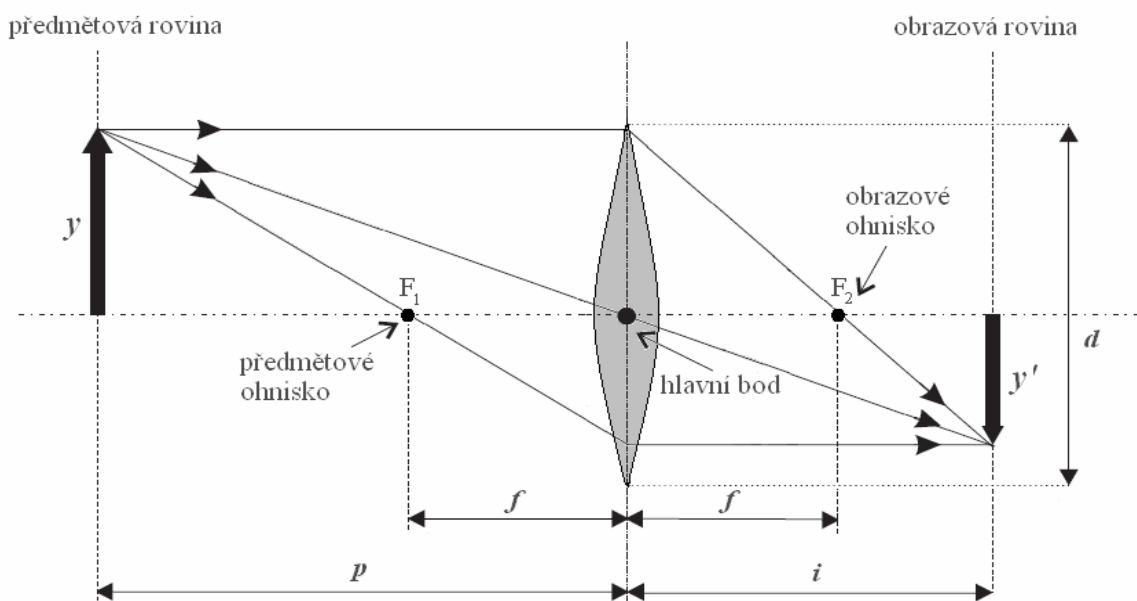
Vztah mezi vzdáleností předmětu, obrazu a ohniskovou vzdáleností vyjadřuje *zobrazovací rovnice čočky* (odvozena z podobnosti trojúhelníků [6])

$$\frac{1}{f} = \frac{1}{p} + \frac{1}{i}, \quad (1.2)$$

kde f je ohnisková vzdálenost (vzdálenost mezi hlavním bodem a ohniskem), p je vzdálenost předmětu (předmětové roviny) a i je vzdálenost obrazu (obrazové roviny). Při zobrazování čočkami mohou vznikat zvětšené nebo zmenšené obrazy předmětů. Pro jejich matematický popis používáme veličinu zvětšení optické soustavy m . Je definována jako podíl velikosti obrazu y' k velikosti předmětu y (odvozené vztahy vyplývají z podobnosti trojúhelníků [6])

$$m = \frac{y'}{y} = -\frac{i}{p} = -\frac{i-f}{f} = -\frac{f}{p-f}, \quad (1.3)$$

záporné znaménko udává, že vzniká převrácený obraz.



Obr. 1.1: Spojná tenká čočka

Není-li objektiv ideálně zaostřen, tzn. je zaostřen mimo obrazovou rovinu senzoru, bude se obraz bodu promítat jako malý kruh o průměru k a nikoliv jako bod. S tímto jevem jsou v optice spojeny dva pojmy – hloubka ostrosti a hloubka pole.

Hloubka ostrosti a hloubka pole

Hloubka ostrosti (angl. *depth of focus*) udává interval hodnot, kde se může nacházet obrazová rovina, aby rozostření (průměr kruhu rozostření) bylo menší než předem stanovená hodnota. Je to subjektivní parametr a vztahuje se vždy k celkovému rozlišení obrazu. Větší rozlišení znamená menší přípustné rozostření.

Ekvivalentem k hloubce ostrosti je v předmětovém prostoru hloubka pole (angl. *depth of field*). Pouze přesně zaostřený bod objektu se znovu zobrazí jako bod, hloubka pole definuje velikost vzdálenosti před a za tímto bodem, kdy je průměr kruhu rozostření ještě dostatečně malý a objekt se jeví jako ostrý. Průměr kruhu rozostření se s vzdáleností od zaostřeného bodu zvyšuje postupně, takže hranice pro „ještě“ ostrý objekt nejsou striktně dané. U digitálních obrazů je povolené rozostření přímo úměrné velikosti pixelu. Jak hloubka ostrosti, tak hloubka pole se zvětšují při snižování velikosti clonové apertury (blíže lit. [9], [19]).

Druhy objektivů

V praxi jsou objektivy složeny ze soustavy čoček a mohou tak měnit svou ohniskovou vzdálenost. Jedna hlavní rovina je nahrazena dvěma hlavními rovinami a máme tedy i dva hlavní body. Clona zabudovaná v objektivu dovoluje regulovat množství světla procházející objektivem. Podle velikosti úhlu záběru, ve kterém je možno snímek pořídit, lze objektivy rozdělit do tří skupin.

Normální objektiv má úhel záběru asi 50°, což je přibližně úhel vnímání lidského oka. *Širokoúhlý objektiv* má kratší ohniskovou vzdálenost a tím pádem i širší záběr, který může dosahovat až 180°. Naproti tomu *teleobjektivy* mají zorný úhel užší a umožňují tak snímat i poměrně vzdálené předměty.

Speciálním druhem objektivů jsou objektivy *telecentrické* a *hypercentrické*. Telecentrický objektiv umožňuje zobrazit objekty ve skutečné velikosti. V obrazovém ohnisku má umístěnu aperturní clonu, čímž se dosáhne toho, že obraz tvoří pouze paprsky přibližně rovnoběžné s optickou osou. Průměr objektivu však musí být minimálně tak velký jako snímáný objekt. U hypercentrického objektivu se aperturní otvor nachází mezi obrazovým ohniskem a rovinou, což způsobuje, že vzdálenější objekty jsou zobrazovány jako větší [9].

1.4 Snímače obrazu používané v kamerách

K tomu aby dopadající energie světla mohla být zachycena a dále převedena na elektrický náboj, slouží v kamerách snímače světelné energie. S prudkým rozvojem vývoje polovodičů se rozšířily zejména snímače, které pracují na principu vnitřního fotoelektrického jevu [7].

V současné době na trhu kralují monolitické snímače CCD (angl. *Charge Coupled Devices*), jejichž hlavním konkurentem jsou snímače vyráběné technologií CMOS (angl. *Complementary Metal Oxid Semiconductor*).

1.4.1 Snímače CMOS

Tato technologie je masová a výrobně jednodušší, a tedy i poměrně levná. Navíc je možno na jednom čipu naintegrovat snímač zároveň s procesorem zpracovávajícím obraz. Jednou z hlavních předností CMOS snímačů je nižší spotřeba energie, která je dána odlišnou konstrukcí a schopností pracovat pouze s jednou napěťovou úrovní. Výhodou je rovněž velká rychlost přenosu akumulovaného náboje ze snímače na převodník náboj / napětí a větší rozsah

intenzit (jasu) světla, které snímač dokáže pojmout. Nevýhodou oproti CCD kamerám je poměrně velký šum [9]. Díky těmto vlastnostem získaly CMOS snímače dominantní postavení při použití ve fotoaparátech mobilních telefonů, kde není kladen takový důraz na kvalitu záznamu a jejichž produkce roste závratným tempem. V současné době se však rozšířily i do jiných oblastí, kde je naopak kvalita vyžadována, důkazem jsou DSLR (*Digital single-lens reflex* – digitální zrcadlovky).

1.4.2 Snímače CCD

Jak už samotný název napovídá, jedná se o zařízení s vázanými náboji. Pro převod světelné energie na elektrický náboj jsou v jednotlivých čidlech použity Schottkyho fotodiody. Snímač je tvořen analogovým posuvným registrem, který umožňuje postupně přesouvat akumulovaný elektrický náboj. Jeho nespornými výhodami jsou vysoká geometrická přesnost obrazu, lineární převodní charakteristika, malé rozměry čipu, velký dynamický rozsah i široký spektrální rozsah, velká citlivost a poměrně nízký šum [4]. Problémem CCD snímačů je vzájemné ovlivňování nábojů v sousedních pixelech (angl. *blooming*). Z důvodu méně rozšířené a náročnější výroby je cena CCD kamer vyšší.

Plošné snímače CCD

V závislosti na tom, jaký druh obrazu potřebujeme snímat, využíváme odlišné CCD struktury. Řádkové (lineární) CCD snímače se používají ke snímání jednorozměrného obrazu nebo případně dvourozměrného obrazu pomocí posuvu po řádcích, příkladem takového zřízení je scanner nebo fax. K tomu abychom mohli snímat dvojrozměrný obraz najednou, což je nezbytné pro všechny kamery, musíme použít plošný CCD snímač.

Co se týká konstrukce, jsou plošné snímače vlastně spojením mnoha lineárních CCD snímačů na jediném čipu. Vyrábějí se ve čtvercovém (obsahují až 4000 x 4000 bodů) nebo obdélníkovém provedení. Existuje řada různých konstrukcí plošných CCD snímačů, které mají za úkol zlepšit kvalitu snímání. My si zde uvedeme snímač typu FIT (*Field Interline Transfer*), který je kombinací dvou jiných typů [8].

Snímač se skládá se dvou shodných částí – snímací část a paměťová část. Ve snímací části dochází k přeměně energie světla na elektrický náboj s využitím velkého počtu světlocitlivých buněk. Náboj z buněk se přesune do neprůsvitných vertikálních registrů. Z registrů je náboj dále vysunut ve svislém směru do zatemněné paměťové části. Po přesunu náboje začíná ve snímací části nový akumulací proces, zatímco z paměťové části se náboje po řádcích přesunují do výstupního horizontálního registru. Z toho se pak celý řádek posouvá na vstup převodníku Q/U, na jehož výstupu se po patřičném zesílení odebírá výstupní obrazový signál. Nové technologie využívají pro zvýšení citlivosti snímače mikroskopické čočky, které se umístí nad jednotlivé světlocitlivé buňky a soustředí dopadající světlo na aktivní plochu snímače [8].

1.5 Digitální kamery

Digitální kamerou budeme v tomto popisu chápat obecně jakékoliv zařízení pro snímání obrazu, které obsahuje objektiv, snímač a potřebnou elektroniku pro zpracování signálu (tedy např. digitální fotoaparát, digitální videokamera atp.).

1.5.1 Snímání barevného obrazu

Až dosud jsme popisovali CCD snímače pro získání černobílého obrazu. K snímání barevného obrazu se v kamerách prakticky používají dvě metody rozlišené podle počtu využívaných plošných snímačů [8].

Kamery se třemi snímači CCD

Pro profesionální účely se je určen blok CCD složený ze tří samostatných snímačů, které díky nanesenému barevnému filtru a soustavě dvou dichroických zrcadel (odráží světlo pouze určité vlnové délky) snímají vždy pouze jednu ze tří základních barev spektra (R, G, B). Získáme tak tři samostatné signály, které společně vytváří barevný obraz. Kvůli požadavku vysokého rozlišení má každý snímač velký počet obrazových bodů. Nevýhodou tohoto uspořádání je větší objem a vyšší cena kamery.

Kamery s jedním snímačem CCD

V digitálních fotoaparátech a v menších amatérských kamerách se využívá pouze jediného plošného CCD snímače s napařenými barevnými filtry a se třemi horizontálními řádkovými registry, které slouží k odvedení náboje. Filtry mohou mít různou strukturu uspořádání (např. svislé proužky R, G, B), ale nejčastěji používána je tzv. Bayerova maska [4]. Je tvořena čtverci, které se skládají ze čtyř pixelů (obrazových bodů). Lidské oko je nejcitlivější na zelenou barvu, proto je informace o této barvě nejdůležitější. Z tohoto důvodu obsahují čtverce barvy v zastoupení červená, modrá a dvě zelené. Výsledný barevný obrazový bod se vytváří interpolací čtveřice sousedních pixelů.

1.5.2 Obvody zpracování signálu v digitálních kamerách

Vlastní kamera, jak jsme již dříve uvedli, je tvořena objektivem, snímačem a obvody pro další zpracování signálu. V současné době se používají digitální kamery, jenž mají oproti analogovým řadu výhod. U signálu dochází k jedinému převzorkování a zpracování signálu se může provádět digitálně již v kameře nebo dále pak v počítači. Při přenosu signálu nevzniká žádný šum, který by rušil kvalitu signálu. Rovněž nedochází k posunu řádku, což je další nevýhoda analogových kamer [9].

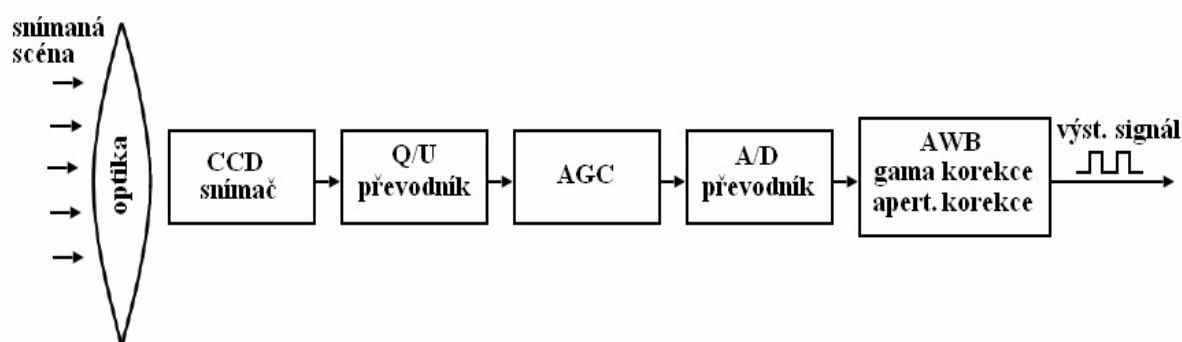
Blokové schéma digitální kamery je uvedeno na **obr. 1.2**. Po převodu náboje na napětí následuje blok AGC (*Automatic Gain Control*), který automaticky mění zesílení signálu v závislosti na množství světla ve snímané scéně. Má zabránit tomu, aby se pro příliš osvětlená místa nedostal výstup kamery do saturace. Poté je signál v A/D převodníku digitalizován (viz **kap. 2.1**). Následují obvody, které slouží k úpravě a korekci signálu, aby výsledný obraz byl přizpůsoben lidskému vnímání.

Gama korekce umožňuje nelineárně transformovat jasovou stupnici a kompenzuje tak charakteristiku vakuové obrazovky, která je exponenciální. U plochých LCD (*Liquid Crystal Display*) monitorů je závislost jasu na vstupním napětí lineární, a tudíž se kompenzace nevyužívá.

Blok AWB (*Automatic White Balance*) slouží k automatickému vyvážení bílé barvy, které se provádí změnou poměru složky R a B vůči složce G. Při snímání objektu mohou nastat různé světelné podmínky v závislosti na druhu zdroje světla. Některé z nich mohou mít negativní vliv na výslednou podobu obrazových dat. Lidské oko se těmto měnícím

podmínkám dokáže relativně dobře přizpůsobit a nevnímá je, avšak světlocitlivé snímače v kamerách tuto vlastnost nemají a zachytí obraz ve skutečných (mnohdy nepřírodných) barvách. Ve výsledném obraze se to projeví nepřírodnou barvou objektů, na které dopadalo jiné než „bílé světlo“ (obsahuje spektrum všech barev). AWB se snaží o to, aby objekty byly pořád stejně barevné, i když jsou osvětleny světlem různé barvy. V obraze se hledají místa s největším jasnem, která poté slouží jako reference bílé. Rovněž se zkoumá celkový barevný nádech obrazu. Podle zjištěných výsledků se poté na pořízených obrazových datech provádí barevná korekce [19].

Aperturová korekce odstraňuje případné nedostatky optické soustavy v objektivu. Kamery mohou obsahovat i další obvody, které korigují různé parametry obrazového signálu. Takto upravený signál můžeme dále zpracovávat v počítači.



Obr. 1.2: Blokové schéma digitální CCD kamery

2 ÚPRAVA DIGITÁLNÍCH OBRAZOVÝCH DAT

2.1 Digitalizace

Až dosud jsme stále pracovali s analogovým signálem. Pro úpravu a zpracování obrazových dat v počítači je nutno obrazovou funkci $f(x, y)$ nejprve digitalizovat, tedy převést na digitální signál. Tento převod se uskutečňuje v počítači nebo v současnosti již převážně v samotné kameře (digitální fotoaparát, digitální videokamera), viz **obr. 1.2** výše. Samotná digitalizace je analogií k digitalizaci jednorozměrných signálů. Proces se skládá ze dvou hlavních částí – vzorkování a kvantování (lit. [9], [19]).

Vzorkování

Dvojměrný obraz vzorkujeme v matici $\mathbf{J} \times \mathbf{K}$ bodů a spojitou úroveň jasu každého vzorku kvantujeme do N intervalů. Takto získáme jasovou funkci obrazu složenou pouze z celočíselných hodnot. Rovněž platí, že pro jemnější vzorkování (větší velikost matice) a kvantování dostáváme lépe aproximovaný původní spojitý obrazový signál.

Při určování intervalu vzorkování (plošné vzorkovací frekvence) musíme respektovat Shannon-Kotělnikovův teorém. Vzorkovací frekvence musí být alespoň dvakrát větší než nejvyšší frekvence ve vzorkovaném signálu. To znamená, že interval vzorkování musíme volit menší nebo rovný polovině rozměru největších detailů v obraze, protože jinak dochází k aliasingu [19] – vlivem překrývání spekter se ve vzorkovaném signálu objeví kmitočty, které v původním signálu nebyly, což má za následek změny ve výsledném obraze. Proto se aliasingu snažíme vyvarovat, kvůli zachování detailů se při zpracování obrazů vzorkuje mnohokrát jemněji než je teoretická mez daná vzorkovací větou. Důležitý je rovněž i výběr vzorkovací mřížky nebo-li plošného uspořádání bodů při vzorkování. V praxi je nejvíce využívaná pravidelná čtvercová mřížka, ta má však někdy problémy se spojitostí obrazové funkce [9]. Každý vzorek odpovídá v digitalizovaném obraze jednomu obrazovému bodu (pixelu).

Kvantování

Druhou částí digitalizace je kvantování vzorků. Velikost jasové úrovně každého vzorku musí být vyjádřena jako číselný údaj. Počet kvantovacích úrovní se volí tak, abychom mohli vyjádřit i jemné detaily v obraze, které je lidský zrak schopen ještě rozeznat. Většinou se používá lineární kvantování do N stejných intervalů, jejichž počet je dán počtem použitých bitů b , $N = 2^b$. V obraze pak počet bitů na jeden pixel udává barevnou hloubku. U monochromatických obrazů je jeden pixel vyjádřen pouze jediným bitem. Barevný model RGB umožňuje při 24 bitové hloubce vyjádřit až 16,7 miliónů barev. Problémem kvantování je stejně jako u jednorozměrných signálů kvantizační šum, který se s jemnějším kvantováním zmenšuje.

2.2 Korekce geometrického zkreslení

Při pořízení obrazu může vlivem nedokonalosti objektivu kamery vznikat geometrické zkreslení obrazu, které se projevuje odlišným tvarem objektů ve výsledném snímku. Nejčastějším zdrojem chyb bývá nevhodná poloha a orientace senzoru vůči snímanému

objektu. K odstranění těchto vad se při zpracování obrazu využívá *geometrické transformace* [9]. Ta na základě souřadnic ve vstupním obraze dokáže určit požadované souřadnice bodů ve výstupním obraze. Umožňuje tak i různé úpravy snímku jako je zvětšování, posouvání, otáčení a mnohé další. Z matematického hlediska dochází ke transformaci souřadnic x, y obrazové funkce $f(x, y)$, které vektorová funkce $\mathbf{T}$ zobrazí do bodu x', y' , tedy

$$x' = T_x(x, y), \quad y' = T_y(x, y). \quad (2.1)$$

Transformační vztahy složek T_x a T_y vektorové funkce $\mathbf{T}$ jsou buď známy předem (rotace, translace, zvětšení obrazu) nebo na základě původního a transformovaného obrazu se snažíme nalézt rovnici, která tuto transformaci popisuje (viz lit. [9]).

2.3 Korekce hodnot jasu

Pro úpravu zkresleného jasu v obraze se používá transformací hodnot jasu. Ty se dělí do dvou skupin na jasové korekce a transformace jasové stupnice (podrobněji lit. [9]).

2.3.1 Jasové korekce

V ideálním případě by měl mít objektiv i snímač stejnou citlivost na světlo pro všechny body snímaného obrazu, v praxi tomu však tak většinou není. V optických soustavách se u čoček projevuje jev zvaný vinětace (viz **kap. 1.3.1**). Rovněž světlocitlivé buňky v různých polohách snímače mohou mít odlišnou citlivost na dopadající záření. Příčinou vad také bývá nerovnoměrné osvětlení snímaného objektu nebo přítomnost drobných částic (např. prach, písek) na čočce objektivu, které pohlcují světlo. Pokud jsou tyto chyby systematické (nevznikají náhodně), můžeme jejich vliv potlačit jasovými korekcemi. K tomu potřebujeme znát odchylku jednotlivých bodů obrazu od ideální převodní charakteristiky. Tato odchylka se nejčastěji vyjadřuje chybovou funkcí $e(x, y)$, jejímž vynásobením původní obrazové funkce $g(x, y)$ získáme pro každý bod hodnotu jasu zkreslené obrazové funkce $f(x, y)$, tzn.

$$f(x, y) = e(x, y) \cdot g(x, y). \quad (2.4)$$

V praxi se tyto chyby potlačují tak, že při stálých světelných podmínkách sejmeme obraz o známém průběhu jasové funkce $g(x, y)$ a poté se snažíme korekci dosáhnout toho, aby se hodnoty jasu digitalizovaného obrazu, co nejvíce blížily originálu. Problémem je, že digitální obrazy mají velikosti jasů jednotlivých pixelů omezeny pouze pro určité rozmezí hodnot. Výsledky po korekci mohou spadat mimo vymezený interval jasů. Provádí se proto buď zhuštění, případně posun jasové stupnice, nebo se hodnoty jasu vně interval nahradí jeho krajními hodnotami [9].

2.3.2 Transformace jasové stupnice

Narozdíl od jasových korekcí nezávisí transformace jasové stupnice na poloze bodu v obraze. Změna hodnot jasu je pro všechny pixely stejná. Použití transformace jasové stupnice v praxi je užitečné především pro úpravy obrazu, které zlepší čitelnost zachyceného obrazu (např. zvýšení kontrastu obrazu) Má však význam pouze pro ta obrazová data, která jsou interpretována a hodnocena člověkem (např. rentgenový snímek).

Realizace transformace jasové stupnice probíhá s využitím vyhledávací tabulky [19]. Ta zabírá paměť o počtu paměťových míst rovných počtu jasových úrovní. Paměť je naplněna novými hodnotami jasu, které vznikly po transformaci. U obrazů ve stupních šedi je počet úrovní nízký (typicky 256 úrovní tvořených 8 bity). Pro barevné obrázky je vyhledávací tabulka vytvořena pro každou ze tří barevných složek R, G, B. Hodnoty obrazové funkce $f(x, y)$ představují indexy do vyhledávací tabulky, jsou tedy zobrazeny barevně. Těmto indexovaným vyhledávacím tabulkám se říká *barevná paleta* [17]. Každý pixel uchovává číslo, které odkazuje na odpovídající položku palety. Zmíněný princip má dvě využití:

I. Pseudobarevné obrazy

Vstupní obraz ve stupních šedi je pomocí vyhledávací tabulky zobrazen pseudobarevně. To znamená, že jednotlivým hodnotám jasů jsou přiřazeny příslušné barvy. Důvodem použití je větší citlivost lidského oka na změnu barvy než na změnu jasu. Takto člověk v obraze rozliší více detailů.

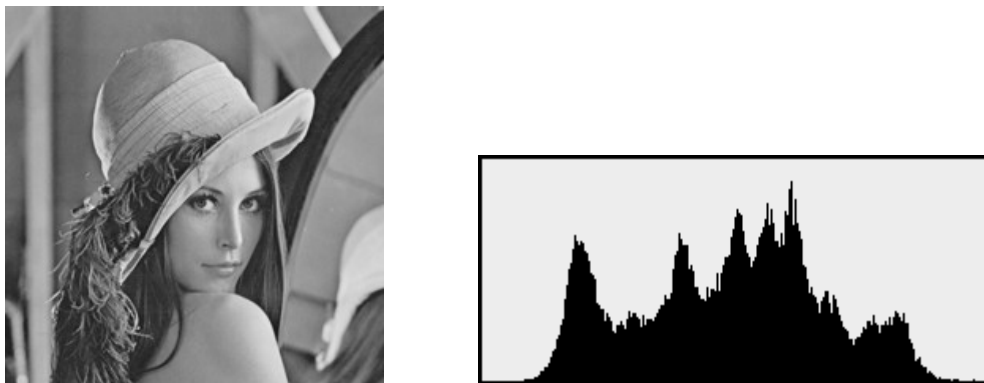
II. Paletové obrazy

Výchozí barevný obraz je kvůli úspoře paměti ztrátově komprimován (viz **kap. 3.1**) do jediného obrazu s N jasovými úrovněmi (N je mocnina 2). Všechny barvy původního obrazu jsou rozděleny do N souvislých skupin, které jsou složeny z podobných barev. Z každé skupiny se vybere jeden typický zástupce (prostřední složka ze všech barev), který bude danou skupinu reprezentovat. Počet N hodnot obrazové funkce představuje indexy do tří vyhledávacích tabulek, přes které se zobrazují vybraní zástupci.

Histogram

Informaci o rozdělení četnosti jasových úrovní v digitálním obraze nám dává *histogram* (**obr. 2.1**) [19]. Je to graf (sloupcový diagram), který zobrazuje, kolik pixelů má příslušnou hodnotu jasu, od černé (hodnota 0) až po zcela bílou (hodnota 255 pro 8-bitovou hloubku šedo-tónového obrazu). Histogram je důležitým pomocníkem pro určení správné expozice (nasvícení) snímku. Určitému obrazu vždy přísluší pouze jeden histogram, obrácené tvrzení však neplatí.

Za účelem zvýšení kontrastu šedo-tónového obrazu se velmi často využívá vyrovnaní jasových složek (ekvalizace) histogramu. Transformace jasové stupnice se snaží dosáhnout toho, aby jednotlivé jasové úrovně měly přibližně stejnou četnost. Po ekvalizaci je zvýšen kontrast úrovní jasu blízko maxim histogramu a snížen kontrast blízko minim histogramu. Častěji zastoupené složky jasu jsou tak v obraze patrnější [9].



Obr. 2.1: Histogram šedo-tónového obrazu pro 256 jasových úrovní

3 KOMPRESSE OBRAZOVÝCH DIGITÁLNÍCH DAT

Nemáme-li k dispozici dostatečnou kapacitu paměti pro uložení existujícího množství zpracovaných digitálních obrazových dat, je nutné toto množství snížit. K tomuto účelu využíváme metodu zvanou *komprese*. Ta usiluje o to, aby ve výsledném obraze byla zredukována redundance (nadbytečnost) a irelevance (zbytečnost) na nejnížší možnou míru. Redundancí se při obrazové kompresi myslí data s informacemi o obrazu, která můžeme vyjádřit menším počtem znaků a nedojde při tom ke ztrátě informace. Irelevance značí množství informace v obrazových datech, které můžeme odstranit, aniž by to ve výsledném obraze postřehlo lidské oko [20]. Odstranění redundance je proces bezeztrátový, odstranění irelevance naopak proces ztrátový.

Jsou-li obrazová digitální data vhodně zpracována, přichází na řadu jejich uložení v požadovaném formátu. Prakticky existují tři způsoby, jak s výslednými daty naložit. Uvedeme je v následujících podkapitolách.

3.1 Nekomprimovaný obraz

Pokud potřebujeme uchovat digitální obraz v původní podobě a se všemi detaily, uložíme jej *nekomprimovaný*. Tento způsob je sice rychlý a spolehlivý, ale nejnáročnější co se týče potřebného místa pro uložení dat. V počítačích se takto uložená obrazová data ukládají např.s koncovkou „.bmp“, což představuje grafický formát BMP (*Microsoft Windows Bitmap*).

3.2 Bezeztrátová komprese

Druhým způsobem je použití *bezeztrátové komprese*. Při použití této metody nedochází ke ztrátě informace, takže po dekódování komprimovaných dat získáme původní obrazová data. Využívá se entropického kódování (nejčastěji Huffmanova kódu), které jednotlivým symbolům přiřazuje kódy podle pravděpodobnosti výskytu symbolu. Princip spočívá v přiřazení nejkratšího kódovacího znaku symbolu s nejčastějším výskytem. Entropické kódování se pak také snaží o predikci právě kódovaného obrazového bodu z předcházejících obrazových bodů. Tyto principy jsou použity v grafických formátech GIF (*Graphics Interchange Format*, „.gif“), PNG (*Portable Network Graphics*, „.png“) a TIFF (*Tag Image File Format*, „.tif“) (viz lit. [16], [19]). Výhodou bezeztrátové komprese je zachování původní informace o digitálním obrazu se současnou úsporou paměti. Velikost uložených obrazových dat je však silně závislá na charakteru obrazu. U obrazu s častými barevnými přechody její účinnost značně klesá.

3.3 Ztrátová komprese

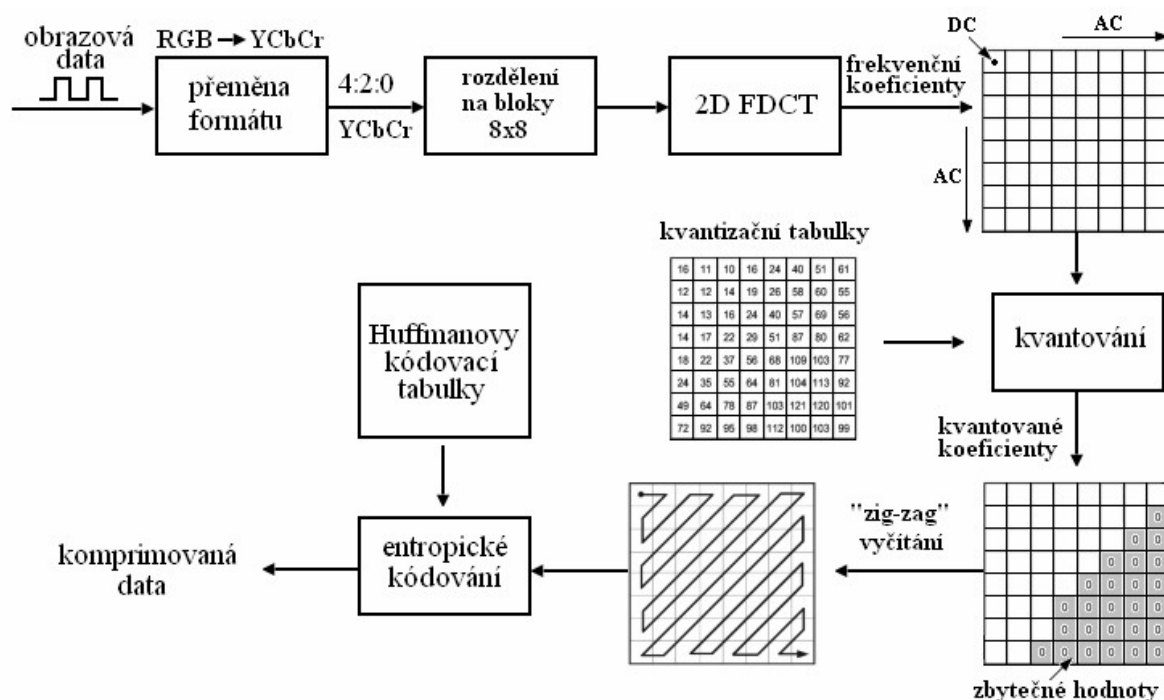
Poslední možností je *ztrátová komprese*. Podstatou této metody je, že lidské oko má větší citlivost na jasové složky obrazu než na složky chrominanční. Člověk rovněž vnímá více nižší frekvence obrazového signálu než vysoké frekvence nesoucí informaci o detailech obrazu. Při komprimaci obrazových dat tedy dochází k nenávratné ztrátě těch informací, na které je lidský zrak nejméně citlivý. Výhodou tohoto postupu je výrazné snížení velikosti ukládaných dat. Ztrátová komprese je tak mnohem efektivnějším nástrojem v úspoře kapacity paměti než komprese bezeztrátová.

4 KOMPRESNÍ ALGORITMY

V praxi byla vyvinuta celá řada ztrátových i bezztrátových kompresních algoritmů, a to jak pro statické snímky, tak i snímky pohyblivé. Pro pohyblivé obrazy je to zejména rodina kompresních standardů MPEG (MPEG-1, MPEG-2, MPEG-4 Part 10, MPEG-4 Visual) [5], jenž se staly normami určující komprimaci a formát přehrávaných obrazových digitálních dat. My se v dalším popisu blíže zaměříme na dva nejznámější a nejpoužívanější kompresní algoritmy pro komprimaci statických obrazů, které využívají standardy JPEG a JPEG2000.

4.1 Algoritmus JPEG

Zkratka je odvozena ze jména pracoviště, které tento kompresní algoritmus definovalo (*Joint Photographic Experts Group*). Standard JPEG podporuje různé režimy operace (módy) v závislosti na požadavku kvality a komprese výsledného snímku. Vychází se z tzv. základního módu, který je založen na sekvenčním kódování (obrázek se kóduje postupně zleva doprava). Postup komprimace je znázorněn na **obr. 4.1** a je rozdělen do několika fází (podrobněji lit. [3], [5], [12], [13], [19], [21], [25]).



Obr. 4.1: Schéma komprese dat u algoritmu JPEG

4.1.1 Úprava vstupních dat

Konverze barevného modelu

Pracujeme-li s barevnými obrazy, nejsou vstupní obrazová data vyjádřena v barevném modelu RGB příliš vhodná pro komprimaci, jelikož změnou jediné složky ovlivníme nejenom barvu, ale také jas obrazového bodu, na který je lidské oko citlivé. Z tohoto důvodu se využívá barevný model YCbCr, kde je oddělena jasová složka (Y) a dvě složky chrominancí

(Cb a Cr). Používá se varianta označovaná jako JPEG–Y'CbCr. Přepočít z barevného modelu RGB na JPEG–Y'CbCr předpokládá všechny složky vyjádřeny jako 8-bitová čísla a je možný podle rovnic uvedených např. v [27].

Změna formátu dat

Dále dochází ke změně formátu digitálního signálu podvzorkováním na formát 4:2:0. To znamená, že počet obou chrominančních složek (Cb a Cr) v matici vzorků je snížen na polovinu v horizontálním i vertikálním směru. Důvodem je menší citlivost oka na barevné složky.

Rozdělení do bloků

Matice vzorků (jasových i chrominančních) se rozdělí na bloky 8x8 bodů (vzorků). Velikost bloku je kompromisem mezi dostatečnou komprimací a výpočetními nároky.

4.1.2 Diskrétní kosinová transformace

Jádrem celé komprimace je nalezení způsobu, jak převést obrazová data do takové formy, abychom mohli oddělit podstatné informace o obraze od méně důležitých. Metody, které toto umožňují, se nazývají transformace (způsob se nazývá transformační kódování). Získáme tak nové vzorky tzv. frekvenční koeficienty, jejichž počet odpovídá původnímu počtu vzorků.

Diskrétní kosinová transformace (anglicky *Discrete Cosine Transform*, zkráceně DCT) je diskrétní transformace vycházející z diskrétní Fourierovy transformace (DFT) [19], ale narušil od ní jsou jejím výsledkem pouze reálné koeficienty složek. Kodér zpracovávající obrazová data používá její dvourozměrnou verzi 2D FDCT (*Two-Dimensional Forward Discrete Cosine Transform*), která je definovaná rovnicí (lit. [5])

$$G(u, v) = \frac{1}{4} C(u) \cdot C(v) \cdot \sum_{x=0}^7 \sum_{y=0}^7 g(x, y) \cdot \cos \frac{(2x+1) \cdot u \cdot \pi}{16} \cdot \cos \frac{(2y+1) \cdot v \cdot \pi}{16}, \quad (4.1)$$

kde $G(u, v)$ je diskrétní obrazová funkce ve frekvenční oblasti, $g(x, y)$ je diskrétní obrazová funkce v prostorové oblasti (za účelem snížení hodnoty výstupních frekvenčních koeficientů posunuta o hodnotu -128), proměnné u, v jsou souřadnice bodu (vzorku) ve frekvenční oblasti, proměnné x, y jsou souřadnice bodu (vzorku) v prostorové oblasti. Pro konstanty $C(u)$ a $C(v)$ platí:

$$C(u) = C(v) = \frac{1}{\sqrt{2}} \quad \text{pro } u = v = 0 \quad \text{a} \quad C(u) = C(v) = 1 \quad \text{pro } u > 0, v > 0.$$

Analogicky lze definovat inverzní diskrétní kosinovou transformaci 2D IDCT (*Two-Dimensional Inverse Discrete Cosine Transform*), kterou využívá dekodér pro zpětnou rekonstrukci dat, tj.

$$g(x, y) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u) \cdot C(v) \cdot G(u, v) \cdot \cos \frac{(2x+1) \cdot u \cdot \pi}{16} \cdot \cos \frac{(2y+1) \cdot v \cdot \pi}{16}, \quad (4.2)$$

přičemž hodnoty obrazových bodů $g(x, y)$ vrátíme do původního intervalu zpětně přičtením čísla 128.

Po provedení 2D FDCT je vyprodukována nová matice koeficientů. Koeficient $G(0,0)$ představuje stejnosměrnou složku (DC), která má největší absolutní hodnotu ze všech koeficientů a nese tak nejvíce podstatné informace o obraze. Každý další koeficient vzdalující se od $G(0,0)$ v horizontálním či vertikálním směru vyjadřuje vyšší frekvenci (AC). Čím větší je vzdálenost určitého koeficientu od koeficientu $G(0,0)$, tím menší je jeho informační hodnota (a tím vyšší kmitočet této složky ve spektru odpovídá).

4.1.3 Kvantování koeficientů

V důsledku různé citlivosti lidského zraku na odlišné prostorové frekvence není nutné uchovávat frekvenční koeficienty se stejnou přesností. Informace o celkovém jasu ve snímku je například mnohem důležitější než malý detail v obraze. Z toho důvodu se zavádí kvantizace koeficientů.

Kvantizační tabulky pro jasové a chrominanční vzorky jsou různé. Pro standard JPEG nejsou tabulky normalizovány, jejich obsahem můžeme měnit účinnost komprimace (kvalitu dekomprimovaného obrazu). Získávají se na základě zkušenosti s výslednou podobou obrazu (empiricky). Stupeň komprimace je určen číslem z rozsahu 1 až 99, které je známé jako faktor kvality komprese, označíme jej fk . Koeficienty $Q_{fk}(u,v)$ kvantizační tabulky se uvádějí pro $fk = 50$ [3]. Pro jiné faktory kvality se kvantizační tabulky přepočítají z $Q_{50}(u,v)$ podle vztahů

$$\begin{aligned} Q_{fk}(u,v) &= Q_{50}(u,v) \cdot \frac{50}{fk} && \text{pro } 1 \leq fk \leq 50, \\ Q_{fk}(u,v) &= Q_{50}(u,v) \cdot \frac{(100 - fk)}{50} && \text{pro } 50 \leq fk \leq 99. \end{aligned} \quad (4.3)$$

Kvantizační matice (tabulka) má stejný rozměr jako je rozměr bloku, tedy 8×8 bodů. Čísla matice frekvenčních koeficientů se dělí odpovídajícími čísly kvantizační tabulky. Vyšší frekvenční složky jsou poděleny většími čísly. Výsledné kvantované koeficienty se zaokrouhlují na nejbližší celá čísla. Kvantování probíhá podle následujícího vzorce

$$G_q(u,v) = \text{round}\left(\frac{G(u,v)}{Q_{fk}(u,v)}\right), \quad (4.4)$$

kde $G(u,v)$ je vstupní matice frekvenčních koeficientů, $G_q(u,v)$ je matice kvantovaných frekvenčních koeficientů a funkce $\text{round}()$ značí zaokrouhlení na celá čísla. Po kvantování zůstanou zachovány pouze koeficienty nesoucí nejdůležitější informace o obraze, ostatní koeficienty jsou nulové. Procesem prahování se zanedbají všechny frekvenční koeficienty, které jsou menší než zvolená prahová hodnota (nejčastěji se volí 0). Zde dochází k nenávratné ztrátě informace o obraze.

V dekodéru se pro dekvantizaci použije vztah

$$R(u,v) = G_q(u,v) \times Q_{fk}(u,v), \quad (4.5)$$

kde $R(u,v)$ značí matici koeficientů vzniklou po dekvantizaci.

4.1.4 Kódování dat

Data po transformaci a kvantizaci je nutno vyjádřit tak, aby zabírala, co nejmenší prostor. Stejnosečná složka a frekvenční složky se kódují zvlášť.

Zakódování DC složky

Koeficient stejnosměrné složky DC se kóduje samostatně pomocí tzv. diferenčního kódování. Jeho hodnota se vyjádří jako rozdíl mezi DC hodnotou právě kódovaného bloku a předchozího bloku. Tento rozdíl je poté kódován pomocí tabulky vytvořené užitím Huffmanova kódu. Tabulka je rozdělena do skupin (kategorií) podle délky bitového slova rozdílu. Nejprve je diferenci přiřazena hodnota symbolu příslušné skupiny a potom je zakódována přídatnými bity samotná difference. Speciálním případem je nulová hodnota difference, které se přiřazuje pouze kód její kategorie.

Kódování AC koeficientů

Abychom zpracovali koeficienty od nejvýznamnějších po ty méně významné, vyčítáme koeficienty z matice od levého horního rohu až po pravý dolní roh metodou zvanou „cik–cak“ (viz **obr. 4.1**). Za poslední nenulový koeficient se vloží značka EOB (*End of Block*, konec bloku), čímž máme definovaný vektor hodnot, které budeme kódovat. Frekvenční koeficienty jsou podrobeny entropickému kódování, kde se aplikují dvě metody – RLE (*Run-Length Encoding*) a Huffmanovo kódování [21].

Princip RLE spočívá v tom, že kóduje sledy za sebou jdoucích stejných znaků do dvojice (počet znaků, symbol znaku). V našem případě to bude vždy znak nula. Pro každou hodnotu nenulového koeficientu se stanoví dvojice znaků. První znak značí počet nul, které předchází nenulovému koeficientu (tzv. délka běhu – *run*), druhý znak udává počet bitů potřebných pro kódování nenulového koeficientu (*size*). Maximální počet nul předcházejících koeficient byl stanoven na 15. Dvojice znaků je zakódována s využitím tabulek sestavených na základě Huffmanova kódu.

Huffmanovo kódování je založeno na četnosti výskytu různě dlouhých symbolů v souboru dat. Symboly, které se v souboru dat vyskytují častěji jsou nahrazeny kódovým slovem s menší délkou bitové posloupnosti. U Huffmanova kódu není žádné kódové slovo různé délky začátkem jiného slova – to umožňuje jednoduché dekódování. Všechna kódová slova jsou sestavena do tabulky, podle níž lze určit četnost výskytu jednotlivých symbolů. Tabulky jsou vytvářeny empiricky a nejsou standardizovány (jsou pouze součástí doporučení).

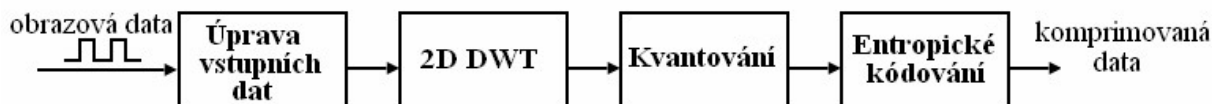
První část výsledného kódu tvoří kódové slovo určené dvojicí (*run/size*). Druhou část kódu vyjadřují přídatné bity, které značí velikost a znaménko nenulového koeficientu. Speciální význam má dvojice (0,0), která označuje EOB (konec kódovaného bloku).

4.1.5 Uložení komprimovaných dat

Na závěr celý zakódovaný blok uložíme a kompresní algoritmus je následně aplikován na dalším bloku obrazu. Komprimovaná obrazová data se ukládají s příponami „.jpg“, „.jpeg“, „.jif“ nebo „.jpe“. Při zpětné dekomprimaci se nejdříve provádí dekódování a dekvantizace a poté se data převedou zpět z frekvenční oblasti pomocí 2D IDCT.

4.2 Algoritmus JPEG2000

Kompresní algoritmus JPEG2000 byl vyvinut skupinou expertů, která se podílela na vytvoření předchozího algoritmu JPEG. Cílem nového algoritmu bylo nejen zlepšení účinnosti komprese, ale rovněž snaha zvýšit odolnost proti chybám při přenosu obrazových dat a vylepšit rozšiřitelnost formátu. Jádrem celého standardu JPEG2000 tvoří Part 1, kde je definován kompresní algoritmus založený na dvourozměrné diskretní vlnkové transformaci (2D DWT – z angl. *Two-Dimensional Discrete Wavelet Transform*). Postup při kompresi dat je popsán v následujícím textu (podrobněji lit. [2], [5], [11], [13], [19], [21]).



Obr. 4.2: Blokové schéma komprese dat u algoritmu JPEG2000

4.2.1 Úprava vstupních dat

Jelikož kódér vyžaduje obrazová data s nulovou střední hodnotou, dojde nejprve k odstranění střední hodnoty (posunu stejnosměrné složky). Stejně jako u algoritmu JPEG musí být obrazová data převedena z barevného modelu RGB do formátu s oddělenou jasovou a dvěma chrominančními složkami, tedy např. YCbCr. Podvzorkování barevných složek v tomto případě není nezbytné, jelikož může být zajištěno samotnou transformací.

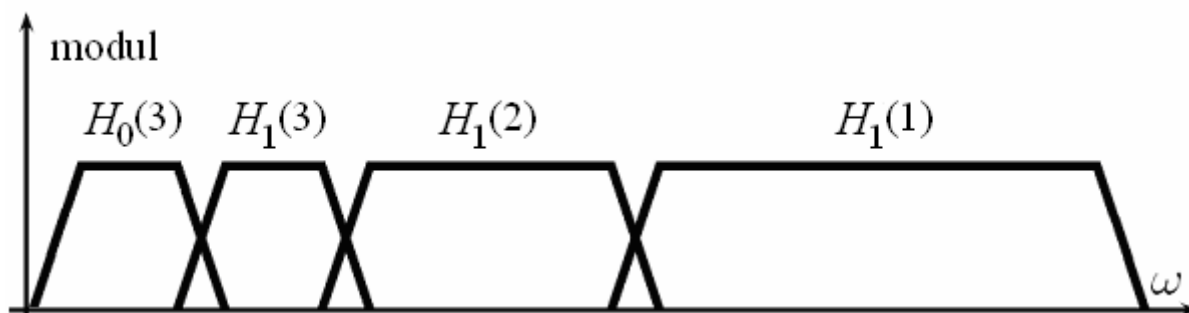
4.2.2 Diskretní vlnková transformace

Diskretní vlnková transformace (DWT) je transformace odvozená z vlnkové transformace pro diskretní vlnky. Jako základní prvky (bázovací funkce) se k rozkladu signálu do jednotlivých složek využívají wavelety – vlnky. Přímá DWT je definována vztahem [22]

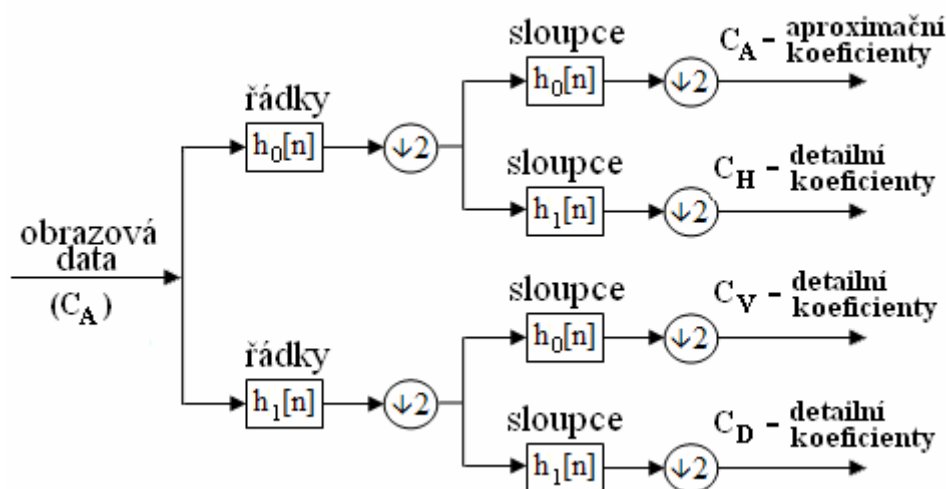
$$S(i, j) = \sum_{n=0}^{N-1} s[n] \cdot \psi_{i,j}[n], \quad (4.6)$$

kde $s[n]$ je vstupní signál, $S(i, j)$ značí vlnkové koeficienty, $\psi_{i,j}[n]$ bázovou funkci, N je délka analyzované posloupnosti a proměnné i, j vyjadřují změnu měřítka a posunutí mateřské vlnky. Při procházení obrazovým signálem v prostoru se hledá jeho podobnost s mateřskou vlnkou v příslušném měřítku. Waveletové koeficienty signálu pak udávají množství informace o podobnosti signálu s vlnkou (smrštění a posunutí waveletu) v určité pozici a měřítku. Využívá se tzv. dyadické dekompozice (viz **obr. 4.3**), při které je změna měřítka a posunutí dána hodnotou 2^j (j je přirozené číslo) [22].

Diskretní vlnková transformace úzce souvisí s lineární filtrací signálu (banka filtrů). Pokud impulsní charakteristika filtru bude představovat časově obrácenou vlnku, můžeme vlnkovou transformaci realizovat s využitím banky filtrů [22]. Analyzovaný signál při DWT prochází sérií filtrů typu dolní (s impulsní odezvou $h_0[n]$) a horní propust (s impulsní odezvou $h_1[n]$) a dochází ke konvoluci s jejich impulsními odezvami. Výstupy z jednotlivých filtrů jsou decimovány (dochází k vypuštění každého druhého vzorku).



Obr. 4.3: Modulové kmitočtové charakt. filtrů pro dyadickou dekompozici úrovně 3



Obr. 4.4: Vlnková transformace obrazu pro 1. úroveň

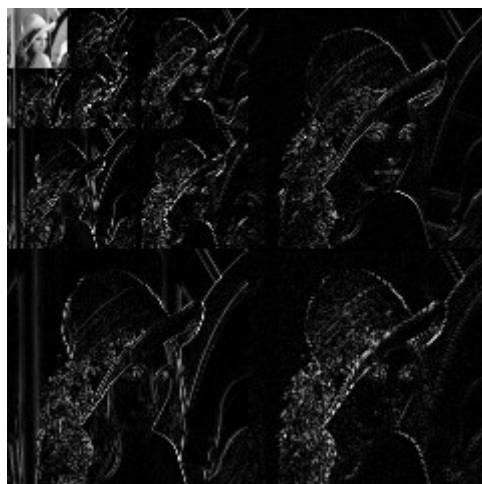
Vlnková transformace obrazu

Při aplikaci DWT na dvourozměrný obrazový signál, využijeme postupně zpracování obrazu po řádcích a následně po sloupcích (**obr. 4.4**) [13]. Z vstupní matice obrazových dat vezmeme postupně všechny řádky a provedeme jejich konvoluci s dekompozičním filtrem typu dolní propust (L) – impulsní odezva $h_0[n]$. Stejnou operaci s řádky provedeme pro dekompoziční filtr typu horní propust (H) – impulsní odezva $h_1[n]$. Výsledek filtrace pro každý řádek u obou filtrů decimujeme (vypustíme každý druhý vzorek). Po zpracování všech řádků získáme dvě matice s polovičním počtem koeficientů oproti matici původní (matice mají poloviční počet sloupců). Nyní vezmeme postupně všechny sloupce z nově vzniklých matic, provedeme jejich konvoluci s filtry L a H a jednotlivé výsledky opět decimujeme. Takto získáme čtyři matice koeficientů, každou s čtvrtinovým počtem koeficientů oproti matici původní. Označují se takto C_A – aproximační koeficienty, C_H – detailní horizontální koeficienty, C_V – detailní vertikální koeficienty, C_D – detailní diagonální koeficienty. Počítání konvoluce s následným podvzorkováním je náročné na výpočetní čas. Optimalizace, u které je snížen počet výpočetních kroků, se označuje jako „Fast lifting wavelet transform“ [13].

Takto jsme provedli dekompozici obrazového signálu pro jednu úroveň (krok). Transformaci můžeme aplikovat opakovaně, jako vstupní matice každé další úrovně se bere právě vypočítaná matice C_A . Jedná se vlastně o rekurzivní proces, kde počet rekurzí udává hloubku dekompozice d . Jako výsledek získáme obrazec, který se nazývá *dekompoziční* (viz **obr. 4.5**).

Pro zmenšení paměťové náročnosti při komprimaci rozměrných obrazů se u JPEG2000 využívá technika nazvaná „tiling“. Vstupní obraz se rozdělí na menší části, označované jako dlaždice (angl. *tile*) a ty se zpracovávají zvlášť.

Důležitým faktorem transformace je volba mateřské vlnky, která ovlivňuje vlastnosti dekompozičních a rekonstrukčních filtrů. Chyby, které se po kvantování v dekódovaném obraze vyskytují, mají právě přímou souvislost s tvarem a průběhem mateřské vlnky. U obrazové komprese se nejčastěji používají vlnky Daubechies, a to ve formě biortogonálních filtrů (filtry typu dolní a horní propust mající rozdílný počet prvků). Ztrátová komprese JPEG2000 využívá 9-ti prvkové filtry typu dolní propust a 7-mi prvkové filtry typu horní propust (filtr 9/7). V případě bezztrátové komprese JPEG2000 se používá filtr 5/3 [13].



Obr. 4.5: Dekompoziční obrazec pro hloubku dekompozice $d = 3$

4.2.3 Kvantování a kódování koeficientů dekompozičního obrazce

Kvantování a následné kódování vlnkových koeficientů se odvíjí podle zvoleného způsobu kvantování. Existují dvě skupiny metod – skalární a vektorová kvantizace.

Skalární kvantizace

Jedná se o základní metodu, u které kvantování příslušné hodnoty koeficientu neovlivňuje koeficienty okolní. Původní hodnoty koeficientů jsou poděleny tzv. kvantizačním krokem, vznikají tak kvantované celočíselné hodnoty. Čím nižší je frekvenční pásmo, tím menší kvantizační krok se v dané úrovni použije. Každý kvantovaný koeficient je srovnáván s určitým prahem, pokud je větší, ponechá se, v opačném případě se neuvažuje a je nahrazen nulou. Zde tedy dochází k nenávratné ztrátě informace.

Po kvantizaci se výsledné hodnoty koeficientů entropicky kódují. Využívá se aritmetického kódování. Obraz je rozdělen na bloky. Tři bloky ležící na stejných pozicích, ale v sousedních subpásmech, se nazývají paketové bloky. Z paketových bloků jsou vytvořeny kódové bloky stejné délky obsahující kvantované koeficienty. V každém bloku jsou bity se stejnou váhou sdruženy do bitových hladin a ty jsou entropicky kódovány. Kodér kóduje všechny bity koeficientů kódového bloku počínaje nejvýznamnějšími bity až po nejméně významné bity. U formátu JPEG 2000 se proces kódování označuje jako EBCOT (*Embedded Block Coding with Optimized Truncation*). Zakódované bloky jsou vkládány do paketů, které vytváří vrstvy a následně úplný datový tok. Výhodou tohoto způsobu kódování je možnost vytváření mnoha podob obrazu (jiné rozlišení, kvalita) ze zakódovaných obrazových dat [5].

Vektorová kvantizace

Efektivnějším nástrojem je vektorová kvantizace [13]. Ta využívá spojitosti mezi stejnými skupinami koeficientů v různých kvantovacích úrovních dekompozičního obrazce. S rostoucí úrovní dekompozice se počet vlnkových koeficientů v matici zmenšuje a koeficienty, které nebyly při decimaci vypuštěny, získávají větší informační hodnotu. V dekompozičním obrazci lze pozorovat stromovitou strukturu, kdy každému vlnkovému koeficientu v některé nižší úrovni odpovídá určitý počet koeficientů o úroveň výše. Tento fakt byl využit při sestavování kódovacích algoritmů, které vycházejí z předpokladu, že určitá skupina koeficientů se bude nacházet na určitém místě.

Algoritmy se nazývají EZW (*Embedded Zerotree Wavelet*) a SPIHT (*Set Partitioning in Hierarchical Trees*) [21]. Jsou rozděleny na jednotlivé kroky, při kterých se hodnoty vlnkových koeficientů v celém stromu srovnávají s určitým prahem. Hodnoty větší než práh se označí jako *signifikantní* (důležité) a zakódují se, ostatní jsou označeny jako *insignifikantní* (nedůležité). Při zvýšení kroku se sníží velikost prahu o 2^{n-1} (n je krok kvantizace a udává počet bitů nutných k vyjádření hodnoty maximálního koeficientu) a s prahem jsou srovnávány koeficienty v minulém kroku označeny jako insignifikantní. Pokud jsou větší, označí se jako signifikantní a jejich kód se připojí na konec kódu z předchozího kroku. Celý postup se dále opakuje. Pro každé snížení prahu se počet bitů reprezentující daný obraz zvětšuje, čímž se rovněž zvyšuje i kvalita výsledného (dekódovaného) obrazu. Toto je velmi výhodné, jelikož v jakékoliv fázi kódování nebo dekodování máme vždy určitou informaci o výsledném obraze.

Vektorové kvantování narozdíl od skalárního však zatím není součástí žádného široce rozšířeného grafického formátu.

4.2.4 Uložení komprimovaných dat

U standardu JPEG2000 se obrázky ukládají s koncovkami „jp2“ nebo „j2k“. Při dekomprimaci se pro zpětnou rekonstrukci obrazového signálu použijí rekonstrukční filtry, jejichž impulsní charakteristika je dána stejnou vlnkou jako v případě filtrů dekompozičních. Signál je ještě před filtrací doplněn nulami na původní velikost (interpolován dvěma). Proces se nazývá inverzní vlnková transformace (IDWT – *Inverse Discrete Wavelet Transform*).

5 KOMPRESNÍ ARTEFAKTY

Komprimace obrazů při vysokém kompresním poměru (nízkém počtu bitů na pixel) je využívána v mnoha vizuálních aplikacích sloužících ke komunikaci mezi uživateli, kdy je kladen důraz především na nízký objem dat. Se zvyšováním kompresního poměru se však u kompresních algoritmů začínají projevovat v obraze nežádoucí artefakty. Těmito artefakty označujeme vady obrazu, které snižují jeho výslednou vizuální kvalitu (jak subjektivní, tak objektivní). Obraz s větším množstvím detailů je při stejném kompresním poměru degradován více než obraz s méně detaily [2].

U výsledné kvality obrazu hledáme vždy kompromis mezi velikostí objemu obrazových dat a přijatelnou mírou degradace obrazu. K nejčastěji se vyskytujícím typům kompresních artefaktů patří tzv. „blocking artifacts“ (blokové artefakty) a „ringing artifacts“ (prstencové artefakty) [10]. My se v následujícím textu blíže zaměříme na popis a možnosti potlačení prvního z nich.

5.1 Blokové artefakty

Tento typ artefaktů se vyskytuje zejména u komprese JPEG, kde je vstupní obraz rozdělen na bloky o rozměrech 8x8 pixelů, které jsou zpracovávány nezávisle na sobě. V důsledku hrubé kvantizace frekvenčních koeficientů při vysokých kompresních poměrech začnou být v obraze patrné hranice jednotlivých bloků. Jelikož je každý blok kódován zvlášť, není jeho obrazový signál korelován se signály sousedních bloků, což se projeví nespojitostmi v hodnotách pixelů na hranicích bloků. Výsledný obraz má nepříjemný „kostkovaný“ vzhled (viz **obr. 8.4** vlevo nahoře), proto se blokové artefakty snažíme co možná nejvíce zredukovat.

5.2 Redukce blokových artefaktů

Existuje celá řada metod, které zmírňují vliv blokových artefaktů (lit. [15], [26]). Tyto metody nejčastěji vycházejí z techniky zvané „post-processing“, kdy jsou upravována obrazová data až po dekodování celého obrazu. Nedochází tak ke zvýšení objemu komprimovaných dat a není potřeba nijak modifikovat kompresní algoritmus. Účinným a jednoduchým řešením je metoda založená na prostorové filtraci (angl. *spatial filtering*) [14].

5.2.1 Metoda prostorové filtrace

Princip této metody spočívá v aplikaci zvoleného filtru na hodnoty pixelů uvnitř obrázku. Konkrétně v našem případě použijeme „vyhlazovací“ filtr na pixely, které se nacházejí v okolí hranice bloků. Tím je dosaženo toho, že ostré hrany na přechodech mezi jednotlivými bloky jsou „rozmáznuty“ do okolních pixelů. Upravený obraz se jeví jako rozostřený, což je však z hlediska subjektivní kvality obrazu přijatelnější vada.

Jako „vyhlazovací filtr“ můžeme obecně použít jakýkoliv lineární filtr typu dolní propust. Konvolucí jeho impulzní odezvy s hodnotami pixelů na okrajích bloků jsou potlačeny parazitní vysoké frekvence, které se tvoří na hranicích mezi bloky. U dvojrozměrných filtrů se jejich diskrétní impulzní odezva označuje jako konvoluční maska. Součet hodnot prvků v konvoluční masce musí být roven jedné, aby při konvoluci nedošlo ke změně celkového jasu. Níže uvedeme příklady lineárních filtrů typu dolní propust [14].

Průměrovací filtr

Jedná se o jednoduchý filtr, u kterého má každý prvek konvoluční masky konstantní hodnotu danou velikostí konvoluční masky. Příklad masky rozměru 3x3 je uveden v rovnici (6.1). Výsledná hodnota pixelu je dána průměrem všech pixelů s umístěním odpovídajícím prvkům v konvoluční masce.

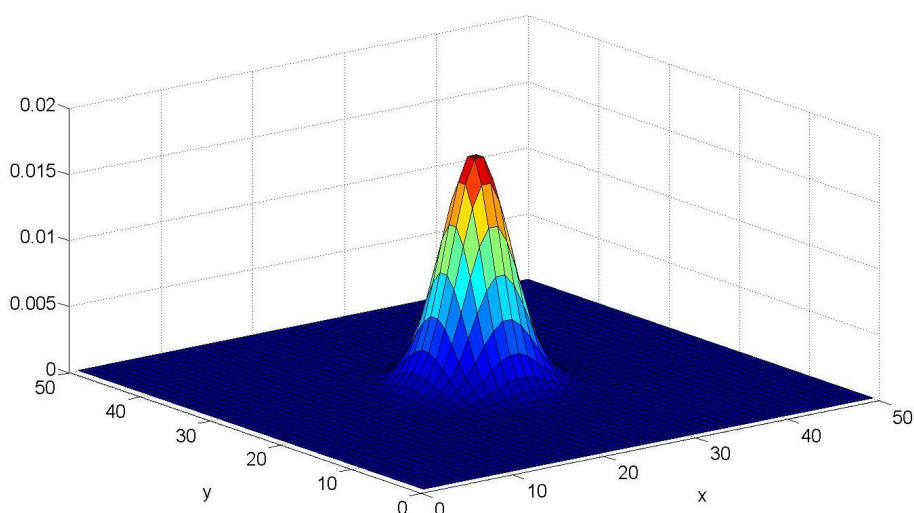
$$\mathbf{M} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \quad (6.1)$$

Gaussov filtr

Tento filtr je speciálním typem dolní propusti, který vychází z Gaussovy funkce rozdělení pravděpodobnosti. Gaussovo dvojrozměrné rozdělení pravděpodobnosti má tvar (při nulových středních hodnotách v obou směrech x i y) [19]

$$\phi(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (6.2)$$

kde $\sigma^2 = \sigma_x^2 = \sigma_y^2$ je rozptyl (σ je směrodatná odchylka). Pro zpracování obrazu je nutno tuto funkci nahradit maticí diskrétních hodnot. Příklad Gaussova dvojrozměrného rozdělení pravděpodobnosti pro $\sigma = 3$ je uvedeno na **obr. 5.1**. Velikost hodnoty σ určuje, jak moc bude filtrovaná hodnota pixelu rozprostřena do okolních pixelů. Zvyšujeme-li hodnotu σ , získáme jako limitní hodnoty Gaussova filtru Průměrovací filtr.



Obr. 5.1: Gaussovo dvojrozměrné rozdělení pravděpodobnosti

6 HODNOCENÍ EFEKTIVITY KOMPRESSE

Při hodnocení efektivity komprese obrazových digitálních dat se u jednotlivých kompresních algoritmů porovnává několik aspektů.

6.1 Kompresní poměr

Na prvním místě je to především velikost výsledných dat. Účinnost komprese se v tomto případě vyjadřuje *kompresním poměrem* (CR), který udává velikost komprimovaných dat vůči velikosti původních dat. Zapisuje se nejčastěji poměrovým způsobem, např. 1:20. Další možností zápisu je vyjádření velikosti komprimovaných obrazových dat v počtech bitů na pixel, např. 0,25 bpp (bits per pixel). Tato hodnota značí, kolik bitů (průměrně) zabírá jeden pixel. Určí se vynásobením počtu bitů na pixel původního obrazu (pb) s podílem velikosti komprimovaného obrazu (vk) a původního obrazu (vp),

$$CR = pb \cdot \frac{vk}{vp} [\text{bpp}]. \quad (5.1)$$

6.2 Kvalita obrázku

Neméně důležitá je vizuální kvalita výsledného (dekomprimovaného) obrazu. K posouzení tohoto hlediska se používají dvě metody. Prvním z nich je subjektivní hodnocení kvality a dalším je hodnocení kvality z objektivního hlediska.

6.2.1 Subjektivní hodnocení obrázku

Tato metoda je závislá na pozorovateli. Subjektivní metoda nemá přesně definovaný postup či matematický vzorec, kvalitu obrazu posuzuje samotný pozorovatel podle parametrů, které mu nejvíce vyhovují. Aby subjektivní hodnocení mělo určitou vypovídající hodnotu, musí se výsledný obraz nechat posoudit statisticky dostatečným počtem lidí.

6.2.2 Objektivní hodnocení obrázku

Pro objektivní měření vizuální kvality komprese se nejčastěji využívá hodnota PSNR (*Peak Signal to Noise Ratio* – špičkový odstup signál/šum). Vypočítá se ze střední kvadratické odchylky MSE (*Mean Squared Error*), která udává rozdíl mezi jednotlivými pixely v původním a dekomprimovaném obraze. Pro barevný obrázek se třemi složkami (RGB) na pixel se MSE definuje [18]

$$MSE = \frac{1}{3 \cdot m \cdot n} \sum_{k=0}^2 \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i, j, k) - K(i, j, k)\|^2, \quad (5.2)$$

kde I a K jsou obrázky o rozměrech $3 \times m \times n$.

Hodnota PSNR se následně vypočítá

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{MAX^2}{\text{MSE}} \right), \quad (5.3)$$

kde MAX značí maximální hodnotu pixelu, což pro 8 bitů na kanál dává $MAX = 255$ [18]. PSNR se udává v decibelech [dB] a typické hodnoty se pohybují v rozmezí 20 až 40 dB.

6.3 Výpočetní náročnost kompresního algoritmu

Pro praktické použití je rovněž podstatná rychlost a náročnost výpočtu algoritmu, a to především z hlediska nároků na velikost paměti a množství dodané elektrické energie při implementaci na signálovém procesoru. Tyto aspekty jsou však silně závislé na množství detailů v původním obraze a jsou tedy značně proměnlivé.

7 REALIZACE KODEKU JPEG V PROSTŘEDÍ MATLAB

Tato kapitola se věnuje praktické implementaci obrazového kodeku (Kodér a DEKodér) JPEG v prostředí MATLAB. Jedná se o skriptovací programovací jazyk – před spuštěním skriptu není potřeba převádět vytvořený program do strojového kódu. Je to ideální nástroj pro práci s obrazovými daty. Veškerá data převádí do podoby matice a obsahuje řadu předtvořených skriptů, které značně usnadňují práci při řešení různých matematických výpočtů. Pro zpracování obrazových dat slouží balíček „Image processing toolbox“, který obsahuje obsáhlou skupinu funkcí.

V následující kapitole bude popsána aplikace, v rámci níž byl vytvořen i obrazový kodek JPEG. Při rozboru se zaměříme spíše na obecný princip činnosti vytvořených funkcí. Bližší popis jednotlivých příkazů představují okomentované zdrojové kódy, které jsou součástí programu na přiloženém CD. Program má název JPEG KoDeK.

7.1 JPEG KoDeK

Jedná se o aplikaci vytvořenou v prostředí MATLAB GUI (*Graphic User Interface*). Toto grafické prostředí umožňuje vytvářet aplikace s uživatelským rozhraním, které poskytuje snadnější a přehlednější ovládání programu.

Hlavní tělo programu tvoří funkce 'kodec_barev.m'. Zde jsou deklarovány všechny globální proměnné a jsou odtud volány ostatní funkce. Grafické rozhraní programu je rozděleno do tří částí, které jsou červeně vyznačeny na **obr. 7.1**. Všechny tři části obsahují okno pro zobrazení výsledné podoby obrázku, údaje o rozměrech obrázku a velikosti jeho dat.



Obr. 7.1: Grafické rozhraní programu JPEG KoDeK

V části 1 se tlačítkem „Načti obrázek“ načítá bitmapový obrázek, který představuje vstupní data kodeku. Posuvná lišta slouží k výběru požadovaného faktoru kvality, čímž je ovlivněn i výsledný kompresní poměr. Stiskem tlačítka „JPEG“ se následně provede komprese obrázku algoritmem JPEG (jak kódování, tak i dekódování). Výsledná podoba dekódovaného obrázku je zobrazena v části 2. Zde jsou také uvedeny informace o kompresním poměru CR a PSNR obrázku (viz **kap. 5**). V panelu s názvem „Výběr filtru“ lze zvolit buď „Průměrování“ nebo „Gaussov“ filtr. U Gaussova filtru je navíc možnost zadat velikost směrodatné odchylky (**kap. 6.2.1**) v požadovaném rozmezí. Označením popisku „Potlačit blokové artefakty“ se zvolený filtr aplikuje na dekódovaný obrázek. Tlačítko „Ulož obrázek“ umožňuje uložit výslednou podobu dekódovaného obrázku. V části 3 lze pomocí tlačítka „Načti obrázek“ načíst ze souboru zvolený typ komprimovaného obrázku – a to JPEG, JPEG2000 nebo JPEG KoDeK, což je dříve uložený soubor s obrázkem komprimovaným v tomto programu. Hodnoty CR a PSNR se zde zobrazují automaticky při shodných rozměrech bitmapového a načteného obrázku.

V dalším textu se blíže zaměříme na princip implementace algoritmu JPEG, který je realizován složením několika samostatných funkcí. Poté budou popsány i některé další části zdrojového kódu, které jsou nezbytné pro správnou funkci programu.

7.1.1 Algoritmus JPEG

Obrazový kodek byl realizován dle obecného postupu uvedeném v **kap. 4.1**. Tohoto postupu se budeme držet i při vysvětlování konkrétní implementace jednotlivých částí algoritmu, avšak zaměříme se spíše na popis některých odlišností a specifických kroků, které jsou spojeny s praktickou realizací kodeku. Uvedeny budou také příslušné funkce, vytvořené za tímto účelem.

Kodek umožňuje komprimaci barevných i šedo-tónových obrázků libovolných rozměrů. Postup při zpracování jasové složky a chrominančních složek je takřka totožný, liší se pouze nepatrně, např. v použitých kvantizačních a kódovacích tabulkách (viz dále). Proto bude popis jednotlivých funkcí společný pro jasovou i chrominanční složky a uvádět budeme pouze případné rozdíly mezi nimi. Nyní již k samotné realizaci.

Konverze barevného modelu

Převod z barevného modelu RGB do YCbCr je proveden využitím předdefinované funkce balíčku Image processing toolbox

$$YCbCr = \text{rgb2ycbcr}(RGB);$$

kde RGB značí obrázek v modelu RGB a YCbCr převedený obrázek v modelu YCbCr.

Úprava velikosti obrázku

Obrázky jejichž rozměry nejsou násobkem osmi je třeba upravit tak, aby mohly být rozděleny na bloky 8x8 px. Řešení je provedeno „zrcadlením“ určitého počtu pixelů z okrajů obrázku do pixelů ležících za hranicemi obrázku tak, aby vyplnily obrázek na potřebný rozměr (násobek osmi). Tím je dosaženo snížení podílu energie ve vyšších frekvencích, čímž je zlepšen kompresní poměr oproti případu, kdyby „vyplňující“ pixely měly pouze nulové hodnoty.

Změna formátu dat

Podvzorkování chrominančních složek je provedeno tak, že každému superpixelu velikosti 2×2 v matici hodnot příslušné barevné složky je přiřazena průměrná barva jeho čtyř pixelů. Takto získáme 2 matice s chrominančními složkami, jejichž rozměry jsou vůči matici s jasovou složkou poloviční – formát 4:2:0.

Rozdělení do bloků

Do matice rozměru 8×8 jsou v každém novém cyklu komprese ukládány sekvenčně vždy další hodnoty pixelů z původního obrázku, které ještě mají být zpracovány kodekem.

Diskrétní kosinová transformace

Kvůli snížení hodnoty výstupních frekvenčních koeficientů je nejprve velikost všech pixelů bloku posunuta směrem dolů o hodnotu -128 (odstranění stejnosměrné složky). Samotná 2D FDCT je realizována opět s využitím již předdefinované funkce MATLABU, která je optimalizována pro rychlejší numerický výpočet:

```
blok_freq = dct2(blok);
```

kde `blok` značí matici s hodnotami pixelů a `blok_freq` matici frekvenčních koeficientů.

Kvantování koeficientů

Koeficienty bloku jsou následně kvantovány kvantizační maticí, jejíž koeficienty se určí z matice $\mathbf{Q}_{50}(\mathbf{u}, \mathbf{v})$ (viz **kap. 4.1.3**) podle zvoleného faktoru kvality. Vytvořená funkce s příslušnými parametry má následující strukturu

```
[blok_kv kvant_tab] = kvantovani(blok_freq, fk, nastav);
```

Vstupní parametr `blok_freq` představuje matici nekvantovaných hodnot, `fk` faktor kvality a `nastav` pomocnou proměnnou. Výstupní parametr `blok_kv` značí matici kvantovaných hodnot a `kvant_tab` použitou kvantizační tabulku. Stejnou strukturu dat má i funkce `kvantovani_chrom` pro kvantování koeficientů chrominančních složek. Rozdíl je pouze v použitých kvantizačních tabulkách $\mathbf{Q}_{50}(\mathbf{u}, \mathbf{v})$. Tyto tabulky jsou odlišné pro jasovou a chrominanční složky. V programu byly použity tabulky, které jsou převzaty z doporučení Mezinárodní telekomunikační unie ITU-T.81 [3] (viz **obr. 7.2**).

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

Jasová složka

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	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	99	99

Chrominanční složky

Obr. 7.2: Použité kvantizační tabulky

Metoda „cik-cak“

Vyčítání koeficientů z matice do vektoru je zajištěno stejnojmennou vytvořenou funkcí

```
[dc_koef ac_koef] = cik_cak(blok_kv); .
```

Parametr `blok_kv` představuje vstupní matici kvantovaných hodnot. Výstupní proměnná `dc_koef` hodnotu DC složky a vektor `ac_koef` hodnoty 63 AC koeficientů. Pořadí vyčítání jednotlivých koeficientů je dáno hodnotami v pomocné matici `cesta`.

Zakódování DC složky

Funkce pro kódování DC složky má tuto podobu

```
[dc_kod, dc_koef_pom] = DC_kodovani(dc_koef, dc_koef_pom); ,
```

kde `dc_koef` udává hodnotu DC složky, `dc_koef_pom` je pomocná proměnná pro ukládání rozdílu DC složek právě kódovaného bloku a předchozího bloku a `dc_kod` je výstupní zakódovaný rozdíl DC složek. Funkce pro kódování DC koeficientů chrominančních složek se nazývá `DC_kodovani_chrom`. První část kódu je tvořena kódovým slovem z Huffmanovy tabulky daný počtem bitů rozdílu DC složek (kategorie). Druhou část kódu tvoří binárně vyjádřený rozdíl DC složek – záporná diference se vyjádří z kladné diference invertováním jednotlivých bitů. Funkce pro jasovou a chrominanční složky se liší Huffmanovými tabulkami použitými ke kódování. Tabulky byly opět převzaty z doporučení ITU-T.81 [3] (viz **obr. 7.3**).

Kategorie	Délka kódu	Kódové slovo	Kategorie	Délka kódu	Kódové slovo
0	2	00	0	2	00
1	3	010	1	2	01
2	3	011	2	2	10
3	3	100	3	3	110
4	3	101	4	4	1110
5	3	110	5	5	11110
6	4	1110	6	6	111110
7	5	11110	7	7	1111110
8	6	111110	8	8	11111110
9	7	1111110	9	9	111111110
10	8	11111110	10	10	1111111110
11	9	111111110	11	11	11111111110

Jasová složka

Chrominanční složky

Obr. 7.3: Použité Huffmanovy tabulky pro kódování DC koeficientů

Kódování AC koeficientů

Hodnoty AC koeficientů jsou kódovány uvnitř této vytvořené funkce

```
ac_blok_kod = AC_kodovani(ac_koef); ,
```

kde `ac_koef` je vstupní vektor 63 AC koeficientů a `ac_blok_kod` je vektor kódovaných hodnot AC koeficientů jednoho bloku. Pro kódování AC koeficientů chrominančních složek je určena funkce `AC_kodovani_chrom`. Kód AC koeficientu se opět skládá ze dvou částí. První část tvoří kódové slovo z Huffmanovy tabulky. Toto slovo nalezneme na příslušném řádku tabulky určeném počtem bitů nenulového koeficientu a počtem nul, který tento koeficient předchází (Run/Size). Druhou část kódu tvoří nenulový koeficient v binární podobě. Za poslední zakódovaný nenulový koeficient se vloží značka EOB, která má speciální kódové slovo. Pokud je počet za sebou jdoucích nul větší než číslo 15, vkládá se značka ZRL (*Zero Run Length*), která reprezentuje 16 nul a má rovněž speciální kódové slovo. Huffmanovy tabulky pro jasovou a chrominanční složky jsou opět odlišné a byly použity dle doporučení ITU-T.81 [3]. Tabulky jsou značně obsáhlé (obsahují 162 kódových slov), proto je na **obr. 7.4** uvedena pouze jejich ukázka.

Run/Size	Délka kódu	Kódové slovo	Run/Size	Délka kódu	Kódové slovo
0/0 (EOB)	4	1010	0/0 (EOB)	2	00
0/1	2	00	0/1	2	01
0/2	2	01	0/2	3	100
0/3	3	100	0/3	4	1010
0/4	4	1011	0/4	5	11000

Jasová složkaChrominanční složky

Obr. 7.4: Ukázka použitých Huffmanových tabulek pro kódování AC koeficientů

Uložení komprimovaných dat

Pro uložení komprimovaného obrázku do souboru byla vytvořena struktura dat, která obsahuje hlavičku a komprimovaná obrazová data (**obr. 7.5**). Hlavička obsahuje informace o rozměrech obrázku (ROZ), použitém faktoru kvality (fk) a příznak (filtrace), který značí, zda-li bylo použito filtrování blokových artefaktů. Zakódovaná data jasové a chrominančních složek všech bloků obrázku jsou uložena jako komprimovaná obrazová data. Z délky tohoto vektoru 0 a 1 je určena výsledná velikost komprimovaných dat. Všechna data jsou ukládána jako datový typ `uint8` (8 bitů na symbol), proto je velikost obrazových dat v souboru osmkrát větší než skutečná velikost komprimovaných dat. Pro odlišení dat jasové a chrominančních složek při dekódování obrázku je mezi obrazová data navíc vložena oddělovací značka EOL. Uložený soubor má příponu „.jbn“.

ROZ (6B)	fk (1B)	filtrace (1B)	komprimovaná obrazová data
----------	---------	---------------	----------------------------

Obr. 7.5: Struktura dat pro uložení komprimovaného obrázku

Dekódování komprimovaného obrázku

Při zpětné rekonstrukci obrázku jsou postupně dekodovány koeficienty jednotlivých bloků. Nejprve DC koeficient a poté zbylé AC koeficienty. Po nalezení značky EOB máme dekodovány všechny koeficienty jednoho bloku. Vytvořené použité funkce se jmenují `DC_dekodovani` a `AC_dekodovani`. Pro chrominanci složky jsou to obdobné funkce `DC_dekodovani_chrom` a `AC_dekodovani_chrom`.

Koeficienty jsou načteny z vektoru dekodovaných hodnot do matice 8x8 pomocí funkce `cik_cak_zpet`. Poté je provedeno dekvantování koeficientů matice s použitím příslušné kvantizační tabulky – funkce `dekvantovani`.

Na závěr jsou koeficienty matice převedeny zpět z frekvenční oblasti pomocí 2D IDCT. Tato transformace je realizována předdefinovanou funkcí `idct2`. Hodnoty pixelů bloku je nutné ještě vrátit zpět do původního intervalu přičtením čísla +128.

Po dekodování všech komprimovaných obrazových dat jsou jednotlivé matice 8x8 poskládány zpět do matice o rozměrech, které jsou násobkem osmi. Obrázky jejichž rozměry nejsou násobkem osmi jsou ořezány na svou původní velikost. Rekonstruovaný obrázek je převeden zpět do modelu RGB s využitím předdefinované funkce `ycbcr2rgb`.

7.1.2 Potlačení blokových artefaktů

K redukci blokových artefaktů byl využit princip metody prostorové filtrace, jejíž teoretický rozbor je uveden v **kap. 6.2.1**. Byly vytvořeny dva možné způsoby potlačení blokových artefaktů – pomocí *Gaussova filtru* nebo jednorozměrné varianty *Průměrovacího filtru*. Pro oba tyto způsoby byly vytvořeny samostatné funkce, které pracují na obdobném principu, kdy jsou filtrovány pixely ležící na hranici mezi bloky (viz **obr. 7.6**).

Gaussův filtr

Zápis vytvořené funkce k redukci blokových artefaktů s využitím Gaussova filtru vypadá následovně:

```
deblok_obr = deblok_gauss(jbn_obr, odchylka);,
```

kde `jbn_obr` představuje vstupní komprimovaný obrázek a `odchylka` velikost zadané směrodatné odchylky (viz **kap. 6.2.1**). Výstupem je pak matice `deblok_obr`, která představuje komprimovaný obrázek s potlačenými blokovými artefakty. Princip funkce spočívá v aplikaci konvoluční masky na dvojice pixelů, které leží na hranici sousedních bloků. Filtrace probíhá postupně ve vertikálním a poté v horizontálním směru. Z důvodu výpočetní náročnosti byla zvolena standardní velikost masky 3x3. Gaussův filtr je definován pomocí skriptu, který je součástí Image processing toolbox

```
gauss_filtr = fspecial('gaussian',[3],odchylka);,
```

kde `odchylka` představuje vstupní hodnotu směrodatné odchylky v rozmezí 0,5–50, „[3]“ je velikost masky a `gauss_filtr` je výsledný Gaussův filtr. K samotné filtraci je pak určen tento předdefinovaný skript

```
filtrovany_obraz = filter2(obr_hran,gauss_filtr,'valid');
```

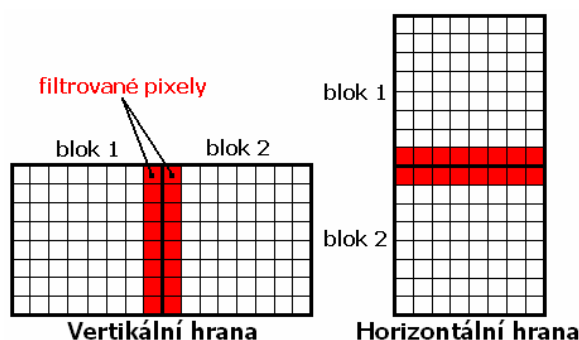
kde `obr_hran` reprezentuje pixely na hranicích jednotlivých bloků obrázku a parametr `'valid'` značí, že budou filtrovány pouze ty pixely obrázku, jejichž hodnotu lze vypočítat pomocí konvoluční masky, bez vyplnění okrajů filtrované oblasti nulami.

Průměrování

Struktura funkce je obdobná jako v předchozím případě:

```
deblok_obr = deblok_prumer(jbn_obr);
```

Použitý způsob filtrace je obdobný jako u Průměrovacího filtru, avšak filtrování není aplikováno konvoluční maskou, ale pouze v určitém směru (jedná se tedy o jednorozměrný filtr typu dolní propust). Princip je následující. Nové hodnoty dvojice pixelů na hranicích mezi sousedními bloky se vypočítají jako jejich aritmetický průměr. Aby se však nevytvořily nové hrany mezi těmito pixely a jejich sousedními pixely téhož bloku v příslušném směru, je hodnota sousedních pixelů posunuta směrem k této nové hodnotě. Toto průměrování hodnot je použito ve vertikálním a následně horizontálním směru. Nevýhodou tohoto způsobu filtrace je značné „rozmazání“ všech hran v obraze.



Obr. 7.6: Princip redukce blokových artefaktů

7.1.3 Program Kakadu

Pro práci s formátem JPEG2000 v prostředí MATLAB je nutné použít externí program, který umožňuje kompresi a dekompresi z/do formátu JPEG2000. Pro potřeby naší aplikace byl zvolen program „Kakadu“, který lze volně stáhnout z internetové adresy [23].

Kakadu je textově orientovaný program bez grafického rozhraní, všechny jeho dílčí programy jsou volány pomocí příkazové řádky. Proto je v MATLABU nutné použít funkci `system(prikaz)`, která z příkazového řádku operačního systému spustí program definovaný v textové proměnné `prikaz`. Kakadu umožňuje zadat celou řadu příkazů, které je možné nalézt v nápovědě k programu Kakadu na přiloženém CD. My si ukážeme použití pouze jediného příkazu v MATLABU, který byl v našem programu využit pro načtení a zobrazení komprimovaného obrázku formátu JPEG2000. Jedná se o dekompresi obrázku, která z původního souboru JPEG2000 vytvoří obrazový soubor formátu BMP:

```
system('kdu_expand -i vstup.jp2 -o vystup.bmp');
```

kde `vstup.jp2` značí vstupní obrázek formátu JPEG2000 a `vystup.bmp` převedený bitmapový obrázek.

8 TESTOVÁNÍ A HODNOCENÍ KOMPRESNÍCH ALGORITMŮ

V této kapitole nejprve provedeme test vytvořeného obrazového kodeku a jeho rozšíření, které umožňuje potlačit blokové artefakty v dekomprimovaném obrázku. V další části se poté zaměříme na srovnání účinnosti kompresních algoritmů JPEG a JPEG2000, které bude rovněž provedeno v programu JPEG KoDeK.

8.1 Test obrazového kodeku JPEG

Při testování kodeku budeme sledovat změnu hodnoty PSNR komprimovaného obrázku při měnícím se faktoru kvality (a tedy i kompresním poměru CR). Dále posoudíme vliv redukce blokových artefaktů na výslednou podobu obrázku. Kodek umožňuje zpracovat bitmapový obrázek (barevný nebo ve stupních šedi) libovolných rozměrů. Z důvodu vyšší náročnosti MATLABU na výpočetní čas při kódování rozměrných obrázků je vhodnější zvolit obrázek menší velikosti, která však bude dostačující k rozeznání všech potřebných detailů. Zvoleny byly proto středně detailní obrázky *Lena.bmp* a *Flower.bmp* o rozměrech 256x256 px a barevné hloubce 24 bpp, které jsou velmi často využívány při podobných testech (viz **obr. 8.1**). Velikost nekomprimovaných obrazových dat, ke kterým se bude vztahovat kompresní poměr, je 196608 B.



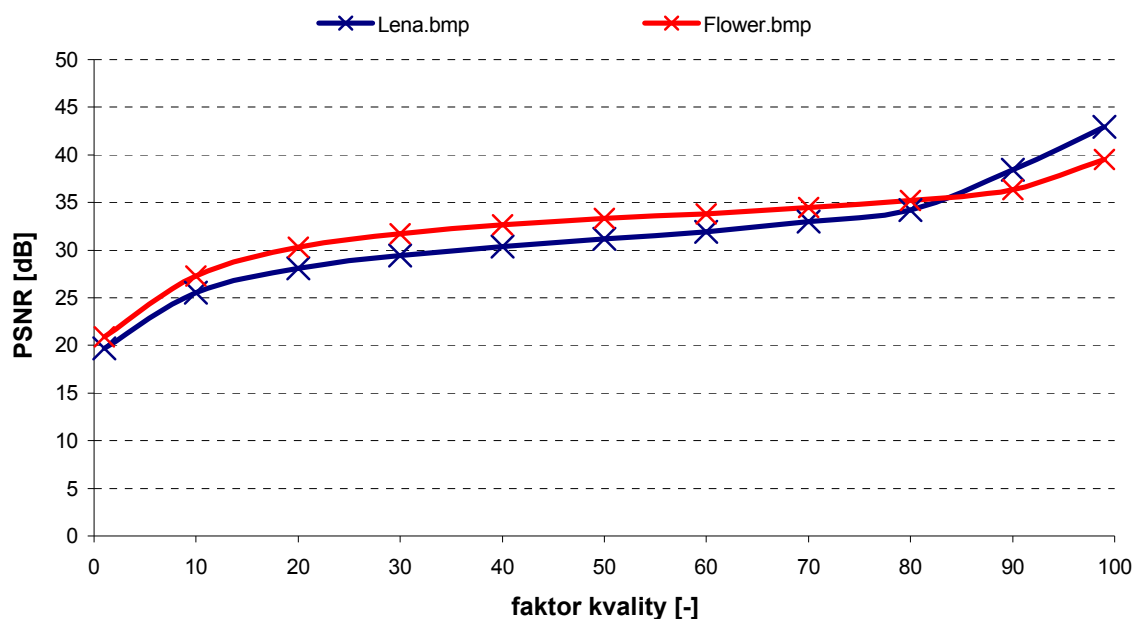
Obr. 8.1: Původní (nahore) a komprimované obrázky, $f_k = 50$

8.1.1 PSNR a CR

V následující tabulce jsou uvedeny hodnoty PSNR a CR pro faktor kvality (fk), který jsme měnili v rozmezí 1–99 s krokem 10. Graf pod tabulkou poté ukazuje v jakém rozpětí se pohybovaly hodnoty PSNR a jak vypadá jejich závislost na měnícím se faktoru kvality.

Tab. 8.1: Naměřené hodnoty závislosti PSNR a CR na faktoru kvality

Lena.bmp											
fk [-]	1	10	20	30	40	50	60	70	80	90	99
CR [bpp]	0,16	0,33	0,49	0,64	0,76	0,89	1,00	1,20	1,50	2,20	4,40
PSNR [dB]	19,72	25,54	28,08	29,45	30,37	31,18	31,90	33,01	34,23	38,46	42,94
Flower.bmp											
fk [-]	1	10	20	30	40	50	60	70	80	90	99
CR [bpp]	0,15	0,26	0,38	0,47	0,55	0,62	0,69	0,82	1,00	1,60	5,00
PSNR [dB]	20,91	27,27	30,29	31,74	32,65	33,30	33,80	34,46	35,24	36,39	39,54



Obr. 8.2: Graf závislosti PSNR na faktoru kvality

Z naměřených hodnot lze vidět, že pro středně detailní obrázek dosahuje kodek maximální kompresní poměr přibližně 1:160 (0,15 bpp), avšak již při velmi špatné kvalitě obrazu (hodnota PSNR menší než 21 dB). Naopak při nejvyšším faktoru kvality se jedná o vynikající kvalitu obrázku (PSNR nad 40 dB) a CR asi 1:5. Kodek je schopen dosáhnout komprese až 1:25 bez znatelnějších vad v obraze (viz **obr. 8.1**).

Poznámka k samotnému testování: V tomto testu jsme neprováděli srovnání s obrázky komprimovanými algoritmem JPEG v rámci jiných aplikací (i když to samotný program umožňuje). Realizovaný kodek vykazuje přibližně o 0,5–1 dB horší hodnotu PSNR při stejném kompresním poměru v závislosti na zvoleném fk . Každá aplikace využívá vlastní kvantizační a kódovací tabulky, takže toto srovnání nemá velkou vypovídací hodnotu. Navíc ani není účelem této práce srovnávat vytvořený program s plně profesionálními aplikacemi.

8.1.2 Vliv redukce blokových artefaktů

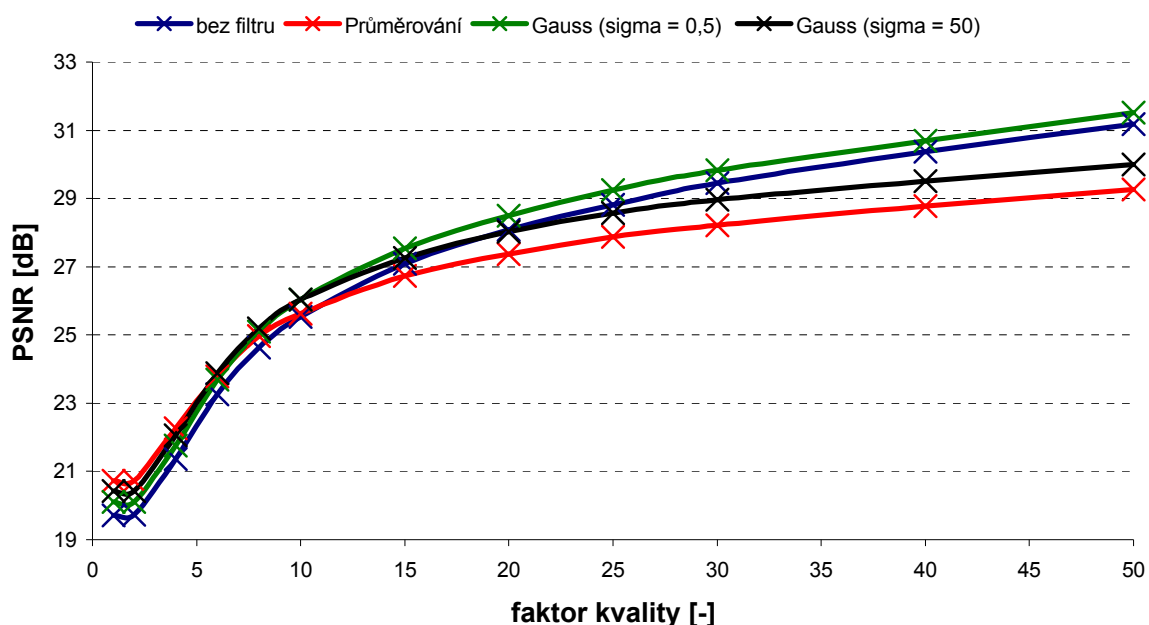
V této části se zaměříme na testování vlivu použitých filtrů na výslednou podobu dekomprimovaného obrázku při redukci blokových artefaktů. Ukazatelem bude subjektivní a objektivní kvalita obrázku vyjádřená hodnotou PSNR. Smysl má samozřejmě redukovat až blokové artefakty, které začnou být v obrázku patrné. Prostorová filtrace při vyhlazování hran mezi bloky je totiž spojená s rozostřením obrázku, což má při nižších kompresních poměrech negativní vliv na výslednou kvalitu obrázku. Testovaným obrázkem je *Lena.bmp*.

Objektivní hodnocení kvality obrázku

Tabulka pod textem uvádí hodnoty PSNR pro různý typ použitého filtru při změně faktoru kvality (fk) v rozmezí 1–50. Pro srovnání jsou uvedeny i hodnoty PSNR nefiltrovaného obrázku. Závislost velikosti PSNR pro odlišný typ použitého filtru jsme vynesli do grafu společně s původními hodnotami PSNR (**obr. 8.3**). Pro větší přehlednost bylo zvoleno měřítko v okolí naměřených hodnot.

Tab. 8.2: Naměřené hodnoty PSNR pro různý typ použitého filtru

–	fk [-]	1	2	4	6	8	10
PSNR [dB]	Bez filtru	19,72	19,73	21,36	23,26	24,63	25,54
	Průměrování	20,73	20,73	22,26	23,79	24,96	25,62
	Gaussův filtr ($\sigma = 0,5$)	20,10	20,11	21,76	23,69	25,10	26,03
	Gaussův filtr ($\sigma = 50$)	20,42	20,42	22,06	23,88	25,20	26,03
–	fk [-]	15	20	25	30	40	50
PSNR [dB]	Bez filtru	27,09	28,08	28,82	29,45	30,37	31,18
	Průměrování	26,73	27,37	27,87	28,21	28,77	29,27
	Gaussův filtr ($\sigma = 0,5$)	27,54	28,50	29,24	29,82	30,69	31,51
	Gaussův filtr ($\sigma = 50$)	27,26	28,02	28,57	28,96	29,51	29,99



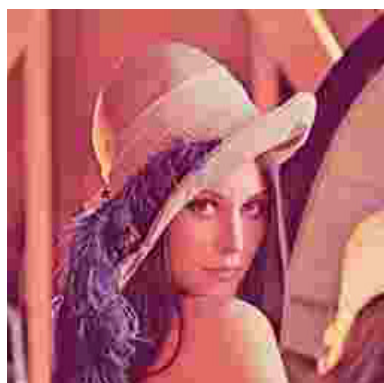
Obr. 8.3: Graf závislosti PSNR na faktoru kvality pro různý typ použitého filtru

Z grafu a z hodnot v tabulce je patrné, že použití určitého filtru k redukci blokových artefaktů je silně závislé na faktoru kvality zvoleném ke kompresi obrázku. Aplikace nevhodného filtru má na výsledný obrázek spíše negativní vliv. Pro vysoký stupeň komprese ($fk < 10$) dochází ke zlepšení hodnoty PSNR pro jakýkoliv typ použitého filtru. Nejlépe si při velmi nízkém faktoru kvality ($fk < 5$) vede metoda *Průměrování*. U takto vysokých stupních komprese je však již obrázek značně deformován a nelze hovořit o nějakém výrazném vylepšení.

Obecně pro nižší hodnoty faktoru kvality, kdy jsou v obraze výrazné blokové artefakty, je výhodnější použít filtr, který tyto ostré hrany více „rozmaže“ a to i za cenu rozostření obrazu. V našem případě to je metoda *Průměrování* a *Gaussův filtr* s vyšší hodnotou směrodatné odchylky ($\sigma = 50$), který vlastně představuje *Průměrovací filtr*. K zjemnění hran mezi bloky pro častěji používané hodnoty faktorů kvality ($fk > 15$) je vhodnější Gaussův filtr s nižší směrodatnou odchylkou ($\sigma = 0,5$).

Subjektivní hodnocení kvality obrázku

K nejlepší redukci blokových artefaktů dochází při použití *Průměrování*. Tato metoda však zároveň značně „rozmazává“ i ostatní ostré hrany v obraze, takže upravený obrázek je dosti rozostřený. Při použití Gaussova filtru sice nedochází k úplnému vymizení hran mezi sousedními bloky, ale za to filtrovaný obrázek není tak výrazně rozostřený, takže jsou lépe patrné některé jeho detaily. Výsledná podoba obrázku při použití různých filtrů a faktoru kvality 10 je zachycena na **obr. 8.4**.



Bez filtru (PSNR = 25,54)



Průměrování (PSNR = 25,62)



**Gaussův filtr $\sigma = 0,5$
(PSNR = 26,03)**



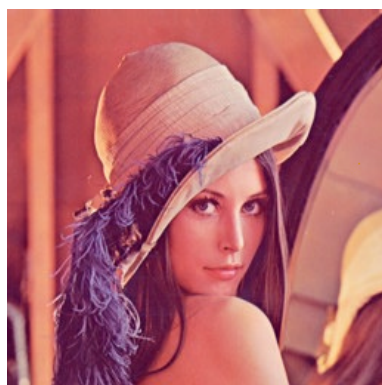
**Gaussův filtr ($\sigma = 50$)
(PSNR = 26,03)**

Obr. 8.4: Subjektivní posouzení kvality obrázku při redukci blokových artefaktů ($fk = 10$)

8.2 Srovnání kompresních algoritmů JPEG a JPEG2000

Nyní se zaměříme na samotné porovnání algoritmů JPEG a JPEG2000. Testování budeme provádět v rámci vytvořené aplikace JPEG KoDeK a hodnotit budeme především objektivní a subjektivní kvalitu komprimovaných obrázků. Ke kompresi obrázků algoritmem JPEG2000 využijeme program JPEG2000 Compressor, který lze volně stáhnout z internetové adresy [1]. Nejprve provedeme kompresi obrázku pomocí vytvořeného kodeku JPEG pro zvolený faktor kvality a následně nastavíme program JPEG2000 Compressor tak, aby dosáhl stejného kompresního poměru. Výslednou podobu obou komprimovaných obrázků poté porovnáme s využitím aplikace JPEG KodeK.

Algoritmy budeme testovat na obrázcích s rozměry 256x256 px a barevnou hloubkou 24 bpp (velikost obrazových dat je 196608 B). Jejich podoba je zobrazena na **obr. 8.5**. Obrázek *Lena.bmp* byl použit již v předešlých testech a spolu s obrázkem *Birds.bmp* představují fotografie se středním výskytem detailů. *Midnight.bmp* reprezentuje fotografii s větším množstvím detailů, a tudíž i více náchylnou na zkreslení při kompresi. Poslední obrázek s názvem *Texture.bmp* představuje barevně oddělené plochy s textem, který obecně není příliš vhodný pro ztrátovou kompresi.



Lena.bmp



Birds.bmp



Midnight.bmp



Texture.bmp

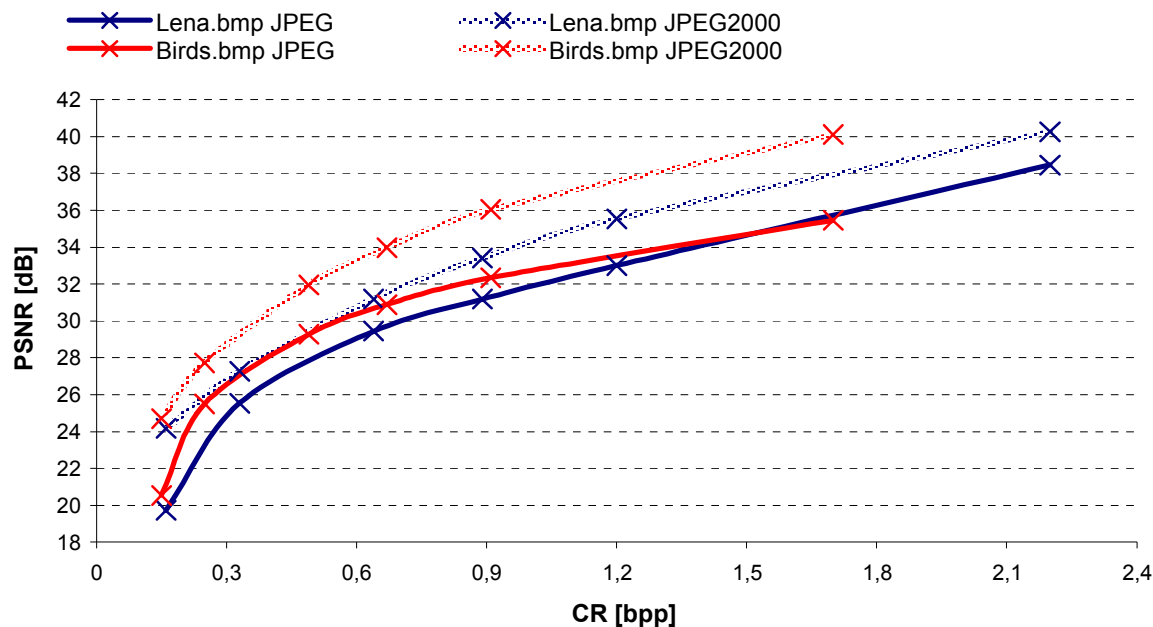
Obr. 8.5: Testované obrázky

8.2.1 Objektivní hodnocení kvality obrázků

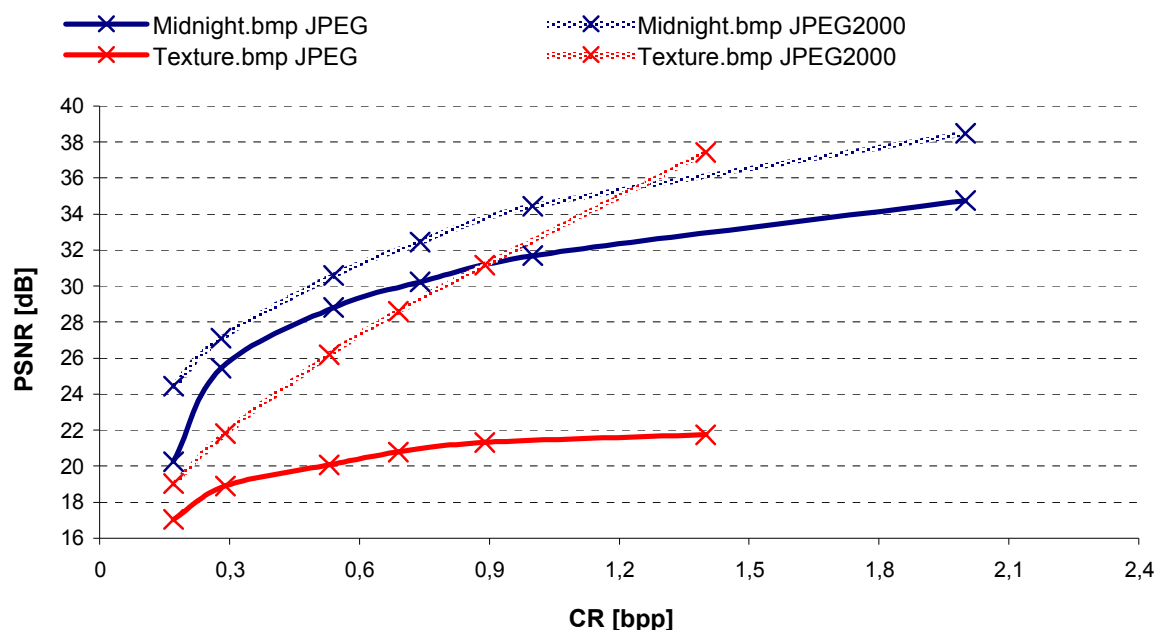
Porovnání naměřených hodnot PSNR pro měnící se kompresní poměr CR je uvedeno v tabulce 8.3. Pod tabulkou jsou rovněž vyneseny grafické závislosti PSNR na CR obou kompresních algoritmů pro všechny testované obrázky.

Tab. 8.3: Naměřené hodnoty PSNR pro měnící se kompresní poměr CR

	JPEG, f_k [-]	1	10	30	50	70	90
Lena.bmp	CR [bpp]	0,16	0,33	0,64	0,89	1,20	2,20
PSNR[dB]	JPEG	19,72	25,54	29,45	31,18	33,01	38,46
	JPEG2000	24,16	27,27	31,18	33,40	35,54	40,26
Birds.bmp	CR [bpp]	0,15	0,25	0,49	0,67	0,91	1,70
PSNR[dB]	JPEG	20,53	25,50	29,27	30,88	32,35	35,45
	JPEG2000	24,70	27,73	31,94	33,97	36,05	40,10
Midnight.bmp	CR [bpp]	0,17	0,28	0,54	0,74	1,00	2,00
PSNR[dB]	JPEG	20,25	25,44	28,80	30,23	31,68	34,73
	JPEG2000	24,44	27,09	30,57	32,44	34,44	38,48
Texture.bmp	CR [bpp]	0,17	0,29	0,53	0,69	0,89	1,40
PSNR[dB]	JPEG	17,03	18,89	20,07	20,78	21,32	21,75
	JPEG2000	19,03	21,81	26,19	28,59	31,16	37,42



Obr. 8.6: Závislost PSNR na kompresním poměru CR (*Lena.bmp*, *Birds.bmp*)



Obr. 8.7: Závislost PSNR na kompresním poměru CR (*Midnight.bmp*, *Texture.bmp*)

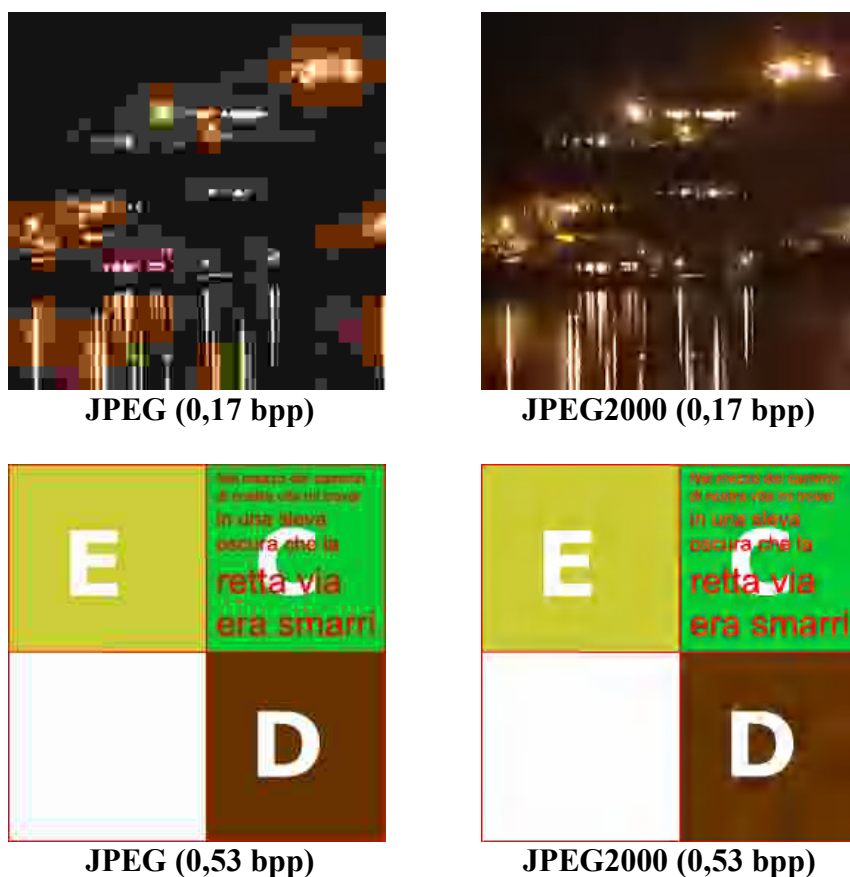
Je zřejmé, že algoritmus JPEG2000 si ve všech případech počínal lépe než JPEG. Pro všechny kompresní poměry si udržoval rozdíl v PSNR minimálně +2 dB, což je dosti znatelný rozdíl především při vyšších kvalitách obrázků (PSNR > 30 dB). Účinnost kompresního nástroje JPEG2000 se začne projevovat při vysokých kompresních poměrech, kde dochází k největším rozdílům v hodnotách PSNR vůči algoritmu JPEG. Obecně lze říci, že tato difference je tím markantnější, čím větší množství detailů má obrázek, který komprimujeme.

Jako naprosto nevhodný se JPEG ukázal při kompresi obrázku *Texture.bmp*, kdy i při vyšších hodnotách faktoru kvality byla textová část špatně čitelná (rozdíl v PSNR vůči JPEG2000 až 16 dB). Kompresní algoritmus JPEG2000 daleko lépe zvládá ostré barevné přechody, jelikož použité báze signálů při waveletové transformaci jsou spíše impulzního charakteru, a proto přesněji vystihují rychlé změny v signálu.

8.2.2 Subjektivní hodnocení kvality obrázků

Co se týče subjektivního posouzení kvality obrazu, tak znatelné chyby v obraze se u obou algoritmů začínaly projevovat až při vyšších kompresních poměrech. Opět v tomto ohledu obstál lépe JPEG2000 jehož zkreslení v obraze se projevovalo daleko méně. U JPEG se při nízkém bitovém poměru začaly objevovat již dříve zmíněné blokové artefakty, které značně znehodnocují výsledný vzhled obrázku. V oblasti ostrých barevných přechodů docházelo u obou kompresních algoritmů k „rozmazávání“. Příčinou je širší spektrum vyšších frekvencí, které tyto přechody obsahují. Vyšší frekvence jsou kvantovány hruběji, případně se zanedbávají úplně.

Nejlépe je patrný rozdíl v účinnosti komprese obou algoritmů při velmi vysokých kompresních poměrech, a to zejména u obrázků s větším množstvím detailů. Zatímco JPEG2000 si „udržuje“ podobu s původním obrázkem, u algoritmu JPEG můžeme jen těžko odhadnout, jak nekomprimovaná obrazová data vypadala. Názorně je tento případ zachycen na **obr. 8.8** (*Midnight.bmp*, CR = 0,17 bpp). Zde je rovněž zobrazen výsledek komprese obrázku *Texture.bmp* pro kompresní poměr 0,53 bpp (povšimněte si zejména čitelnosti textu při kompresi oběma algoritmy).



Obr. 8.8: Porovnání subjektivní kvality obrázků (*Midnight.bmp* – nahoře, *Texture.bmp*)

8.2.3 Shrnutí

V našem testu kompresních algoritmů JPEG a JPEG2000 se potvrdilo, že efektivnějším nástrojem pro komprimaci digitálních obrazových dat je JPEG2000. Obrázky zpracované tímto algoritmem měly při stejném kompresním poměru znatelně vyšší hodnoty PSNR, ale zejména byla patrná výrazně lepší subjektivní kvalita komprimovaných obrázků.

Jelikož komprese obrázků jednotlivými algoritmy probíhala v rámci odlišných aplikací, nebylo možné porovnat jejich výpočetní náročnost. Obecně však platí, že kratší čas pro kompresi (dekompresi) vyžaduje JPEG. U algoritmu JPEG2000 je výpočetní náročnost způsobena realizací transformace DWT, která je prováděna pomocí konvoluce.

Kompresní algoritmus JPEG2000 nabízí oproti JPEG i řadu dalších výhod, jako je např. definování tzv. „oblasti zájmu“ (ROI – angl. *region of interest*) [5], což je část v obraze, která může být komprimována v lepší kvalitě na úkor zbytku obrazu. Důvody, proč se tento formát zatím příliš nerozšířil, i když kvalitativně algoritmus JPEG jasně překonává, je několik. Především je na vině vyšší výpočetní náročnost spojená s vyšším odběrem elektrické energie a dále také skutečnost, že cena paměťových médií neustále klesá (např. paměťových karet do fotoaparátů a mobilních telefonů).

ZÁVĚR

V této bakalářské práci jsme shrnuli obecné postupy pro pořízení a zpracování obrazových dat. Důraz byl kladen především na kompresní algoritmy JPEG a JPEG2000, jakožto účinné nástroje ke snížení objemu již digitalizovaných obrazových dat.

V úvodní kapitole teoretické části jsme si nejprve rozebrali fyzikální podstatu jevů, ze kterých vycházíme při snímání obrazu. Byla uvedena rovnice ozáření jako důležitý element popisující způsob vzniku obrazu z elektromagnetického záření. Princip funkce objektivu vycházejícího z perspektivního zobrazení jsme vysvětlili na idealizovaném modelu dírkové komory. Dále byly uvedeny dvě v současnosti nejpoužívanější technologie pro výrobu snímačů obrazu – CMOS a CCD. Popsána byla struktura plošných snímačů, které jsou nezbytné při snímání dvojrozměrného obrazu v kamerách. S využitím zjednodušeného blokového schématu jsme uvedli používané obvody zpracování signálu v digitálních kamerách.

Další kapitola se zabývá digitálním zpracováním signálu a úpravou obrazových dat získaných ze snímačů. Základ tvoří digitalizace obrazových dat, která se skládá ze dvou částí – vzorkování a kvantování. Naznačeny byly rovněž postupy a možná řešení při korekci geometrického zkreslení a korekcí hodnot jasu.

Třetí kapitola pojednává o kompresi obrazových dat a kompresních algoritmech. Byly zde rozebrány jednotlivé druhy kompresí a grafické formáty, se kterými je možno se setkat. Následující kapitola se poté zabývá studiem principu dvou konkrétních algoritmů využívaných ke komprimaci statických obrazů, jsou to standardy JPEG a JPEG2000. Popsány jsou především konkrétní metody a matematické principy při postupu komprese obrazových dat. Dále je zmíněn problém kompresních artefaktů, které v obraze vznikají při vysokém stupni komprese. Kapitola je věnována zejména možnostem redukce tzv. blokových artefaktů projevujících se u komprese JPEG. Uvedená metoda prostorové filtrace s vybranými příklady lineárních filtrů je následně využita při praktické realizaci. V poslední kapitole teoretické části jsou popsána kritéria, pomocí kterých budeme v praktické části hodnotit účinnost algoritmů JPEG a JPEG2000.

Jako praktická část této práce byla vytvořena aplikace v prostředí MATLAB s názvem JPEG KoDeK. Součástí této aplikace je obrazový kodek JPEG, jenž byl realizován na základě postupu komprese JPEG uvedeném v teoretické části. Rozšíření kodeku tvoří implementované filtrační metody pro možnost potlačení blokových artefaktů v dekomprimovaném obrázku. Účinnost kodeku jsme ověřili na vybraných obrázcích se středním výskytem detailů, kdy kodek dosahoval maximálního kompresního poměru přibližně 1:160. Ukázalo se, že redukce blokových artefaktů je z hlediska objektivního posouzení kvality vhodná především při vysokých kompresních poměrech ($fk < 10$). Její použití je přínosné zejména pro zlepšení subjektivní kvality obrázku při patrném výskytu hran mezi sousedními bloky. Nejlepších výsledků bylo dosaženo aplikací Gaussova filtru s hodnotou směrodatné odchylky $\sigma = 0,5$.

V poslední části práce jsme provedli srovnání kompresních algoritmů JPEG a JPEG2000 v rámci vytvořené aplikace JPEG KoDeK. Pro všechny testované obrázky byla lepší objektivní i subjektivní kvalita komprimovaných obrazových dat při použití JPEG2000. Účinnost tohoto algoritmu se nejvíce projevila při vysokých kompresních poměrech a u obrázků s větším množstvím detailů. Dosažené výsledky potvrdily, že algoritmus JPEG2000 je mnohem sofistikovanějším nástrojem pro kompresi obrazových dat než JPEG, avšak na své rozšíření v praxi stále ještě čeká.

SEZNAM POUŽITÉ LITERATURY

- [1] Anything 3D Corp. *JPEG2000 Compressor* [počítačový program]. [USA] 2000–2006. Dostupné z URL: <http://www.anything3d.com/product/software/freeware.php3#compress>.
- [2] BODEČEK, Kamil. *Kompresa rozměrných obrazů pomocí JPEG2000*. [online]. 2005 [cit. 2007-11-28]. Dostupné z URL: <http://www.elektrorevue.cz/clanky/05011/index.html>.
- [3] CCITT / ITU – *Recomm. T.81* (ISO/IEC IS 10918-1) [online pdf dokument]. 09/1992 [cit. 2008-03-28]. Dostupné z: <http://www.w3.org/Graphics/JPEG/itu-t81.pdf>.
- [4] *CCD* [online]. 2006 [cit. 2007-11-15]. Dostupné z URL: <http://cs.wikipedia.org/wiki/CCD>.
- [5] ČÍKA, Petr. *Multimediální služby* [online elektronická skripta]. Brno: 30.11.2007 [cit. 2008-04-21]. Dostupné z URL: https://www.feec.vutbr.cz/et/skripta/utko/Multimedialni_sluzby_P.pdf.
- [6] HALLIDAY, David, RESNICK, Robert, WALKER, Jearl. *Fyzika: Vysokoškolská učebnice obecné fyziky / Část 4 Elektromagnetické vlny – Optika – Relativita*. 1.vyd., Brno : VUTIUM, 2003. 890-1032 s. ISBN 80-214-1868-0 (brož.).
- [7] HALLIDAY, David, RESNICK, Robert, WALKER, Jearl. *Fyzika: Vysokoškolská učebnice obecné fyziky / Část 5 Moderní fyzika*. 1.vyd., Brno : VUTIUM, 2003. 1034-1198 s. ISBN 80-214-1868-0 (brož.).
- [8] HANUS, Stanislav. *Základy televizní techniky, třetí přednáška: Opto-elektrické měniče*. [ppt dokument]. 2007.
- [9] HLAVÁČ, Václav, SEDLÁČEK, Miloš. *Zpracování signálů a obrazů*. 2. přeprac. vyd., 2007, 255 s. ISBN 978-80-01-03110-0.
- [10] JAKULIN, Aleks. *Baseline JPEG and JPEG2000 Artifacts Illustrated* [online]. 2002–2004, poslední aktualizace 11.3.2005 [cit. 2008-04-21]. Dostupné z URL: <http://www.stat.columbia.edu/~jakulin/jpeg/artifacts.htm>.
- [11] *JPEG2000 Our New Standart!* [online]. 2007 [cit. 2007-11-28]. Dostupné z WWW: <http://www.jpeg.org/jpeg2000/>.
- [12] *JPEG Homepage* [online]. 2007 [cit. 2007-11-20]. Dostupné z URL: <http://www.jpeg.org/jpeg/index.html>.
- [13] MALÝ, Jan. *Srovnání metod pro ztrátovou kompresi obrazu*. [online pdf dokument]. 2006 [cit. 2007-11-20] Dostupné z URL: <http://www.elektrorevue.cz/cz/clanky/zpracovani-signalu/5/srovnani-metod-pro-ztratovou-kompresi-obrazu/>.

- [14] McANDREW, Alsdair. *An Introduction to Digital Image Processing with MATLAB* [online pdf dokument]. 2004 [cit. 2008-04-21]. Dostupné z URL: <<http://sci.vu.edu.au/~amca/SCM2511/book2511.pdf>>.
- [15] Nallaperumal, Krishnan et al. *A Method for Filtering the Blocking Artifacts in Block Coded JPEG Images* [online pdf dokument]. 01/2008 [cit. 2008-04-21]. Dostupné z URL: <http://itfrindia.org/journals/ijise/vol_005_jan2008/pdf/216.pdf>.
- [16] NOVÁK, Jan. *Digitální fotografie a video v praxi*. 1. vyd., 2001. 159 s., ISBN 8024790718.
- [17] *Palette* [online]. 2007 [cit. 2007-11-15]. Dostupné z URL: <http://en.wikipedia.org/wiki/Palette_%28computing%29>.
- [18] *Peak signal-to-noise ratio* [online]. 2007 [cit. 2008-04-21]. Dostupné z URL: <<http://en.wikipedia.org/wiki/PSNR>>.
- [19] RAJMÍČ, Pavel. *Základy počítačové sazby a grafiky* [online pdf dokument]. 2006, poslední aktualizace 29.5.2006 [cit. 2008-03-01]. Dostupné z URL: <http://www.utko.feec.vutbr.cz/~rajmic/bzsg/pdf/BZSG_pracovni_text_2006.pdf>.
- [20] ŘÍČNÝ, Václav, KRATOCHVÍL, Tomáš. *Základy televizní techniky* [online elektronická skripta]. Brno: 2004 [cit. 2007-11-15]. Dostupné z URL: <https://www.feec.vutbr.cz/et/skripta/urel/Zaklady_televizni_techiky_S.pdf>.
- [21] SAYOOD Khalid. *Introduction to Data Compression*, Second Edition, Academic Press /Morgan Kaufmann Publishers: 2000, 636 s., ISBN 1558605584.
- [22] SYSEL, Petr. *Číslíkové filtry, 10. přednáška: Banky filtrů, vlnková transformace*. [pdf dokument]. Brno: 2007 [cit. 2007-11-28].
- [23] TAUBMAN, S. David. *Kakadu* [počítačový program]. Ver. 5 [USA], 2005. Dostupné z URL: <<http://www.kakadusoftware.com/>>.
- [24] The MathWorks, Inc. *MATLAB* [počítačový program]. Ver. 7.1 SP3 [USA], 2005 Dostupné z URL: <<http://www.mathworks.com/>>.
- [25] TIŠNOVSKÝ, Pavel. *Ztrátová komprese obrazových dat pomocí JPEG* [online]. 2006 [cit. 2007-11-20]. Dostupné z URL: <<http://www.root.cz/clanky/ztratova-kompresse-obrazovych-dat-pomoci-jpeg/>>.
- [26] TZOVARAS, Dimitrios et al. *Combined Frequency and Spatial Domain Algorithm for the Removal of Blocking Artifacts* [online pdf dokument]. Hindawi Publishing Corporation 2002 [cit. 2008-04-21]. Dostupné z URL: <<http://www.hindawi.com/GetPDF.aspx?pii=S1110865702203133>>.
- [27] *YCbCr* [online]. 2006 [cit. 2007-11-25]. Dostupný z URL: <<http://en.wikipedia.org/wiki/YCbCr>>.