

Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Bc. Jan Lopata

Odstraňování artefaktů JPEG komprese obrazových dat

Katedra numerické matematiky

Vedoucí diplomové práce: RNDr. Václav Kučera, Ph.D.

Studijní program: Matematika

Studijní obor: Numerická a výpočtová matematika

Praha 2014

Na tomto místě bych chtěl poděkovat RNDr. Václavu Kučerovi, Ph.D. za příkladné vedení diplomové práce. Dále bych chtěl poděkovat svému konzultantovi RNDr. Michalovi Šorelovi, Ph.D. za cenné rady a připomínky z oblasti zpracování obrazu a za poskytnuté testovací fotografie.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Název práce: Odstraňování artefaktů JPEG komprese obrazových dat

Autor: Bc. Jan Lopata

Katedra: Katedra numerické matematiky

Vedoucí diplomové práce: RNDr. Václav Kučera, Ph.D., Katedra numerické matematiky

Konzultant: RNDr. Michal Šorel, Ph.D., ÚTIA AV ČR

Abstrakt: Tato práce se zabývá odstraňováním obrazových artefaktů typických pro JPEG kompresi. Nejprve jsme popsali matematickou formulaci formátu JPEG a problému odstraňování obrazových artefaktů. Poté jsme tento problém přeformulovali na optimalizační problém, kde minimalizovaná funkce je získána pomocí Bayesovy věty a komplexních waveletů. Popsali jsme proximální operátory a algoritmy a aplikovali je na minimalizaci daného funkcionálu. Finální algoritmus jsme implementovali v MATLABu a testovali na několika testovacích problémech.

Klíčová slova: JPEG komprese, optimalizační problém, proximální algoritmy, obrazové artefakty, dvourozměrná waveletová transformace

Title: Removal of JPEG compression artefacts in image data

Author: Bc. Jan Lopata

Department: Department of Numerical Mathematics

Supervisor: RNDr. Václav Kučera, Ph.D., Department of Numerical Mathematics

Consultant: RNDr. Michal Šorel, Ph.D., ÚTIA AV ČR

Abstract: This thesis is concerned with the removal of artefacts typical for JPEG image compression. First, we describe the mathematical formulation of the JPEG format and the problem of artefact removal. We then formulate the problem as an optimization problem, where the minimized functional is obtained via Bayes' theorem and complex wavelets. We describe proximal operators and algorithms and apply them to the minimization of the given functional. The final algorithm is implemented in MATLAB and tested on several test problems.

Keywords: JPEG compression, optimization problem, proximal algorithms, image artifact, two-dimensional spatial wavelet transform

Obsah

| | |
|---|-----------|
| Úvod | 3 |
| 1 JPEG komprese | 4 |
| 1.1 Vznik formátu JPEG | 4 |
| 1.2 Popis relevantní části algoritmu JPEG komprese | 5 |
| 1.2.1 Diskrétní kosinová transformace (DCT) | 5 |
| 1.2.2 Kvantizace a dekvantizace DCT koeficientů | 6 |
| 1.2.3 Proces JPEG komprese a dekomprese | 7 |
| 1.3 Popis JPEG artefaktů | 8 |
| 1.3.1 Blokové artefakty | 9 |
| 1.3.2 Ringing artefakty | 9 |
| 2 Odstraňování JPEG artefaktů | 10 |
| 2.1 Běžně používané přístupy | 10 |
| 2.2 Užité pojmy z teorie pravděpodobnosti | 11 |
| 2.3 Formulace úlohy jako minimalizační problém | 14 |
| 2.3.1 Popis uvažovaného modelu | 15 |
| 2.3.2 Formulace funkcionálu | 18 |
| 2.4 Apriorní informace o obrazu | 20 |
| 3 Použité algoritmy | 22 |
| 3.1 Rozbor minimalizovaného funkcionálu | 22 |
| 3.1.1 Derivace věrohodnostního členu funkcionálu | 23 |
| 3.1.2 Derivace bayes. členu v prostoru DCT koeficientů | 24 |
| 3.1.3 Minimum věrohodnostního členu | 26 |
| 3.2 Proximální algoritmy | 26 |
| 3.2.1 Definice a vlastnosti | 27 |
| 3.2.2 Algoritmus proximální minimalizace | 27 |
| 3.2.3 Metoda proximálního gradientu | 28 |
| 3.2.4 Akcelerovaná metoda proximálního gradientu (APGM) | 28 |
| 3.3 Vyčíslení proximálního operátoru | 29 |
| 4 Popis implementace | 31 |
| 4.1 Obecné informace | 31 |
| 4.1.1 Použité externí podprogramy | 31 |
| 4.1.2 Přehled procedur programu | 31 |
| 4.1.3 Volba globálních parametrů | 32 |
| 4.1.4 Implementace APGM | 33 |
| 4.1.5 Implementace vyčíslení proximálního operátoru | 33 |

| | | |
|----------|---|-----------|
| 5 | Výsledky | 35 |
| 5.1 | Sada testovacích obrázků | 35 |
| 5.2 | Návrh a výsledky numerických experimentů | 36 |
| 5.2.1 | Měření chyby vůči testovacím obrázkům | 36 |
| 5.2.2 | Hledání optimálního parametru alfa | 37 |
| 5.2.3 | Konvergenční chování algoritmů | 38 |
| 5.2.4 | Vlastnosti metody pro nezašuměné obrázky | 40 |
| 5.2.5 | Závislost výkonnosti metody na šumu a kvalitě | 42 |
| 5.2.6 | Srovnání s odšuměním | 42 |
| 5.3 | Obrazové ukázky výsledků | 43 |
| | Závěr | 46 |
| | Seznam použité literatury | 47 |

Úvod

Formát JPEG/JFIF je v dnešní době jedním z nejpoužívanějších digitálních obrazových formátů. Je pojmenován po skupině Joint Photographic Experts Group, založené v roce 1986. Formát je vhodný zejména pro obrazy s velkým počtem barev, spojitými barevnými přechody a nepříliš ostrými hranami. Typicky se jedná o běžné fotografie, naskenované dokumenty a podobně. Naopak není vhodný pro obrazy charakteristické strmými barevnými přechody, malým počtem použitých barev a ostrými hranami. Typicky jde o počítačem psaný text, jednoduchou počítačovou grafiku či kresbu, komiksy a podobné. Zkratka JFIF (JPEG File Interchange Format) označuje formát souboru JPEG obrázku. Tento formát se běžně používá v režimu ztrátové komprese, kdy se obraz pomocí diskrétní kosinové transformace a ořezání koeficientů obrazu převede na řídkou matici. Tuto matici pak lze jistým způsobem úsporně uložit do souboru, tím se však v této práci zabývat nebudeme.

JPEG komprese je ztrátová metoda komprese obrazu, přičemž ztráta dat při kompresi vede k rušivým obrazovým elementům, takzvaným JPEG artefaktům. V této práci navrhne metodu pro jejich odstranění z komprimovaného JPEG obrázku. K tomuto problému existuje řada přístupů. V pracích Yang a kol. (1995) a Zou a Yan (2005) například autoři používají metodu projekcí na konvexní množiny. Hardwarovou implementaci své metody nabízejí autoři v článku Castagno a kol. (1998). V článcích Alter a kol. (2004) a Bredies a Holler (2012) je používán model minimalizace totální variace. V pracích Hsung a kol. (1996) a Nallaperumal a kol. (2006) používají autoři k odstraňování JPEG artefaktů waveletové filtry.

Narozdíl od citovaných prací v našem přístupu navrhne model, ve kterém je „ideální“ obrázek ještě před ztrátovou kompresí zatížen aditivním gaussovským šumem. K řešení problému se dostaneme pomocí minimalizace jistého funkcionálu, vyjadřujícího pravděpodobnost, že daný obrázek je „správný“ za daných informací z JPEG obrázku a apriorní informace. Tuto apriorní informaci získáme pomocí komplexních waveletových filtrů popsanych v článku Kingsbury (2001).

V první kapitole uvedeme formát JPEG, popíšeme algoritmus JPEG komprese a jeho matematické jádro. Připojíme také obrazové ukázky odstraňovaných artefaktů. V druhé kapitole zmíníme existující přístupy k problému odstraňování obrazových artefaktů. Následně pomocí Bayesovy věty formulujeme funkcionál, který bude klíčovým matematickým objektem při řešení našeho problému. Ve třetí kapitole rozebereme způsoby minimalizace zavedeného funkcionálu pomocí proximálních algoritmů. Dále navrhne praktický algoritmus k řešení problému odstraňování artefaktů JPEG komprese. Ve čtvrté kapitole uvedeme implementační podrobnosti navržené metody. V poslední kapitole se budeme věnovat výsledkům naší metody pro jistou sadu testovacích obrázků a provedeme srovnání se vstupními JPEG obrázky.

Kapitola 1

JPEG komprese

1.1 Vznik formátu JPEG

Technologický pokrok v 90. letech 20. století přinesl mnohé možnosti využití digitálního obrazu. Zpracování digitálního obrazu však bylo v té době značně dražší, než tehdy běžné použití analogového obrazu. Klíčovou překážkou v užití digitálního obrazu bylo velké množství dat k přenosu a uchovávání na tehdejší parametry přenosových sítí a paměťových médií. Proto byly hledány způsoby komprese obrazových dat.

Na nový obrazový formát měla podle Wallace (1991) skupina JPEG následující požadavky:

1. Možnost uživatelské volby kvality výsledného obrazu v porovnávání s originálem a tedy možnost volby poměru mezi kvalitou a kompresí.
2. Použitelnost formátu pro libovolný digitální obraz s převážně souvislým barevným tónem.
3. Možnost implementace na velké škále zařízení při rozumné výpočetní složitosti.
4. Přítomnost následujících režimů: sekvenční kódování, progresivní kódování, bezztrátové kódování a hierarchické kódování. (Nás bude zajímat standardní sekvenční kódování. Podrobnosti viz Wallace (1991).)

Jako základ kompresního algoritmu nového obrazového formátu zvolila skupina JPEG diskrétní kosinovou transformaci provedenou na blocích obrazu velikosti 8×8 .

V roce 1992 byla Mezinárodní organizací pro normalizaci vydána první ISO norma týkající se obrazového formátu JPEG navrženého skupinou JPEG.

V následující kapitole představíme proces komprese a dekomprese JPEG obrázku.

1.2 Popis relevantní části algoritmu JPEG komprese

1.2.1 Diskrétní kosinová transformace (DCT)

Definice 1. Nechť čísla $n, m = 0, \dots, 7$ jsou souřadnice obrazových bodů, čísla $u, v = 0, \dots, 7$ jsou souřadnice výsledných DCT koeficientů, $X(n, m)$ značí intenzity jasu obrazového bloku X velikosti 8×8 a $Y(u, v)$ značí DCT koeficienty obrazu X . **Dopřednou DCT transformaci** definujeme vztahem

$$Y(u, v) = \frac{1}{4} C(u) C(v) \left[\sum_{n=0}^7 \sum_{m=0}^7 X(n, m) \cos \frac{(2n+1)u\pi}{16} \cos \frac{(2m+1)v\pi}{16} \right], \quad (1.1)$$

kde $C(0) = 1/\sqrt{2}$ a $C(v) = 1$ pro $v \geq 1$.

Definice 2. **Zpětnou DCT transformaci (IDCT)** definujeme vztahem

$$X(n, m) = \frac{1}{4} \left[\sum_{u=0}^7 \sum_{v=0}^7 C(u) C(v) Y(u, v) \cos \frac{(2n+1)u\pi}{16} \cos \frac{(2m+1)v\pi}{16} \right]. \quad (1.2)$$

Poznámka. V normě probíraného obrazového formátu není specifikován algoritmus, kterým by se měla DCT transformace provádět.

Každý blok zdrojového obrázku velikosti 8×8 lze považovat za 64-bodový diskretní signál, který je funkcí dvou prostorových proměnných n a m . Dopředná DCT transformace tento signál rozloží na násobky bázeových matic

$$\left\{ \frac{1}{4} \cos \frac{(2n+1)u\pi}{16} \cos \frac{(2m+1)v\pi}{16} \right\}_{n,m=0}^7, \quad (1.3)$$

kde $u, v = 0, \dots, 7$. Tyto matice jsou ortogonální v prostoru $l^2(\mathbb{Z}_8 \times \mathbb{Z}_8)$. Bázeové matice odpovídají dvourozměrným prostorovým frekvencím (viz Obrázek 1.1).

Poznámka. Prostorem $l^2(\mathbb{Z}_8 \times \mathbb{Z}_8)$ rozumíme prostor zobrazení $a : \mathbb{Z}_8 \times \mathbb{Z}_8 \rightarrow \mathbb{R}^{8 \times 8}$ splňujících

$$\sum_{i=0}^7 \sum_{j=0}^7 \left(a(i, j) \right)^2 < \infty.$$

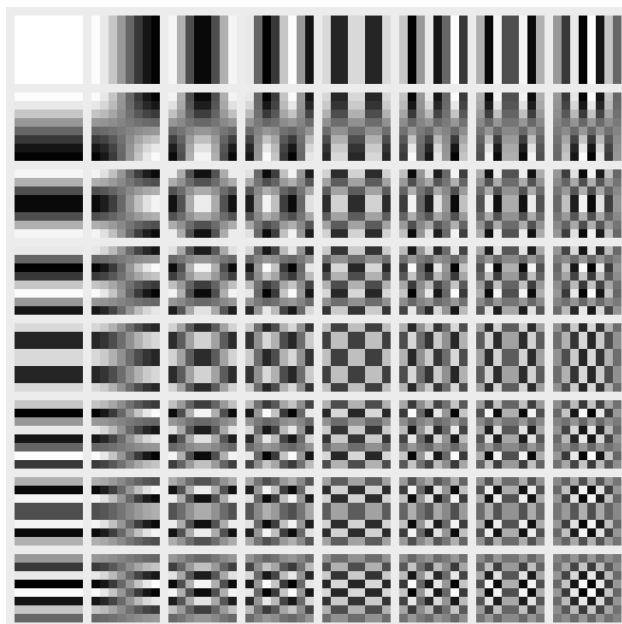
Skalárním součinem zobrazení a, b na tomto prostoru definujeme předpisem

$$(a, b) := \sum_{i=0}^7 \sum_{j=0}^7 a(i, j) b(i, j).$$

Výsledkem DCT transformace je tedy množina 64 „DCT koeficientů“, jejichž hodnoty jsou jednoznačně určeny 64-bodovým vstupním obrazem. Hodnoty DCT koeficientů pak udávají míru zastoupení jednotlivých 2D prostorových frekvencí obsažených ve vstupním obrazu.

Definice 3. Koeficient odpovídající nulové frekvenci v obou dimenzích se nazývá **DC koeficient**, zbývající se nazývají **AC koeficienty**.

Jak jsme již uvedli, báze uvedená ve vztahu (1.3) je ortogonální v prostoru $l^2(\mathbb{Z}_8 \times \mathbb{Z}_8)$ a DCT i IDCT jsou prosté a na. Popsanou transformací tedy nedochází ke ztrátě informace.



Obrázek 1.1: Vizualizace bázových funkcí DCT transformace. Zleva doprava a shora dolů roste frekvence v daném směru.

1.2.2 Kvantizace a dekvantizace DCT koeficientů

DCT transformace je výhodná pro zpracování obrazu se souvislým barevným tónem, protože v takovém obrazu jsou přítomny převážně nízké prostorové frekvence a většina DCT koeficientů je proto blízká nule a může být opomenuta. Účelem kvantizace je dosáhnout komprese zredukováním přesnosti DCT koeficientů na přesnost odpovídající požadované kvalitě.

Definice 4. Symbolem $T : \mathbb{R} \rightarrow \mathbb{Z}$ budeme značit **operátor zaokrouhlení** na nejbližší celé číslo.

Definice 5. Pod pojmem **kvantizační tabulka** budeme rozumět matici Q přirozených čísel v rozsahu 1 až 255 typu 8×8 . Rozšířenou kvantizační tabulkou pro obrázek X rozměrů $N \times M$ dělitelných osmi budeme rozumět matici typu $N \times M$, jejíž bloky jsou rovny matici Q .

Vlastnosti kvantizační tabulky se obvykle shrnují v celé číslo q v rozmezí 0 až 100. Označujeme ho pojmem **kvalita JPEG komprese**. Čím vyšší je číslo q , tím více informace JPEG komprese zachová.

Norma probíraného obrazového formátu nestanovuje, jaké kvantizační tabulky se mají používat. Existuje například velké množství kvantizačních tabulek používaných různými výrobci digitálních fotoaparátů (viz Hass (2008)).

Poznamenejme, že pro různé barevné kanály jednoho obrázku lze použít různé kvantizační tabulky.

Definice 6. Mějme matici DCT koeficientů Y obrazového bloku X . Operátor **kvantizace** DCT koeficientů definujeme vztahem

$$Y^Q(u, v) = T\left(\frac{Y(u, v)}{Q(u, v)}\right) \quad (1.4)$$

pro všechna $u, v = 0, \dots, 7$. Číslem $Y^Q(u, v)$ říkáme **kvantizované DCT koeficienty** obrazového bloku X .

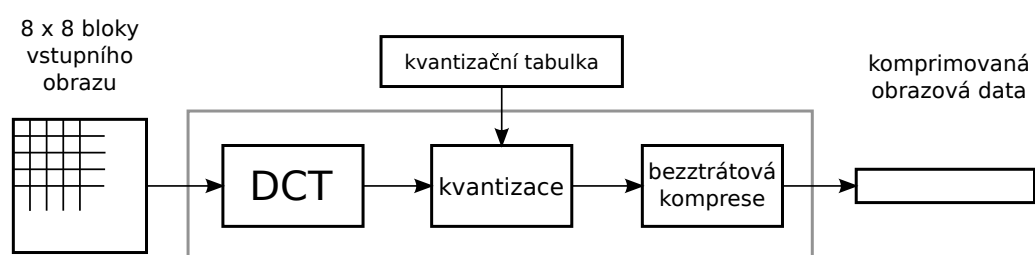
Zobrazení Y^Q zjevně není prosté a jeho zařazením do procesu zpracování obrazu tedy dochází ke ztrátě informace.

Definice 7. Dekvantizací rozumíme operaci definovanou vztahem

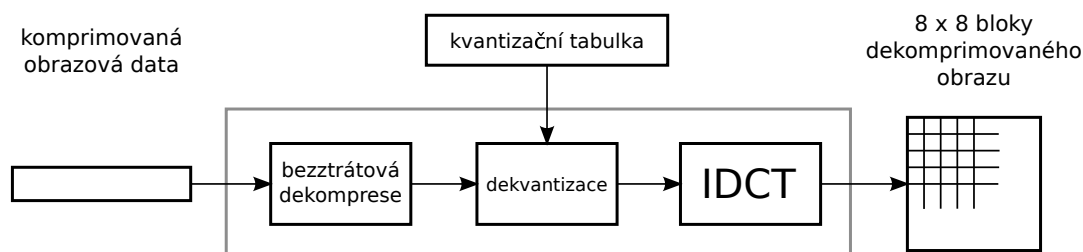
$$Y^{Q'}(u, v) = Y(u, v)Q(u, v), \quad (1.5)$$

čísla $Y^{Q'}(u, v)$ nazýváme **dekvantizované DCT koeficienty**.

1.2.3 Proces JPEG komprese a dekomprese



Obrázek 1.2: Diagram JPEG komprese.



Obrázek 1.3: Diagram JPEG dekomprese.

V následujícím odstavci vysvětlíme průběh JPEG komprese a dekomprese a budeme jej ilustrovat tabulkami a)-f) v souhrnné Tabulce 1.1. Zajímat nás bude především část procesu, při které dochází ke ztrátě informace. Naopak nebudeme rozebírat technické detaily bezztrátové komprese a ukládání dat do souboru.

Komprese obrázku do formátu JPEG je provedena následujícími kroky (viz Obrázek 1.2).

- Nejprve jsou jednotlivé bloky obrazu (viz a)) převedeny DCT transformací na DCT koeficienty (viz b)).
- Tyto koeficienty se s pomocí kvantizační tabulky (viz c)) kvantizují na kvantizované DCT koeficienty (viz d)).
- Nakonec se získaná data bezztrátově zkomprimují do JPEG/JFIF souboru.

| | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-------|-------|------|-------|------|------|------|------|
| 105 | 106 | 112 | 111 | 109 | 111 | 115 | 117 | 925.5 | -15.7 | -2.6 | -10.3 | 3.2 | 7.2 | -1.6 | 3.0 |
| 109 | 110 | 112 | 110 | 120 | 114 | 117 | 119 | -10.9 | -10.6 | -2.2 | 3.9 | 0.3 | -9.9 | -2.7 | 1.6 |
| 110 | 115 | 112 | 116 | 119 | 115 | 120 | 125 | -12.0 | 2.1 | 10.8 | -4.0 | -2.4 | 4.0 | 4.5 | 3.4 |
| 112 | 111 | 116 | 120 | 120 | 116 | 125 | 117 | -5.5 | 1.3 | -8.0 | 1.2 | -0.1 | 0.7 | 3.7 | -4.5 |
| 112 | 119 | 118 | 124 | 122 | 117 | 114 | 119 | -0.3 | 0.8 | 0.4 | -4.6 | -2.0 | 2.4 | -1.5 | -0.2 |
| 109 | 108 | 116 | 125 | 115 | 120 | 121 | 118 | -2.8 | 2.9 | 1.5 | 2.2 | -0.7 | 5.3 | 1.3 | 1.5 |
| 123 | 113 | 121 | 117 | 113 | 112 | 115 | 114 | -0.2 | -8.7 | 0.5 | -4.6 | -1.2 | 2.8 | -1.3 | -5.0 |
| 117 | 113 | 122 | 118 | 107 | 109 | 124 | 123 | 1.4 | 7.1 | 2.3 | -1.9 | 2.4 | -7.0 | 0.5 | -2.0 |

a) Jasové hodnoty pixelů vzorku

b) DCT koeficienty vzorku

| | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----|----|---|---|---|---|---|---|
| 27 | 18 | 17 | 27 | 40 | 66 | 85 | 101 | 34 | -1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 20 | 23 | 32 | 43 | 96 | 100 | 91 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | 22 | 27 | 40 | 66 | 95 | 115 | 93 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | 28 | 37 | 48 | 85 | 144 | 133 | 103 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 30 | 37 | 61 | 93 | 113 | 181 | 171 | 128 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 40 | 58 | 91 | 106 | 134 | 173 | 188 | 153 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 81 | 106 | 129 | 144 | 171 | 201 | 199 | 168 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 120 | 153 | 158 | 163 | 186 | 166 | 171 | 164 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

c) Kvantizační tabulka Q ($q = 30$)

d) Kvantizované DCT koeficienty

| | | | | | | | | | | | | | | | |
|-----|-----|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|
| 918 | -18 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 101 | 103 | 106 | 109 | 112 | 114 | 115 |
| -20 | -20 | 0 | 0 | 0 | 0 | 0 | 0 | 103 | 104 | 106 | 109 | 112 | 114 | 116 | 117 |
| -23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 108 | 109 | 111 | 113 | 116 | 118 | 119 | 120 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 114 | 114 | 116 | 117 | 119 | 120 | 121 | 122 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 117 | 117 | 118 | 119 | 120 | 120 | 121 | 121 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 118 | 118 | 118 | 118 | 118 | 118 | 119 | 119 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 117 | 117 | 117 | 116 | 116 | 116 | 115 | 115 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 116 | 116 | 115 | 115 | 114 | 114 | 113 | 113 |

e) Dekvantizované DCT koeficienty

f) Rekonstruovaný obraz

Tabulka 1.1: Typický průběh JPEG komprese ($q = 30$).

Dekomprese obrázku z JPEG/JFIF souboru probíhá opačným postupem (viz Obrázek 1.3).

- Nejprve ze souboru získá použitá kvantizační tabulka a dekomprimují se kvantizované DCT koeficienty.
- Následně se provede jejich dekvantizace (viz e) v Tabulce 1.1).
- Z dekvantizovaných DCT koeficientů se zpětnou DCT transformací získají bloky dekomprimovaného obrázku (viz f)).

Barevný obraz lze reprezentovat jako složení obrazů tří černobílých složek (například převodem z barevného prostoru RGB do prostoru YCbCr – viz monografie Poynton (2003)) a zpracovává se tedy v principu stejně.

1.3 Popis JPEG artefaktů

JPEG artefaktem rozumíme poruchu obrazu způsobenou JPEG kompresí. Pro formát JPEG jsou typické dva druhy artefaktů. Stručně si je popíšeme.

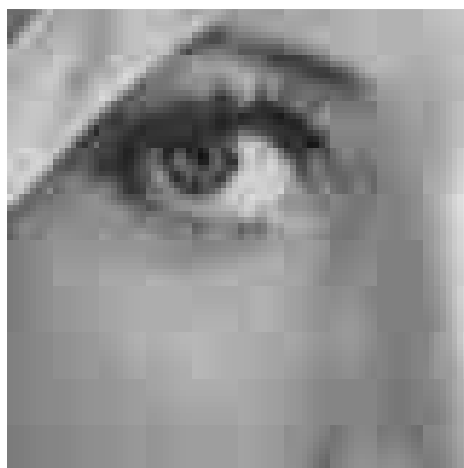
1.3.1 Blokové artefakty

Na Obrázku 1.4 a) najdeme ¹ projevy blokových artefaktů. Jsou patrné jako opakující se struktura skoků mezi sousedními bloky velikosti 8×8 pixelů. V procesu JPEG komprese tyto skoky vznikají hlavně

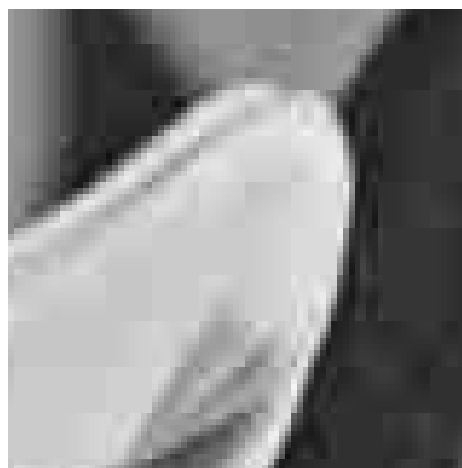
- oříznutím vysokofrekvenčních složek v obrázku,
- zaokrouhlením DC složek sousedních obrazových bloků na znatelně různé hodnoty.

1.3.2 „Ringing“ artefakty

Na Obrázku 1.4 b) lze pozorovat „ringing“ artefakty, viditelné jako oscilace blízko hran. Jsou způsobeny převážně aproximací nespojitých funkcí spojitými funkcemi. Jedná se vlastně o analogii Gibbsova jevu, který se projevuje při Fourierově transformaci.



a) Ukázka blokových artefaktů



b) Ukázka „ringing“ artefaktů

Obrázek 1.4: Ukázka obrazových artefaktů v JPEG obrázku.

¹Obrázky v této práci doporučujeme prohlížet v elektronické verzi.

Kapitola 2

Odstraňování JPEG artefaktů

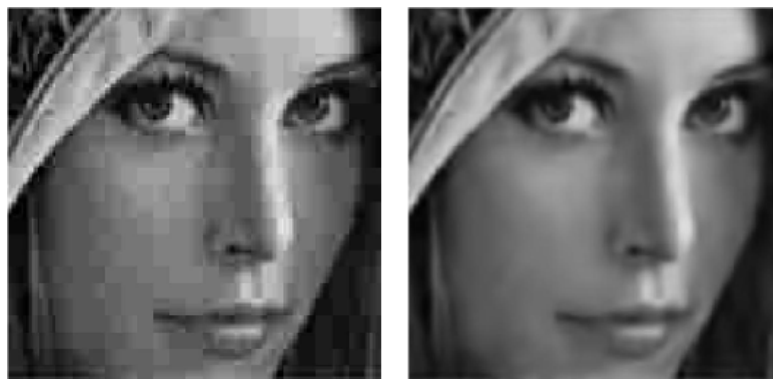
2.1 Běžně používané přístupy

Jedním z přístupů k odstraňování obrazových poruch je metoda projekcí na konvexní množiny (POCS – z anglického projection onto convex sets). Tato metoda je založena na střídajících se úpravách obrazu tak, aby upravený obraz náležel střídavě do jistých konvexních množin, které mají společný průnik. Takto lze nalézt obraz, který bude náležet do všech uvažovaných konvexních množin. Metodu projekcí na konvexní množiny k odstraňování JPEG artefaktů ve svých člancích využili například autoři Yang a kol. (1995) či Zou a Yan (2005).

Jiný přístup k problému odstraňování JPEG artefaktů je založen na minimalizaci totální variace. Tím se zabývali například autoři článků Alter a kol. (2004) a Bredies a Holler (2012). Autoři předpokládají, že původní obraz je více méně po částech konstantní a tvrdí, že takový obraz dobře aproximuje obraz s malou totální variací. Takový předpoklad však v jistých případech vede k výsledkům, na kterých jsou patrné „schodovité“ či „šupinaté“ plochy (viz Obrázek 2.1).



Obrázek 2.1: Ukázka schodovitých ploch při použití minimalizace totální variace. Převzato z článku Bredies a Holler (2012).



Obrázek 2.2: Ukázka výsledku metody používající waveletové filtry. Nalevo: původní JPEG obrázek. Napravo: výsledek metody. Převzato z článku Nallaperumal a kol. (2006).

V článku Nallaperumal a kol. (2006) autoři používají pro odstraňování blokových artefaktů speciální waveletové filtry. Jejich výsledky jsou více uspokojivé (viz Obrázek 2.2). To nás inspirovalo k použití waveletů v naší metodě.

Oproti zmíněným pracím budeme navíc v našem modelu uvažovat náhodný gaussovský šum, jímž je původní obrázek před kompresí zatížen.

2.2 Užité pojmy z teorie pravděpodobnosti

V našem přístupu budeme potřebovat pojmy z teorie pravděpodobnosti. Zavedeme je podle standardních učebnic Dupač a Hušková (2005) a Anděl (2007). K tomu budeme také využívat pojmy z teorie míry a integrálu, ty lze nalézt v knize Lukeš a Malý (2002).

Definice 8. Nechť (Ω, \mathcal{A}) je měřitelný prostor. Prvky množiny Ω budeme nazývat **elementární jevy** a značit ω . Prvky σ -algebry \mathcal{A} budeme nazývat **jevy** a značit velkými písmeny. Symbolem \mathcal{B}^n budeme označovat **borelovské množiny** v \mathbb{R}^n .

Pravděpodobnost P je definována jako míra na \mathcal{A} s vlastností $P(\Omega) = 1$. Jinými slovy se jedná o množinovou funkci na \mathcal{A} s vlastnostmi

- (i) $P(A) \geq 0, A \in \mathcal{A}$,
- (ii) $P(\Omega) = 1, P(\emptyset) = 0$,
- (iii) $P(\cup_{n=1}^{\infty} A_n) = \sum_{n=1}^{\infty} P(A_n)$, je-li $\{A_n\}$ posloupnost po dvou disjunktních jevů.

Trojice (Ω, \mathcal{A}, P) se nazývá pravděpodobnostní prostor.

Definice 9. Nechť $A, B \in \mathcal{A}, P(B) > 0$. **Podmíněnou pravděpodobnost jevu A za podmínky B** definujeme vztahem

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

Věta 1 (Bayesova). Mějme $A, B \in \mathcal{A}$, $P(B) > 0$ Potom platí

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

Důkaz. Okamžitě plyne z definice podmíněné pravděpodobnosti. \square

Definice 10. Jevy A, B jsou **nezávislé**, jestliže platí $P(AB) = P(A)P(B)$.

Definice 11. Nechť (Ω, \mathcal{A}, P) je pevně daný pravděpodobnostní prostor. Měřitelná reálná funkce $X : (\Omega, \mathcal{A}) \rightarrow (\mathbb{R}, \mathcal{B}^1)$ se nazývá **náhodná veličina**.

Měřitelné zobrazení $X : (\Omega, \mathcal{A}) \rightarrow (\mathbb{R}^n, \mathcal{B}^n)$ se nazývá **(n -rozměrný) náhodný vektor**.

Definice 12. Nechť X je náhodná veličina. Její **distribuční funkce** F_X je definována vztahem

$$F_X(x) = P(X \leq x), \quad x \in \mathbb{R}.$$

Definice 13. Distribuční funkce F se nazývá **absolutně spojitá**, jestliže existuje nezáporná, borelovsky měřitelná funkce f taková, že

$$F(x) = \int_{-\infty}^x f(t)dt, \quad x \in \mathbb{R}.$$

Funkce f se nazývá **hustota**.

Definice 14. Buď X náhodná veličina na pravděpodobnostním prostoru (Ω, \mathcal{A}, P) . Její **střední hodnota** $E X$ je definována vztahem

$$E X = \int_{\Omega} X(\omega) dP(\omega).$$

Množinu všech náhodných veličin s konečnou střední hodnotou značíme symbolem $\mathcal{L}_1(\Omega, \mathcal{A}, P)$.

Definice 15. Mějme $n \in \mathbb{N}$. Definujeme **n -tý moment náhodné veličiny** X jako $E X^n$. Množinu všech náhodných veličin s konečným n -tým momentem značíme $\mathcal{L}_n(\Omega, \mathcal{A}, P)$.

Definice 16. **Rozptyl** náhodné veličiny X definujeme vztahem

$$\text{var } X = E (X - E X)^2.$$

Tvrzení 2. Pro $X \in \mathcal{L}_1$, $a, b \in \mathbb{R}$ platí

$$E (aX + b) = a E X + b.$$

Pro $X \in \mathcal{L}_2$, $a, b \in \mathbb{R}$ platí

$$\text{var}(aX + b) = a^2 \text{var } X, \quad \text{var } X = E X^2 - (E X)^2.$$

Důkaz. Důkaz nalezneme v knize Dupač a Hušková (2005). \square

Definice 17. Symbolem $N(0, 1)$ značíme **normované normální rozdělení** dané hustotou

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{x^2}{2} \right\}, \quad -\infty < x < \infty.$$

Jeho distribuční funkci značíme písmenem Φ , její tvar je

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x \exp \left\{ -\frac{t^2}{2} \right\} dt, \quad -\infty < x < \infty.$$

Snadno lze ukázat, že pro $N(0, 1)$ platí

$$\mathbb{E} X = 0, \quad \text{var } X = 1.$$

Definice 18. **Obecné normální rozdělení** je definováno hustotou

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\}, \quad -\infty < x < \infty,$$

značíme je $N(\mu, \sigma^2)$ a jeho distribuční funkce má tvar

$$F(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^x \exp \left\{ -\frac{(t-\mu)^2}{2\sigma^2} \right\} dt = \Phi\left(\frac{x-\mu}{\sigma}\right), \quad x \in \mathbb{R}.$$

Podobně lze ukázat, že pro $N(\mu, \sigma^2)$ platí

$$\mathbb{E} X = \mu, \quad \text{var } X = \sigma^2.$$

Mezi normální rozdělení budeme počítat i **degenerované rozdělení**, kdy veličina X nabývá hodnoty μ s pravděpodobností jedna. Rozptyl tohoto rozdělení je nulový, budeme jej tedy značit $N(\mu, 0)$.

Definice 19. Nechť $X = (X_1, X_2, \dots, X_n)^T$ je n -rozměrný náhodný vektor. Jeho **distribuční funkce** je definována vztahem

$$F_X(x_1, x_2, \dots, x_n) = \mathbb{P} \left(\bigcap_{i=1}^n \{X_i \leq x_i\} \right), \quad -\infty < x_i < \infty, \quad 1 \leq i \leq n.$$

Definice 20. Nechť $\mathbb{E} X^2 < \infty$ a $\mathbb{E} Y^2 < \infty$. **Kovariance** $\text{cov}(X, Y)$ náhodných veličin X, Y je definována předpisem

$$\text{cov}(X, Y) = \mathbb{E} (X - \mathbb{E} X)(Y - \mathbb{E} Y).$$

Definice 21. Nechť $X = (X_1, X_2, \dots, X_n)^T$ je náhodný vektor, jehož složky mají konečný druhý moment. **Varianční matice** $\text{var } X$ tohoto náhodného vektoru je matice velikosti $n \times n$ s prvky

$$\text{cov}(X_i, X_j) = \mathbb{E} (X_i - \mathbb{E} X_i)(X_j - \mathbb{E} X_j), \quad 1 \leq i, j \leq n.$$

Definice 22. Buď $X = (X_1, X_2, \dots, X_n)^T$ náhodný vektor, $\mu = (\mu_1, \mu_2, \dots, \mu_n)^T$ je daný vektor a $V = (\sigma_{ij})$ symetrická pozitivně semidefinitní matice typu $n \times n$. Řekneme, že X má **n -rozměrné normální rozdělení** s parametry (μ, V) , jestliže libovolný vektor $c \in \mathbb{R}^n$ má náhodná veličina $c^T X$ rozdělení $N(c^T \mu, c^T V c)$.

Tvrzení 3. Mějme X_1, \dots, X_n nezávislé náhodné veličiny s rozděleními $N(\mu_j, \sigma_j^2)$. Potom $X = (X_1, \dots, X_n)^T$ má n -rozměrné normální rozdělení a varianční matice V vektoru X má tvar

$$V = \begin{pmatrix} \sigma_1^2 & & & 0 \\ & \sigma_2^2 & & \\ & & \ddots & \\ 0 & & & \sigma_n^2 \end{pmatrix}.$$

Důkaz. Důkaz nalezneme v knize (Jacod a Protter, 2000, Kapitola 16). \square

Tvrzení 4. Necht $X = (X_1, X_2, \dots, X_n)^T$ je náhodný vektor s rozdělením $N(\mu, V)$. Buď dále $a \in \mathbb{R}^m$ a $A \in \mathbb{R}^{m \times n}$. Pak má náhodný vektor $Y = a + AX$ rozdělení $N(a + A\mu, AVA^T)$.

Důkaz. Důkaz nalezneme v knize Anděl (2007). \square

Tvrzení 5. Necht $X = (X_1, X_2, \dots, X_n)^T$ je náhodný vektor s rozdělením $N(\mu, V)$. Složky X_j vektoru X jsou nezávislé právě tehdy, když je varianční matice V vektoru X diagonální.

Důkaz. Důkaz je uveden v knize Jacod a Protter (2000). \square

Tvrzení 6. Mějme X_1, \dots, X_n nezávislé náhodné veličiny s rozděleními $N(0, \sigma^2)$ a ortonormální matici A . Pak $(AX)_1, \dots, (AX)_n$ jsou také nezávislé náhodné veličiny s rozděleními $N(0, \sigma^2)$.

Důkaz. Podle Tvrzení 3 je varianční matice náhodného vektoru X rovna $\sigma^2 I$. Podle Tvrzení 4 má AX n -rozměrné normální rozdělení s varianční maticí

$$\text{var}(AX) = AVA^T = \sigma^2 AIA^T = \sigma^2 I.$$

Matice $\text{var}(AX)$ je tedy diagonální. S použitím Definice 22 a Tvrzení 5 pak již dostáváme, že náhodné veličiny $(AX)_j$ jsou nezávislé a mají požadované rozdělení. \square

Poznámka. Předchozí tvrzení bylo možné takto dokázat pouze pro normální rozdělení. Obecně totiž diagonální varianční matice nezávislost náhodných veličin nezaručuje.

2.3 Formulace úlohy jako minimalizační problém

V této kapitole budeme používat Bayesovský přístup, založený na maximalizaci aposteriorní pravděpodobnosti, jež pro náš model povede k ekvivalentní formulaci problému odstraňování JPEG artefaktů pomocí minimalizace jistého funkcionálu.

2.3.1 Popis uvažovaného modelu

Abychom mohli sestavit vhodný algoritmus na odstraňování artefaktů z obrázku, musíme věnovat pozornost tomu, jak vlastně tento obrázek vznikl. V této sekci sestavíme model popisující vznik obrázku poškozeného JPEG kompresí.

Pro jednoduchost budeme předpokládat, že *rozměry každého obrazu jsou násobky osmi*.

Definice 23. **Originální obraz** velikosti $N \times M$ obrazových bodů budeme reprezentovat jako matici jasových hodnot $X_0 \in \mathbb{R}^{N \times M}$. Jednotlivé obrazové body budeme značit x_0 . Cílem naší metody je nalezení co nejlepšího přiblížení tohoto obrazu.

Definice 24. Každý uvažovaný obraz lze zřejmě rozložit na množinu nepřekrývajících se matic velikosti 8×8 . Ty budeme nazývat **(obrazovými) bloky**. Pro účely DCT transformace budeme obrazový blok chápat tak, jakoby byl přeuspořádán do sloupcového vektoru.

V našem modelu předpokládáme, že originální obraz byl nejprve zatížen aditivním gaussovským šumem, každý obrazový bod nezávisle na ostatních. Symbolicky tuto skutečnost vyjádříme vztahem

$$x_s = x_0 + N(0, \sigma^2).$$

Tento obrazový signál se dále zpracovává po blocích algoritmem JPEG komprese. V následujícím tvrzení ukážeme, že se aditivní šum beze změny přenesení do DCT koeficientů.

Tvrzení 7. *Nechť B je obrazový blok a X je náhodný vektor s 64-rozměrným rozdělením $N(0, I\sigma^2)$. Pak má také náhodný vektor*

$$\text{DCT}(B + X) - \text{DCT}(B)$$

rozdělení $N(0, I\sigma^2)$ a jeho složky jsou nezávislé náhodné veličiny s rozděleními $N(0, \sigma^2)$.

Důkaz. Z linearity DCT plyne

$$\text{DCT}(B + X) = \text{DCT}(B) + \text{DCT}(X).$$

DCT je navíc ortonormální transformace, existuje tedy ortonormální matice A taková, že pro libovolný vektor $x \in \mathbb{R}^{64}$ platí

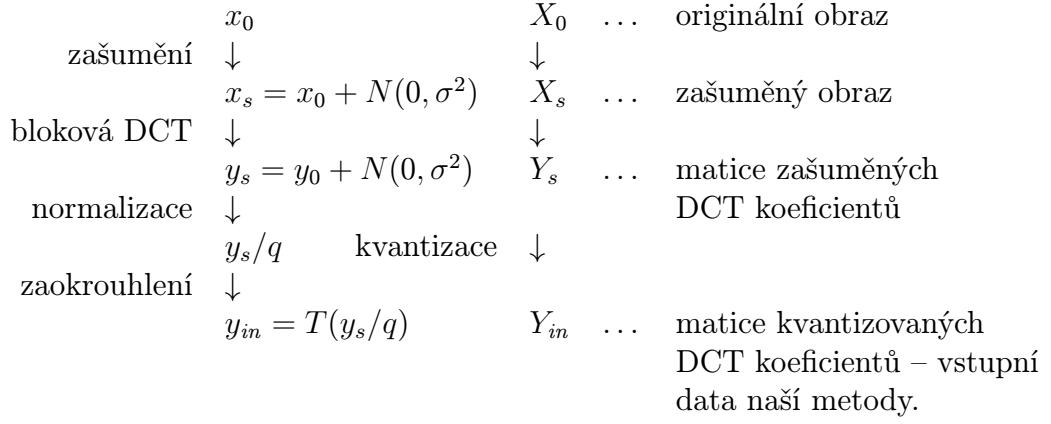
$$\text{DCT}((x_1, \dots, x_{64})^T) = A(x_1, \dots, x_{64})^T.$$

Dohromady tedy máme

$$\text{DCT}(B + X) - \text{DCT}(B) = AX.$$

Díky Tvrzení 4 má náhodný vektor AX požadované rozdělení pravděpodobnosti a z Tvrzení 6 pak dostáváme dokazovanou nezávislost náhodných veličin. \square

Následující část modelu odpovídá JPEG kompresi, přičemž budeme počítat i s aditivním gaussovským šumem.



Obrázek 2.3: Stručné schéma použitého modelu vzniku vstupních dat.

Definice 25. DCT transformaci provedenou zvlášť na každém obrazovém bloku nazýváme **blokovou DCT transformací** a značíme ji symbolem BDCT.

Blokovou DCT transformaci originálního obrazu X_0 budeme značit Y_0 a její prvky y_0 budeme nazývat DCT koeficienty originálního obrazu. Platí vztah

$$BDCT(X_0) \stackrel{def}{=} Y_0.$$

V Tvzení 7 jsme ukázali, že se blokovou DCT transformací nezmění charakter šumu a platí

$$BDCT(X_0 + N(0, I\sigma^2)) = Y_0 + N(0, I\sigma^2).$$

Označujme $y_s = y_0 + N(0, \sigma^2)$. Jednotlivé DCT koeficienty y_s se dále v JPEG kompresi zpracovávají tak, že se jednotlivě podělí číslem q – odpovídajícím prvkem kvantizační tabulky Q – a poté zaokrouhlí na nejbližší celé číslo. Tento proces popíšeme vztahem

$$y_{in} = T(y_s/q).$$

Čísla y_{in} nazýváme kvantizovanými DCT koeficienty obrazu X_s . Matice Y_{in} těchto koeficientů bude vstupem naší metody. Celý model je shrnut v schématu na Obrázku 2.3.

Symbolem X_{in} rozumíme obrázek, který vzniknul vynásobením kvantizovaných DCT koeficientů odpovídajícími prvky matice Q a provedením blokové IDCT. Jinými slovy se jedná o obrázek, který dostaneme standardní JPEG dekompresí.

Pro přehlednost nyní zopakujeme znění Bayesovy věty: Mějme dva náhodné jevy A a B s pravděpodobnostmi $P(A)$ a $P(B)$, přičemž $P(B) > 0$. Potom platí

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \quad (2.1)$$

Při odstraňování JPEG artefaktů se nacházíme v následující situaci. Originální obraz X_0 je neznámý, našimi vodítky jsou kvantizované DCT koeficienty y_{in} a kvantizační tabulka Q . V následujícím odstavci situaci rozebereme pomocí Bayesovy věty.

Předpokládejme, že máme jistý obraz X . Bude nás zajímat, s jakou pravděpodobností je tento obraz roven originálnímu obrazu X_0 za podmínky, že se

originální obraz X_0 popsaným procesem zašumění a JPEG komprese transformoval na DCT koeficienty Y_{in} . Označujeme

$$P(\{X = X_0\} | \{X_0 \rightarrow Y_{in}\}) = P(A|B). \quad (2.2)$$

Budeme hledat takový obraz X , že pravděpodobnost (2.2) je maximální. Její vyčíslení provedeme pomocí Bayesovy věty. Zavedeme ještě následující značení:

- $P(B) = P(\{X_0 \rightarrow Y_{in}\})$ je pravděpodobnost toho, že vstupní data – kvantizované DCT koeficienty Y_{in} – vznikla procesem zašumění a JPEG komprese originálního obrázku. Budeme předpokládat, že máme validní data a tedy platí $P(B) > 0$.
- $P(\{X_0 \rightarrow Y_{in}\} | \{X = X_0\}) = P(B|A)$ vyjadřuje pravděpodobnost jevu, kdy vstupní data vznikla z originálního obrázku popsaným způsobem (viz Obrázek 2.3) za podmínky, že je originální obraz roven X . V dalším textu ukážeme, že tuto pravděpodobnost lze spočítat.
- $P(\{X = X_0\}) = P(A)$ značí pravděpodobnost toho, že je originální obraz roven X . Tuto pravděpodobnost lze jen těžko určit. Vyjadřuje však naši představu o tom, jak má správný originální obrázek vypadat. Bude proto sloužit jako apriorní informace o původním obrázku. Více se jí budeme věnovat v sekci 2.4.

V následujícím textu budeme počítat pravděpodobnost

$$P(\{X_0 \rightarrow Y_{in}\} | \{X = X_0\}).$$

Pro získání základního pohledu na věc nejprve uvažujme jednoduchý případ.

Případ JPEG komprese bez šumu pro jeden DCT koeficient.

Nechť $y_{in} \in \mathbb{R}$ a $q \in \mathbb{N}$ jsou pevná čísla a je známo, že platí vztah

$$y_{in} = T(y_0/q),$$

kde $y_0 \in \mathbb{R}$ je neznámé. To odpovídá kvantizaci DCT koeficientu y_0 při JPEG kompresi. Definujme čísla y_{up}, y_{down} (toto označení budeme používat i v následujícím textu) následovně:

$$\begin{aligned} y_{up} &:= \sup \{y \in \mathbb{R} \mid Ty = y_{in}\} = y_{in} + 0.5, \\ y_{down} &:= \inf \{y \in \mathbb{R} \mid Ty = y_{in}\} = y_{in} - 0.5. \end{aligned}$$

Pak je zřejmé, že hledané y_0 leží v intervalu $(q y_{down}, q y_{up}]$.

Dále uvažujme následující situaci.

Případ JPEG komprese s šumem pro jeden DCT koeficient.

Nechť $y_{in} \in \mathbb{R}$ a $q \in \mathbb{N}$ jsou pevná čísla, $Z \in \mathbb{R}$ je náhodná veličina s rozdělením $N(0, \sigma^2)$ a je známo, že platí vztah

$$y_{in} = T((y_0 + Z)/q).$$

To odpovídá našemu modelu JPEG komprese zašuměného obrazu (viz Obrázek 2.3). Počítejme

$$\begin{aligned} P(\{y_{in} = T((y + Z)/q)\}) &= P(\{(y + Z)/q \leq y_{up}\}) - P(\{(y + Z)/q \leq y_{down}\}) \\ &= P(\{y + Z \leq qy_{up}\}) - P(\{y + Z \leq qy_{down}\}). \end{aligned}$$

Poznamenejme, že náhodná veličina $y + Z$ má rozdělení $N(y, \sigma^2)$. Z definice distribuční funkce a z tvaru distribuční funkce obecného normálního rozdělení (viz Definice 12 a 18) plynou vztahy

$$\begin{aligned} P(\{y + Z \leq qy_{up}\}) &= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{qy_{up}} \exp\left\{-\frac{(t - y)^2}{2\sigma^2}\right\} dt = \Phi\left(\frac{qy_{up} - y}{\sigma}\right), \\ P(\{y + Z \leq qy_{down}\}) &= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{qy_{down}} \exp\left\{-\frac{(t - y)^2}{2\sigma^2}\right\} dt = \Phi\left(\frac{qy_{down} - y}{\sigma}\right). \end{aligned}$$

Dohromady tedy máme

$$P(\{y_{in} = T((y + Z)/q)\}) = \Phi\left(\frac{qy_{up} - y}{\sigma}\right) - \Phi\left(\frac{qy_{down} - y}{\sigma}\right). \quad (2.3)$$

2.3.2 Formulace funkcionálu

Bude užitečné uvědomit si následující pozorování.

Pozorování. Budiž X_0, X obrazy a Y_0, Y jejich odpovídající DCT koeficienty, t. j.

$$\text{BDCT}(X_0) = Y_0, \quad \text{BDCT}(X) = Y.$$

Pak jevy $\{X = X_0\}$ a $\{Y = Y_0\}$ nastávají současně a pro jejich pravděpodobnosti platí

$$P(\{X = X_0\}) = P(\{Y = Y_0\}).$$

Stejné tvrzení samozřejmě platí i pro jednotlivé odpovídající prvky zúčastněných matic.

Pozorování. Nechť $A, B \in \mathbb{R}^{N \times M}$. Označme a^j, b^j prvky matice A , resp. B . Jev $\{A = B\}$ je ekvivalentní průniku jevů $\{a^j = b^j\}$ přes všechna $j = 1, \dots, NM$. Jinými slovy platí

$$\{A = B\} = \bigcap_j \{a^j = b^j\}.$$

Mějme dány kvantizované DCT koeficienty Y_{in} , kvantizační matici Q , parametr $\sigma > 0$ a nechť Z je náhodná veličina s rozdělením $N(0, \sigma^2)$. Podle předchozích pozorování počítejme

$$\begin{aligned} P(\{X_0 \rightarrow Y_{in}\} | \{X = X_0\}) &= P(\{X_0 \rightarrow Y_{in}\} | \{Y = Y_0\}) \\ &= P(\{Y_0 \rightarrow Y_{in}\} | \{Y = Y_0\}) \\ &= P(\{Y \rightarrow Y_{in}\}) \\ &= \prod_j P(\{y^j \rightarrow y_{in}^j\}) \\ &= \prod_j P(\{y_{in}^j = T((y^j + Z)/q^j)\}). \end{aligned}$$

Podle vztahu (2.3) pak dostáváme

$$P(\{X_0 \rightarrow Y_{in}\} | \{X = X_0\}) = \prod_j \left[\Phi\left(\frac{q^j y_{up}^j - y^j}{\sigma}\right) - \Phi\left(\frac{q^j y_{down}^j - y^j}{\sigma}\right) \right], \quad (2.4)$$

kde j prochází přes všechny obrazové body obrazu X a q^j jsou prvky po blocích rozšířené kvantizační matice.

Nyní můžeme podle Bayesovy věty (2.1) a výše uvedených výpočtů psát

$$P(\{X = X_0\} | \{X_0 \rightarrow Y_{in}\}) = \frac{P(\{X_0 \rightarrow Y_{in}\} | \{X = X_0\}) P(\{X = X_0\})}{P(\{X_0 \rightarrow Y_{in}\})}.$$

Obraz X_0 je pevný (ač neznámý) a koeficienty Y_{in} jsou dány. Pravděpodobnost $P(\{X_0 \rightarrow Y_{in}\})$ je tedy podle předpokladů kladná konstanta. Jejím zanedbáním dostáváme

$$P(\{X = X_0\} | \{X_0 \rightarrow Y_{in}\}) \propto P(\{X_0 \rightarrow Y_{in}\} | \{X = X_0\}) P(\{X = X_0\}),$$

kde \propto značí úměrnost. Podle (2.4) máme

$$P(\{X = X_0\} | \{X_0 \rightarrow Y_{in}\}) \propto \prod_j \left[\Phi\left(\frac{q^j y_{up}^j - y^j}{\sigma}\right) - \Phi\left(\frac{q^j y_{down}^j - y^j}{\sigma}\right) \right] P(\{X = X_0\}). \quad (2.5)$$

Nášim úkolem bude nalézt takový obraz X , pro který bude pravděpodobnost $P(\{X = X_0\} | \{X_0 \rightarrow Y_{in}\})$ pro dané DCT koeficienty Y_{in} , kvantizační tabulku Q a $\sigma > 0$ maximální. Maximalizovaná funkce (2.5) je ve tvaru součinu a proto bude výhodné namísto (2.5) maximalizovat logaritmus vztahu (2.5). Navíc, vlastností pravděpodobnosti je, že náleží do intervalu $[0, 1]$. Logaritmus čísla z tohoto intervalu je však menší nebo roven nule. Bude tedy vhodné minimalizovat opačnou hodnotu logaritmu uvedeného vztahu.

Označme

$$L(X, Y_{in}, Q) = - \sum_j \log \left[\Phi\left(\frac{q^j y_{up}^j - y^j}{\sigma}\right) - \Phi\left(\frac{q^j y_{down}^j - y^j}{\sigma}\right) \right], \quad (2.6)$$

kde y^j jsou DCT koeficienty obrazu X zpracovaného po blocích a q^j jsou kvantizační koeficienty po blocích rozšířené kvantizační matice Q . Ještě označme

$$\text{prior}(X) = - \log P(\{X = X_0\}).$$

Nášim úkolem pak bude nalézt takový obraz X , který bude minimalizovat funkcionál

$$P(X, Y_{in}, Q) = L(X, Y_{in}, Q) + \text{prior}(X). \quad (2.7)$$

Funkcionál L budeme nazývat **věrohodnostní člen** a $\text{prior}(X)$ budeme zvat **apriorní člen**.

2.4 Apriorní informace o obrazu

V této kapitole se budeme věnovat vhodné volbě apriorního členu $\text{prior}(X)$ funkcionálu (2.7). Existují různé přístupy k tomu, jak má vypadat „ideální“ obrázek. V článku Chambolle a Pock (2011) například používají model, kdy ideální obraz má malou totální variaci, definovanou pro obraz $X \in \mathbb{R}^{N \times N}$ následovně. Položme

$$\begin{aligned} (\nabla X)_{i,j}^1 &= \begin{cases} X_{i+1,j} - X_{i,j} & \text{pro } i < N, \\ 0 & \text{pro } i = N, \end{cases} \\ (\nabla X)_{i,j}^2 &= \begin{cases} X_{i,j+1} - X_{i,j} & \text{pro } j < N, \\ 0 & \text{pro } j = N. \end{cases} \end{aligned} \quad (2.8)$$

Pak totální variace obrazu X má tvar

$$\text{TV}(X) = \sum_{1 \leq i,j \leq N} |(\nabla X)_{i,j}|. \quad (2.9)$$

Snaha o minimální totální variaci má za následek již zmíněné „schodovité“ plochy ve výsledném obrázku (viz Obrázek 2.1). Tento nedostatek lze do jisté míry odstranit vhodnou úpravou tvaru totální variace, popsanou například v článku Alter a kol. (2004).

Další možností je použít model, kdy ideální obrázek minimalizuje funkcionál

$$\|\nabla X\|_2^2 = \sum_{1 \leq i,j \leq N} |(\nabla X)_{i,j}|^2,$$

kde ∇X je definována vztahem (2.8). Experimentálně jsme zjistili, tato volba vede k značnému rozmazání obrázku a není tedy vhodná.

Jiným přístupem je použít model, ve kterém má ideální obraz velký počet malých waveletových koeficientů. Tento způsob zvolili například autoři článku Nallaperumal a kol. (2006). Protože dává dobré výsledky, zapojili jsme komplexní wavelety do našeho modelu. Problematika waveletů je obšírně rozepsána například v klasické učebnici Najzar (2004). Nás bude zajímat, že dvourozměrná diskretní waveletová transformace obrazu $X \in \mathbb{R}^{N \times M}$ je transformací na jistou speciální bázi prostoru $l^2(\mathbb{Z}_N \times \mathbb{Z}_M)$. Koeficienty obrazu X ve waveletové bázi budeme nazývat waveletovými koeficienty. Značíme

$$\mathcal{W} : \mathbb{R}^{N \times M} \rightarrow \mathbb{R}^{N_1 \times M_1},$$

kde $N_1, N_2 \in \mathbb{N}$, $N_1 \geq N$, $M_1 \geq M$ (wavelety mohou být takzvaně přeurené). Waveletovou transformaci $\mathcal{W}(X)$ na vektoru $X \in \mathbb{R}^{NM}$ budeme chápat tak, jako by byla provedena na matici velikosti $N \times M$ a výsledná matice waveletových koeficientů velikosti $N_1 \times M_1$ byla opět transformována na sloupcový vektor. Podobně budeme chápat inverzní waveletovou transformaci \mathcal{W}^{-1} .

V našem modelu apriorní informace položíme

$$\text{prior}(X) = \alpha \|\mathcal{W}(X)\|_1 = \alpha \sum_i |(\mathcal{W}(X))_i|, \quad (2.10)$$

kde $\alpha > 0$ je vhodně volené číslo, které určuje, jakou váhu přikládáme apriornímu členu. Dále i ve vzorci (2.10) prochází všechny waveletové koeficienty obrazu X .

Typů waveletových transformací je celá řada. V článku Kingsbury (2001) autor představuje takzvané Dual-Tree CWT wavelety (z anglického Dual-Tree complex wavelet transform). Výhodou této waveletové transformace je dobrá směrová selektivita. Za \mathcal{W} ve vzorci (2.10) jsme v naší metodě zvolili transformaci Dual-Tree CWT.

Kapitola 3

Použité algoritmy

Standardním přístupem k uvedenému minimalizačnímu problému by bylo použít například metodu největšího spádu. Algoritmus metody největšího spádu pro diferencovatelnou funkci $F : \mathbb{R}^n \rightarrow \mathbb{R}$ má následující tvar.

Algoritmus 1: Metoda největšího spádu

- Inicializace: Zvolíme $\lambda^0 > 0$ a startovní hodnotu $z^0 \in \mathbb{R}^n$.
- Iterace: pro $k \geq 0$ upravujeme proměnné dle následujícího:

$$\begin{aligned} z^{k+1} &= z^k - \lambda^k \nabla F(z^k), \\ \lambda^{k+1} &\text{ zvolíme vhodné kladné číslo.} \end{aligned}$$

Ukážeme, že tento algoritmus není výhodné takto použít, nicméně je užitečné se jím zabývat.

3.1 Rozbor minimalizovaného funkcionálu

Připomeňme, že minimalizujeme funkcionál (2.7) tvaru

$$P(X, Y_{in}, Q) = L(X, Y_{in}, Q) + \text{prior}(X),$$

kde věrohodnostní člen má tvar

$$L(X, Y_{in}, Q) = - \sum_j \log \left[\Phi \left(\frac{q^j y_{up}^j - y^j}{\sigma} \right) - \Phi \left(\frac{q^j y_{down}^j - y^j}{\sigma} \right) \right]$$

(y^j jsou prvky matice $Y = \text{BDCT}(X)$) a apriorní člen je tvaru

$$\text{prior}(X) = \alpha \|\mathcal{W}_d(X)\|_1.$$

Nejprve spočteme, jak vypadá gradient funkcionálu $L(X, Y_0, Q)$ (Y_0 a Q jsou pevné, derivujeme podle X) v bodě X pro obraz velikosti 8×8 .

Poznámka. Je zřejmé, že je-li obraz X velikosti násobků osmi složený z bloků X_n (a stejně tak DCT koeficienty Y_{in} obrazu stejné velikosti), platí

$$L(X, Y_{in}, Q) = \sum_n L(X_n, Y_{in}, Q).$$

Funkcionál L je v jistém smyslu lokální a změna hodnot obrazu v jednom bloku neovlivní hodnotu funkcionálu počítaného na jiném bloku.

Podle předchozí poznámky stačí spočítat gradient funkcionálu L pro obraz velikosti 8×8 .

3.1.1 Derivace věrohodnostního členu funkcionálu

S obrazem $X \in \mathbb{R}^{8 \times 8}$ budeme pracovat jako s vektorem, s výjimkou situace, kdy budeme provádět 2D DCT transformaci – tehdy jej budeme považovat za matici. S výslednými DCT koeficienty obrazu X budeme zase pracovat jako s vektorem.

Mějme Y_{in} , Q a σ dané, zvolme pevně libovolné $j = 8j_1 + j_2$, $l = 8l_1 + l_2$, kde $j_1, j_2, l_1, l_2 \in \{0, \dots, 7\}$ a podle (2.6) položíme

$$K(X, j) = \log \left[\Phi \left(\frac{q^j y_{up}^j - (\text{DCT}(X))^j}{\sigma} \right) - \Phi \left(\frac{q^j y_{down}^j - (\text{DCT}(X))^j}{\sigma} \right) \right].$$

Poznamenejme, že korektnější by bylo namísto $K(X, j)$ psát $K(X, Y_{in}, Q, \sigma, j)$, to by však vedlo ke komplikovanému značení.

Budeme derivovat funkcionál $K(X, j)$ ve směru x^l . Podle definice diskrétní kosinové transformace pro blok velikosti 8×8 platí

$$Y(u, v) = \frac{1}{4} C(u) C(v) \left[\sum_{n=0}^7 \sum_{m=0}^7 X(n, m) \cos \frac{(2n+1)u\pi}{16} \cos \frac{(2m+1)v\pi}{16} \right],$$

kde $C(0) = 1/\sqrt{2}$ a $C(v) = 1$ pro $v \geq 1$. Budeme značit

$$\psi_{j,k} = \cos \frac{(2j_1+1)l_1\pi}{16} \cos \frac{(2j_2+1)l_2\pi}{16}.$$

Pak máme

$$\frac{\partial (\text{DCT}(X))^j}{\partial x^l} = \psi_{j,l}.$$

Dále platí

$$\Phi'(z) = \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{z^2}{2} \right\}. \quad (3.1)$$

Označme

$$A_{up}(X, j) := \left(\frac{q^j y_{up}^j - (\text{DCT}(X))^j}{\sigma} \right), \quad (A_{down}(X, j) \text{ analogicky}).$$

Pak platí vztah

$$\frac{\partial A_{up}(X, j)}{\partial x^l} = -\frac{1}{\sigma} \psi_{j,l}$$

a z řetízkového pravidla

$$\frac{\partial \Phi(A_{up}(X, j))}{\partial x^l} = \Phi'(A_{up}(X, j)) (-1) \frac{\partial A_{up}(X, j)}{\partial x^l} = \frac{1}{\sqrt{2\pi}\sigma^2} \psi_{j,l} \exp \left\{ -\frac{A_{up}^2(X, j)}{2} \right\}.$$

Vztahy obsahující A_{down} jsou analogické. Ještě označme

$$R(X, j) := \Phi(A_{up}(X, j)) - \Phi(A_{down}(X, j)).$$

Nyní již snadno spočteme, že

$$\frac{\partial K(X, j)}{\partial x^l} = \frac{1}{R(X, j)} \frac{\partial}{\partial x^l} R(X, j),$$

což díky předchozím vztahům dává

$$\frac{\partial K(X, j)}{\partial x^l} = \frac{1}{\sqrt{2\pi\sigma^2}} \psi_{j,l} \frac{1}{R(X, j)} \left(\exp \left\{ -\frac{A_{up}^2(X, j)}{2} \right\} - \exp \left\{ -\frac{A_{down}^2(X, j)}{2} \right\} \right).$$

Gradient funkcionálu $L(X, Y_{in}, Q)$ je pak součtem příspěvků od každého obrazového bodu j a má tvar

$$\left(\nabla L(X, Y_{in}, Q) \right)_l = - \sum_{j=0}^{63} \frac{\partial K(X, j)}{\partial x^l}. \quad (3.2)$$

Tento způsob výpočtu není příliš výhodný. Pokud bychom chtěli využít některou z minimalizačních metod využívajících i Hessovu matici funkcionálu, narazili bychom na velmi velkou paměťovou náročnost, nehledě na velké množství nutných operací k jeho výpočtu. Hessova matice pro obrázek velikosti $N \times M$ má totiž $N^2 M^2$ prvků a i přes zmíněnou nezávislost jednotlivých bloků není dostatečně řídká.

Ještě horší situace může nastat při výpočtu gradientu apriorního členu minimalizovaného funkcionálu, kdy gradient vůbec nemusí existovat. Proto je nutné hledat jiný, efektivnější způsob řešení problému minimalizace našeho funkcionálu.

3.1.2 Derivace věrohodnostního členu funkcionálu v prostoru DCT koeficientů

Předpokládejme na chvíli, že je apriorní člen nulový a minimalizujeme jenom funkcionál $L(X, Y_{in}, Q)$ vzhledem k $X \in \mathbb{R}^{8 \times 8}$ metodou největšího spádu. Pak v každém kroku počítáme výraz

$$X^{k+1} = X^k - \lambda \nabla L(X^k, Y_{in}, Q). \quad (3.3)$$

Obraz X^k lze reprezentovat jako matici DCT koeficientů Y^k . DCT transformace je lineární a platí tedy, že

$$\text{DCT}(X^{k+1}) = \text{DCT}(X^k) - \lambda \text{DCT}(\nabla L(X^k, Y_{in}, Q)).$$

Označme $\tilde{L}(Y, Y_{in}, Q) = L(\text{IDCT}(X), Y_{in}, Q)$. Funkcionál \tilde{L} je stejného tvaru jako (2.6).

Nyní můžeme psát

$$Y^{k+1} = Y^k - \lambda \text{DCT}(\nabla \tilde{L}(Y^k, Y_{in}, Q)). \quad (3.4)$$

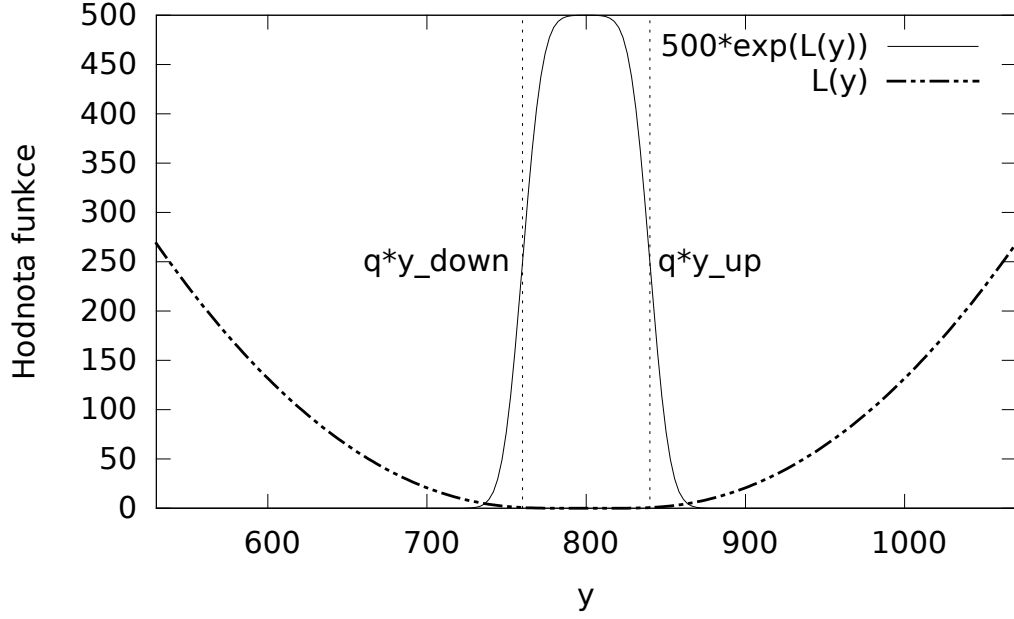
Zpětnou DCT transformací pak z Y^{k+1} dostaneme obázek X^{k+1} . V dalším textu ukážeme, že počítat výraz (3.4) namísto (3.3) je velmi výhodné.

Spočtíme gradient funkcionálu $\tilde{L}(Y, Y_{in}, Q)$ v proměnné Y . Máme

$$\tilde{L}(Y, Y_{in}, Q) = - \sum_j \log \left[\Phi \left(\frac{q^j y_{up}^j - y^j}{\sigma} \right) - \Phi \left(\frac{q^j y_{down}^j - y^j}{\sigma} \right) \right].$$

Položme

$$\tilde{K}(y, y_{in}, q, \sigma) := \log \left[\Phi \left(\frac{q y_{up} - y}{\sigma} \right) - \Phi \left(\frac{q y_{down} - y}{\sigma} \right) \right].$$



Obrázek 3.1: Ilustrace tvaru funkcionálu \tilde{L} v jedné dimenzi pro $y_{in} = 10$, $q = 80$, $\sigma = 10$. Plná čára značí 500x zvětšenou hodnotu odlogaritmovaného funkcionálu \tilde{L} a je úměrná pravděpodobnosti $P(\{y \rightarrow y_{in}\})$. Přerušovaná čára značí hodnotu funkcionálu \tilde{L} . Svislé přerušované čáry vymezují oblast, kde se mohou vyskytovat DCT koeficienty při běžné JPEG kompresi bez započítaného šumu (viz případ na straně 17).

Pak máme

$$\tilde{L}(Y, Y_{in}, Q) = - \sum_j \tilde{K}(y^j, y_{in}^j, q^j, \sigma)$$

a funkcionál \tilde{L} je tedy součtem příspěvků \tilde{K} , které závisejí jen na jednom z prvků matice Y . Díky tomu máme

$$\frac{\partial \tilde{L}}{\partial y^j}(Y, Y_{in}, Q) = - \frac{\partial \tilde{K}}{\partial y}(y^j, y_{in}^j, q^j, \sigma).$$

Označíme-li

$$\tilde{A}_{up}(y, y_{in}, q, \sigma) := \left(\frac{q y_{up} - y}{\sigma} \right), \quad (\tilde{A}_{down} \text{ analogicky})$$

a

$$\tilde{R}(y, y_{in}, q, \sigma) := \Phi(\tilde{A}_{up}(y, y_{in}, q, \sigma)) - \Phi(\tilde{A}_{down}(y, y_{in}, q, \sigma)), \quad (3.5)$$

pak stejně jako v sekci 3.1.1 použitím vztahu (3.1) spočteme, že platí

$$\frac{\partial \tilde{R}}{\partial y}(y, y_{in}, q, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \left(\exp \left\{ -\frac{\tilde{A}_{up}^2(y, y_{in}, q, \sigma)}{2} \right\} - \exp \left\{ -\frac{\tilde{A}_{down}^2(y, y_{in}, q, \sigma)}{2} \right\} \right). \quad (3.6)$$

Dále platí

$$\frac{\partial \tilde{K}}{\partial y}(y, y_{in}, q, \sigma) = \frac{1}{\tilde{R}(y, y_{in}, q, \sigma)} \frac{\partial \tilde{R}}{\partial y}(y, y_{in}, q, \sigma). \quad (3.7)$$

Podle předchozích výpočtů je tedy l -tá složka $\nabla \tilde{L}(Y, Y_{in}, Q)$ rovna výrazu

$$\frac{-1}{\tilde{R}(y^l, y_{in}^l, q^l, \sigma)} \left(\exp \left\{ -\frac{\tilde{A}_{up}^2(y^l, y_{in}^l, q^l, \sigma)}{2} \right\} - \exp \left\{ -\frac{\tilde{A}_{down}^2(y^l, y_{in}^l, q^l, \sigma)}{2} \right\} \right).$$

Tento výraz je v porovnání se vztahem (3.2) mnohem vstřícnější praktickým výpočtům. Obsahuje totiž jen l -tý prvek matic Y, Y_{in} . Navíc je důležité, že se minimalizace funkcionálu L převede na jednotlivé jednodimenzionální minimalizační problémy.

V jednodimenzionálním případě je také možné snáze odvodit konvexitu funkcionálu \tilde{L} . Lze ukázat, že výraz (3.7) je v proměnné y klesající a funkcionál \tilde{L} je jako součet konvexních funkcí konvexní.

Tvar minimalizovaného funkcionálu v jedné dimenzi ilustruje Obrázek 3.1.

3.1.3 Minimum věrohodnostního členu

Nyní ukážeme, že v našem přístupu je apriorní člen funkcionálu nezbytný. Minimum se totiž nabývá na obrázku odpovídajícímu kvantizovaným DCT koeficientům Y_{in} . Minimalizací věrohodnostního členu bychom tedy dostali pouze dekomprimovaný JPEG obrázek se všemi jeho obrazovými artefakty.

Tento fakt ověříme – počítejme podle (3.7)

$$\frac{\partial \tilde{K}}{\partial y}(qy_{in}, y_{in}, q, \sigma) = \frac{1}{\tilde{R}(qy_{in}, y_{in}, q, \sigma)} \frac{\partial \tilde{R}}{\partial y}(qy_{in}, y_{in}, q, \sigma).$$

lze snadno ověřit, že $\tilde{R}(t, y_{in}, q, \sigma) > 0$ pro všechna t na jistém prstencovém okolí qy_{in} . Dále

$$\tilde{A}_{up}(qy_{in}, y_{in}, q, \sigma) = \tilde{A}_{down}(qy_{in}, y_{in}, q, \sigma) = 0$$

a tedy podle (3.5) a (3.6) máme

$$R(qy_{in}, y_{in}, q, \sigma) = 0, \quad \frac{\partial \tilde{R}}{\partial y}(qy_{in}, y_{in}, q, \sigma) = 0.$$

Z uvedeného pak již plyne, obrázek X_{in} odpovídající kvantizovaným DCT koeficientům Y_{in} (lokálně) minimalizuje věrohodnostní člen. Obrázek X_{in} (výsledek běžné JPEG dekomprese) je tedy vhodným startovacím bodem našeho algoritmu. Navíc z toho vyplývá, že bez apriorní informace bychom se s pomocí našeho přístupu nedostali dále.

V následující kapitole zavedeme jiný způsob minimalizace funkcionálu, který umožní zapojení potřebného apriorního členu.

3.2 Proximální algoritmy

V této kapitole zavedeme proximální operátor a proximální algoritmy podle přehledového článku Parikh a Boyd (2013) a tyto pojmy použijeme ke konstrukci vhodného algoritmu pro řešení uvedené minimalizační úlohy.

3.2.1 Definice a vlastnosti

Definice 26. Necht $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ je taková funkce, jejíž epigraf

$$\text{epi} f = \{(x, t) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) \leq t\}$$

je uzavřený, neprázdný a konvexní. O takové funkci řekneme, že se jedná o CPC funkci (z anglického closed proper convex function).

Poznámka. To, že funkce $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ může nabývat i nekonečné hodnoty lze využít k zahrnutí oblasti, na které funkci minimalizujeme, přímo do tvaru minimalizované funkce f .

Definice 27. Proximální operátor $\text{prox}_f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ funkce f definujeme předpisem

$$\text{prox}_f(v) := \underset{x \in \mathbb{R}^n}{\text{argmin}} \left(f(x) + \frac{1}{2} \|x - v\|_2^2 \right) \quad (3.8)$$

kde $\|\cdot\|_2$ značí eukleidovskou normu.

Pro konvexní funkci f je proximální operátor definován korektně (viz Rockafellar (1997)). Budeme také pracovat s proximálním operátorem funkce f škálované parametrem $\lambda > 0$. Zavedeme jej následovně:

$$\text{prox}_{\lambda f}(v) := \underset{x \in \mathbb{R}^n}{\text{argmin}} \left(f(x) + \frac{1}{2\lambda} \|x - v\|_2^2 \right). \quad (3.9)$$

V určitých situacích je snazší minimalizovat namísto daného funkcionálu odpovídající proximální operátor. Minimalizovaná funkce v definici proximálního operátoru je v jistém smyslu „konvexnější“ než původní funkcionál. Pro některé speciální funkce dokonce existuje explicitní vzorec pro proximální operátor.

Nyní uvedeme několik standardních algoritmů využívajících proximálních operátorů pro řešení úloh konvexní optimalizace.

3.2.2 Algoritmus proximální minimalizace

Pro CPC funkci $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ a počáteční bod $z^0 \in \mathbb{R}^n$ je algoritmus (v literatuře též známý jako proximal minimization algorithm) definován následovně:

$$z^{k+1} := \text{prox}_{\lambda^k f}(z^k), \quad (3.10)$$

kde $\lambda^k > 0$ je velikost kroku. O tomto algoritmu je v knize Bauschke a Combettes (2011) dokázáno, že jestliže má funkce f minimum a λ^k jsou voleny tak, aby platilo $\sum_{k=1}^{\infty} \lambda^k = \infty$, pak algoritmus konverguje k bodu minima funkce f . Tento algoritmus sice nepoužijeme, ale je dobrým příkladem nejzákladnější metody využití proximálního operátoru.

Další uvedený algoritmus již bude praktičtější.

3.2.3 Metoda proximálního gradientu

Tento algoritmus (v literatuře uváděný jako proximal gradient method) je navržen pro řešení úlohy

$$\text{minimalizujte } f(x) + g(x),$$

kde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ a $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ jsou CPC funkce a f je navíc diferencovatelná. Mějme $z^0 \in \mathbb{R}^n$. **Metoda proximálního gradientu** je definována předpisem

$$z^{k+1} := \mathbf{prox}_{\lambda^k g}(z^k - \lambda^k \nabla f(z^k)),$$

kde $\lambda^k > 0$ je velikost kroku.

Pro funkci f takovou, že ∇f je L -lipschitzovsky spojitý, a pro pevně zvolené $\lambda^k \in (0, 1/L]$ lze v článku Combettes a Pesquet (2011) nalézt důkaz konvergence uvedeného algoritmu.

Poznámka (Speciální případy algoritmu). Jestliže máme $f = 0$, algoritmus se redukuje na algoritmus proximální minimalizace. Je-li naopak $g = 0$, pak tento algoritmus není nic jiného, než metoda největšího spádu.

Tento algoritmus je již podstatně vhodnější pro řešení našeho úkolu. Připomeňme, že minimalizujeme funkcionál (2.7) tvaru

$$P(X, Y_{in}, Q) = L(X, Y_{in}, Q) + \text{prior}(X),$$

tedy součet dvou funkcí, z nichž jedna je – jak jsme ukázali na začátku této kapitoly – diferencovatelná. Uvedme ještě modifikaci tohoto algoritmu.

3.2.4 Akcelerovaná metoda proximálního gradientu (APGM)

Stejně jako v předchozím algoritmu minimalizujeme součet

$$f(x) + g(x),$$

přičemž funkce f , g jsou stejné, jako v sekci 3.2.3. Algoritmus definujeme následovně.

Algoritmus 2: Akcelerovaná metoda proximálního gradientu

- Inicializace: Zvolíme $\lambda^0 > 0$ a startovní hodnoty $z^0 \in \mathbb{R}^n$ a $z^{-1} = z^0$.
- Iterace: pro $k \geq 0$ upravujeme proměnné dle následujícího:

$$\begin{aligned} s^{k+1} &:= z^k + \omega^k(z^k - z^{k-1}), \\ z^{k+1} &:= \mathbf{prox}_{\lambda^k g}(s^{k+1} - \lambda^k \nabla f(s^{k+1})). \end{aligned} \quad (3.11)$$

Volbu vhodných λ^k probereme v sekci 4.1.4 věnované implementaci. V akcelerované verzi předchozího algoritmu navíc volíme extrapoláčnı parametr $\omega^k \in [0, 1)$. V naší metodě položíme

$$\omega^k = \frac{k}{k+3},$$

jak je doporučeno v přehledovém článku Parikh a Boyd (2013).

Akcelerovanou metodu proximálního gradientu použijeme pro funkce

$$f(X) = L(X, Y_{in}, Q), \quad g(X) = \text{prior}(X). \quad (3.12)$$

Uvedený algoritmus přirozeně rozděluje problém minimalizace $f(x) + g(x)$ na dvě podúlohy, a to

- spočtení gradientu funkce f v bodě s^{k+1} a určení bodu

$$d^{k+1} = s^{k+1} - \lambda^k \nabla f(s^{k+1}).$$

- vyčíslení proximálního operátoru v bodě d^{k+1} .

První bod jsme již pro volbu (3.12) vyřešili v sekci 3.1.2. Zbývá zajistit vyčíslení proximálního operátoru. Tomu se budeme věnovat v následující kapitole.

3.3 Vyčíslení proximálního operátoru

Mějme danou funkci $g : \mathbb{R}^n \rightarrow \mathbb{R}$, parametr $\lambda > 0$ a argument proximálního operátoru $d \in \mathbb{R}^n$ ve tvaru sloupcového vektoru. S obrazem $X \in \mathbb{R}^{N \times M}$ budeme pracovat jako se sloupcovým vektorem délky n .

Dle definice lze hodnotu proximálního operátoru $\text{prox}_{\lambda g}(d)$ zapsat jako

$$\text{argmin}_{z \in \mathbb{R}^n} \left(g(z) + \frac{1}{2\lambda} \|z - d\|_2^2 \right).$$

V našem případě ($g(X) = \|\mathcal{W}(X)\|_1$) minimalizujeme funkci

$$\|\mathcal{W}(X)\|_1 + \frac{1}{2\lambda} \|X - d\|_2^2 \quad (3.13)$$

vzhledem k proměnné X .

V článku (Chambolle a Pock, 2011, strana 4 a 24) je k nalezení algoritmus pro řešení podobné úlohy

$$\text{TV}(X) + \frac{1}{2\lambda} \|X - d\|_2^2,$$

kde TV je totální variace definovaná vztahem (2.9). Pro tento případ lze v uvedeném zdroji nalézt i důkaz konvergence.

Algoritmus z článku Chambolle a Pock (2011) jsme upravili pro řešení našeho problému (3.13) do následujícího tvaru.

Algoritmus 3: Vyčíslení proximálního operátoru

- Inicializace: Zvolíme konstanty $\tau, \rho > 0, \theta \in [0, 1]$, startovní hodnoty $\bar{X} \in \mathbb{R}^n$, $z^0 \in \mathbb{R}^n$ a pro vhodný rozměr m (tak, aby $\mathcal{W}(\bar{X}) \in \mathbb{R}^m$) vektor $w^0 \in \mathbb{R}^m$.
- Iterace: pro $k \geq 0$ upravujeme proměnné dle následujícího

$$\begin{aligned} v &= w^k + \rho \mathcal{W}(\bar{X}), \\ w_i^{k+1} &= v_i / \max(1, |v_i|) \quad (\text{pro všechna } i = 1, \dots, m), \\ v &= z^k - \tau \mathcal{W}^{-1}(w^{k+1}), \\ z^{k+1} &= (v + \tau \lambda d) / (1 + \tau \lambda), \\ \bar{X} &= z^{k+1} + \theta(z^{k+1} - z^k). \end{aligned} \quad (3.14)$$

Pro tento algoritmus nám není znám důkaz konvergence, nicméně numerické experimenty ukazují, že pro vhodnou volbu parametrů tento algoritmus konverguje.

Volbě počátečních hodnot a parametrů algoritmu (3.14) se budeme věnovat v sekci 4.1.5.

Kapitola 4

Popis implementace

4.1 Obecné informace

Implementaci naší metody odstraňování JPEG artefaktů jsme provedli v prostředí MATLAB, běžícím na linuxovém operačním systému Gentoo.

Implementační omezení. Metoda je naprogramována pouze pro černobílé JPEG obrázky, jejichž rozměry jsou násobky osmi.

4.1.1 Použité externí podprogramy

V našem programu jsme využili několik externích podprogramů, a to

- matlabovskou implementaci komplexní 2D waveletové transformace, autoři Cian Shaffrey, Nick Kingsbury, Cambridge University, Srpen 2000,
- balík JPEG Toolbox, verze 1.4, autor Phil Sallee, červen 2003,
- balík ImageMagick[®], viz <http://www.imagemagick.org/>,
- matlabovskou implementaci SSIM Index, verze 1.0, autor Zhou Wang, Institute of Mathematical Sciences, New York University, 2003.

4.1.2 Přehled procedur programu

V následujícím textu uvedeme nejdůležitější procedury, které v našem programu slouží k odstraňování JPEG artefaktů.

- **steep_and_denoise** je hlavní procedura, která provede celou navrženou metodu. Jejími parametry jsou vstupní obraz X_{in} , matice Y_{down} a Y_{up} , regularizační parametr α , parametr šumu σ , maximální počet vyčíslení proximálního operátoru, jméno souboru k případnému uložení informací o konvergenci, obraz k porovnání a tzv. cell se jmény typů měření chyby (SSIM/EPP).
- **nacti_jpeg** je podprogram určený k získání vstupních dat pro proceduru **steep_and_denoise**. Jejími parametry jsou jméno JPEG souboru a jméno souboru obrázku k případnému porovnání.

- **proximal_step** je procedura, která provádí jednotlivé kroky akcelerované metody proximálního gradientu (3.11). Vstupními parametry jsou: pořadí aktuálního kroku, z^k , z^{k-1} , λ^k , matice Y_{down} a Y_{up} , parametr šumu σ , regularizační parametr α a jméno souboru k případnému uložení dat o konvergenci.
- **spocti_steepdesc** zajišťuje spočtení výrazu $d^{k+1} = s^{k+1} - \lambda^k \nabla f(s^{k+1})$ podle sekce 3.1.2. Parametry metody jsou s^{k+1} , Y_{down} a Y_{up} , σ a λ^k .
- **do_minim_cwt** se stará o vyčíslení proximálního operátoru podle algoritmu (3.14). Jejími parametry jsou obrázek d , počet maximálních iterací algoritmu a čísla λ , τ , ρ a θ .

Uvedeme také nejdůležitější z procedur, které jsme napsali pro účely testování našeho programu.

- **komplet_find_optimal_alpha** je určena k nalezení optimálního regularizačního parametru α (nejlépe v situaci, kdy máme k dispozici původní obrázek).
- **testuj_vsechny_kvality** je procedura určená k testování naší metody na různých zadaných kvalitách JPEG obrázku pro daný originální obrázek.
- **testuj_vsechny_sumy_kvality** je podprogram k testování naší metody pro různé zadané parametry šumu σ a kvality JPEG obrázku při daném originálním obrázku.

4.1.3 Volba globálních parametrů

Těmito parametry rozumíme regularizační parametr α a parametr šumu σ . Ty ovlivňují celý proces odstraňování JPEG artefaktů v naší metodě.

Numerickými experimenty jsme zjistili, že algoritmus dává nejlepší výsledky pro σ v okolí čísla 20, nehledě na skutečné zašumění obrázku před JPEG kompresí. Zvolili jsme proto

$$\sigma = 20.$$

Poznamenejme, že pro velké hodnoty σ se vytrácí vliv věrohodnostního členu a úloha přechází v minimalizaci apriorního členu. Naopak pro $\sigma \rightarrow 0+$ dochází k numerické nestabilitě. To je způsobeno tvarem funkcionálu

$$\tilde{L}(Y, Y_{in}, Q) = - \sum_j \log \left[\Phi \left(\frac{q^j y_{up}^j - y^j}{\sigma} \right) - \Phi \left(\frac{q^j y_{down}^j - y^j}{\sigma} \right) \right].$$

Pro hodnoty y^j příliš vzdálené od intervalu $[q^j y_{down}^j, q^j y_{up}^j]$ má totiž distribuční funkce normálního rozdělení Φ v bodech $\frac{q^j y_{up}^j - y^j}{\sigma}$ a $\frac{q^j y_{down}^j - y^j}{\sigma}$ podobné hodnoty a dochází k odečtení dvou blízkých čísel a tedy ke ztrátě přesnosti. Čím menší je parametr σ , tím méně se pro vznik popsané nestability stačí odchýlit od zmíněného intervalu.

Existence tohoto problému také odpovídá tomu, že pro $\sigma \rightarrow 0+$ přechází úloha v obtížnější úlohu nelineárního programování (v anglicky psané literatuře „constrained optimization“). Formulace v řeči nelineárního programování pak zní:

Minimalizujte funkcionál $\text{prior}(X)$ na množině takových obrázků $X \in \mathbb{R}^{N \times M}$, pro jejichž DCT koeficienty platí $y^j \in [q^j y_{down}^j, q^j y_{up}^j]$ pro každé $j = 1, \dots, NM$.

Informace o způsobech řešení tohoto problému lze mimo jiné nalézt v knize Boyd a Vandenberghe (2004).

4.1.4 Implementace APGM

Algoritmus APGM (viz str. 28) jsme implementovali jako cyklické volání procedury `proximal_step`. Parametr algoritmu λ^k jsme volili následovně.

$$\lambda^k := \frac{\lambda^0}{(k+1)^{0.8}}, \quad \lambda^0 = 400,$$

což znamená postupné snižování kroku v jednotlivých iteracích APGM. Jako startovní hodnotu z^0 jsme volili vstupní JPEG obrázek X_{in} . Tuto volbu jsme již obhájili v sekci 3.1.3.

Závislost na parametrech. Volba příliš malého λ^0 vede k příliš pomalé konvergenci. Naopak volba příliš velkého λ^0 vede k numerické nestabilitě vysvětlené v sekci 4.1.3. Příliš velký parametr λ totiž může dovolit proximálnímu operátoru posunout zpracovávaný obrázek do nestabilní oblasti.

Zastavovací kritérium. Pro algoritmus APGM jsme implementovali zastavovací kritérium následovně:

Jestliže X^k, X^{k+1} jsou obrázky v k -tém a $(k+1)$ -ním kroku a platí

$$P(X^{k+1}, Y_{in}, Q) > P(X^k, Y_{in}, Q),$$

pak zamítneme $(k+1)$ -ní iteraci a zastavíme výpočet. (Připomeňme, že funkcionál P je definován vztahem (2.7).) Toto zastavovací kritérium je v kódu pro potřeby testování vypnuto a pro rychlejší běh metody jej lze zapnout.

Pokud není splněna podmínka zastavovacího kritéria, provede se výpočet pro krok $k = 0$ až MAXITER.

4.1.5 Implementace vyčíslení proximálního operátoru

Vyčíslení proximálního operátoru obstarává procedura `do_minim_cwt`, implementovaná podle algoritmu (3.14). Jako startovní hodnoty algoritmu jsme volili $\bar{X} = z^0 = d$ a $w^0 = 0$.

Závislost na parametrech. Parametry algoritmu jsme volili takto.

$$\tau = 1, \quad \rho = 1, \quad \theta = 1.$$

V našich numerických experimentech jsme zjistili, že algoritmus (3.14) není příliš citlivý na volbu těchto parametrů.

Zastavovací kritérium. Nechť X^k, X^{k+1} jsou obrázky v k -tém a $(k+1)$. kroku vyčíslení proximálního operátoru. Jako zastavovací kritérium jsme v naší implementaci nastavili podmínku

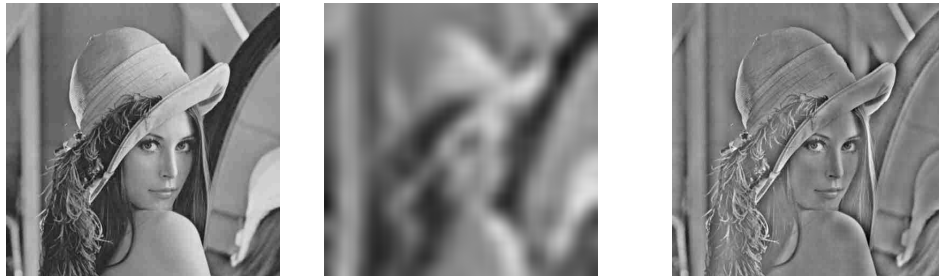
$$\left(\|\mathcal{W}(X^{k+1})\|_1 - \|\mathcal{W}(X^k)\|_1\right)^2 + \frac{1}{2\lambda} \left(\|X^{k+1} - d\|_2^2 - \|X^k - d\|_2^2\right)^2 < N^2 M^2 \text{TOL},$$

kde $M \times N$ jsou rozměry zpracovávaného obrázku. Podmínka vyjadřuje situaci, kdy se již minimalizovaný funkcionál mění příliš pomalu na to, aby se vyplatilo počítat další kroky. V našich experimentech jsme volili $\text{TOL} = 10^{-3}$. Toto zastavovací kritérium lze v kódu taktéž vypnout.

Pokud výpočet hodnoty proximálního operátoru neskončí splněním zastavovacího kritéria, provede se NUMIT+1 kroků algoritmu. V našem programu jsme použili volbu $\text{NUMIT} = 30$.

Nastavení waveletové transformace. Použitá waveletová transformace Dual-Tree CWT má ve své implementaci několik volitelných parametrů. Zvolili jsme následující sadu parametrů: `near_sym_a`, t. j. „Near-Symmetric 5,7 tap filters“, `qshift_a`, t. j. „Q-shift 10,10 tap filters“ při 5. úrovni waveletové transformace.

Zlepšení vlastností algoritmu. Ukázalo se, že je vhodné pozměnit tvar funkcionálu $\text{prior}(X)$. Pro funkcionál tvaru (2.10) totiž dochází k snižování kontrastu obrazu z důvodu minimalizace všech waveletových koeficientů. Proto v naší implementaci minimalizujeme pouze ty waveletové koeficienty, které připadají prostoru detailů (viz Najzar (2004)). Rozklad obrázku na „coarse“ složku a složku detailů ilustruje Obrázek 4.1.



JPEG obraz ($q = 20$). „Coarse“ složka. Waveletová složka detailů.

Obrázek 4.1: Pohled na „coarse“ složku a složku detailů Dual-Tree CWT waveletu s parametry `near_sym_a` a `qshift_a` hloubky 5. Minimalizujeme koeficienty obrázku vpravo.

Kapitola 5

Výsledky

5.1 Sada testovacích obrázků

Pro testování navržené metody použijeme testovací obrázky lena, barbara, ulice a město (viz Obrázek 5.1).



lena



barbara



ulice



město

Obrázek 5.1: Testovací obrázky.

Obrázky lena a barbara jsou standardně používané obrázky pro testování metod zpracování obrazu. Obrázky ulice a město jsou de facto běžné fotografie, byly však vytvořeny takovým způsobem, aby neobsahovaly šum. V Tabulce 5.1 nalezneme srovnání velikostí uvedených obrázků a jejich JPEG kompresí.

| Obrázek | Rozlišení | Velikost (kB) | | | | | | |
|---------|------------------|---------------|---------|----------|----------|----------|----------|----------|
| | | png | $q = 5$ | $q = 10$ | $q = 20$ | $q = 30$ | $q = 40$ | $q = 50$ |
| lena | 512×512 | 232.3 | 5.4 | 9.6 | 16.8 | 22.5 | 27.1 | 31.3 |
| barbara | 512×512 | 151.2 | 3.7 | 6.5 | 10.8 | 14.3 | 17.4 | 20.3 |
| ulice | 648×432 | 176.2 | 3.8 | 7.1 | 12.4 | 17.0 | 20.8 | 24.3 |
| praha | 648×432 | 205.7 | 3.7 | 7.5 | 13.9 | 19.4 | 23.9 | 28.2 |

Tabulka 5.1: Velikost testovacích obrázků ve formátu PNG a jejich JPEG kompresí.

5.2 Návrh a výsledky numerických experimentů

Nejprve krátce probereme použité způsoby porovnání obrázků.

5.2.1 Měření chyby vůči testovacím obrázkům

K porovnání obrázků použijeme následující metody.

Structural similarity (SSIM)

Index SSIM vyjadřuje míru podobnosti dvou obrázků. Byl navržen tak, aby rozdíl mezi obrázky zachycoval podobně, jako jej vnímá lidské oko. Na výřezích $x, y \in \mathbb{R}^{N \times N}$ transformovaných do sloupcových vektorů je SSIM index je definován následovně:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (5.1)$$

kde μ_x, μ_y jsou průměrné hodnoty x, y , σ_x^2, σ_y^2 jsou rozptyly x, y a dále

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(x_i - \mu_y),$$

$c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$, kde $k_1 = 0.01$, $k_2 = 0.03$ a L je dynamický rozsah obrázku, v našem případě $L = 255$.

SSIM nabývá hodnot -1 až 1 a platí $\text{SSIM}(x, x) = 1$. Pro celý obraz se SSIM počítá jako průměr SSIM přes jednotlivé výřezy. Obvyklá používaná velikost výřezu je $N = 8$. Podrobnější informace o SSIM jsou k nalezení v článku Wang a kol. (2004).

Průměrná chyba na pixel (EPP)

Průměrnou chybu na pixel obrázku $x \in \mathbb{R}^{N \times M}$ vůči $y \in \mathbb{R}^{N \times M}$ spočteme následovně:

$$\text{EPP}(x, y) = \frac{1}{NM} \left(\sum_{i=1}^N \sum_{j=1}^M (x_{i,j} - y_{i,j})^2 \right)^{1/2}. \quad (5.2)$$

Průměrnou chybu na pixel lze snáze interpretovat, než SSIM – udává, jak daleko jsou jednotlivé pixely průměrně odchýleny od originálního obrázku.

Vždy, když budeme mluvit o chybě obrázku, budeme tím rozumět porovnání obrázku s odpovídajícím testovacím obrázkem.

Definice 28. Pro snadnou interpretaci výsledků naší metody zavedeme ukazatel **relativního zlepšení EPP** obrázku vztahem

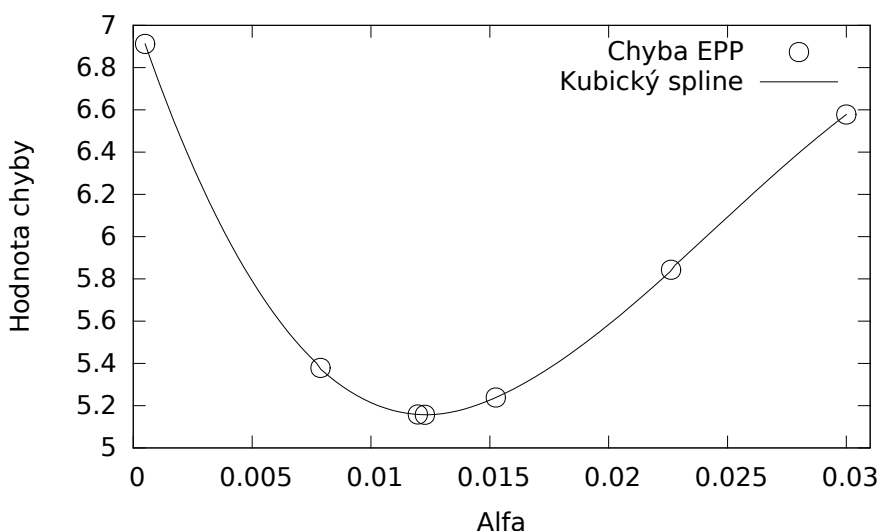
$$\text{REPP} = 1 - \frac{\text{chyba výsledného obrázku (EPP)}}{\text{chyba vstupního JPEG obrázku (EPP)}}$$

a ukazatel **relativního zlepšení SSIM** obrázku

$$\text{RSSIM} = 1 - \frac{1 - \text{chyba výsledného obrázku (SSIM)}}{1 - \text{chyba vstupního JPEG obrázku (SSIM)}}.$$

Jak je vidět, porovnáním chyby dvou identických obrázků dostaneme pro SSIM i EPP nulové relativní zlepšení. Dále, stoprocentní relativní zlepšení pro EPP, respektive pro SSIM, dostaneme v případě, kdy je chyba výsledného obrázku EPP nulová, resp. hodnota chyby výsledného obrázku SSIM je rovna jedné.

5.2.2 Hledání optimálního parametru alfa



Obrázek 5.2: Hledání optimálního parametru α . Graf ukazuje závislost chyby EPP na volbě regularizačního parametru α . Konkrétní hodnoty: Obrázek město, $\sigma = 10$, $q = 40$, MAXITER = 10.

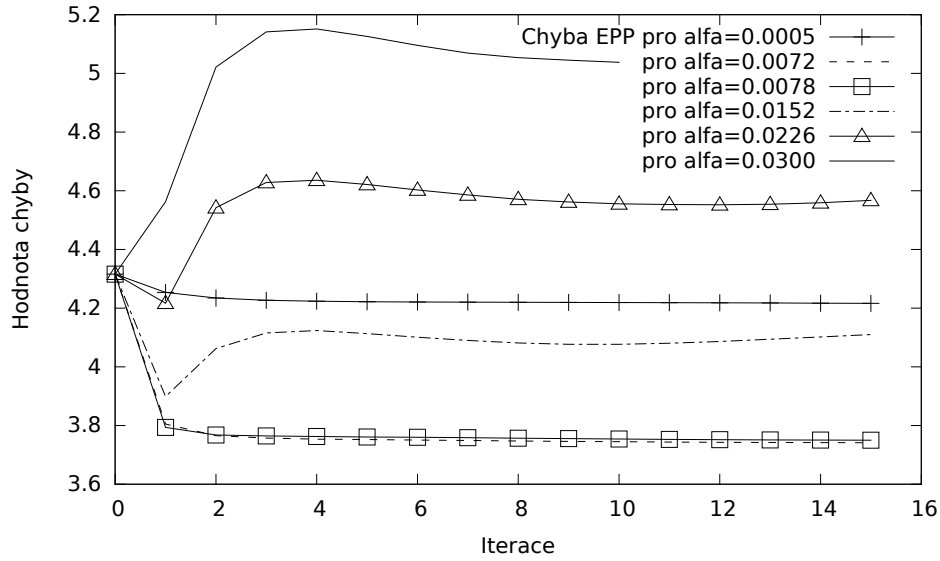
Pro naše experimenty bude užitečné najít způsob, jak automaticky nalézt optimální parametr α , pro které naše metoda dává nejlepší výsledky. Díky tomu, že známe originální obrázek, můžeme změřit chybu výsledných obrázků pro jednotlivá α a určit nejlepší volbu.

Konkrétně v našich experimentech rovnoměrně rozdělíme vhodně zvolený interval na 5 bodů a pro hodnoty α na dělicích bodech provedeme naši metodu. Poté metodami SSIM a EPP změříme chybu výsledku a získáme jistou informaci

o závislosti chyby na zvoleném α . Tyto data proložíme kubickým splinem (ukazuje se, že závislost chyby na parametru α má průběh nezpůsobující oscilace, viz graf na Obrázku 5.2) a nalezneme bod minima (resp. maxima u SSIM).

Parametr α odpovídající nalezenému extrému kubického splinu prohlásíme za optimální hodnotu α pro daný obrázek. Pro získání optimálního výsledku pak naši metodu zavoláme pro nalezené optimální α . V naší implementaci popsany postup provádí funkce `komplet_find_optimal_alpha`.

Ověřit správnost výběru optimálního α můžeme jednoduše. Stačí porovnat chybu výsledků počítaných pro jednotlivá α . Chyba pro optimální volbu α by měla být lepší či velmi blízká chybě nejlepšího ze zbývajících výsledků. Situaci ilustruje graf na Obrázku 5.3.



Obrázek 5.3: Průběh APGM pro různé volby parametru α . Konkrétní hodnoty: Testovací obrázek lena, rozptyl šumu = 5, $q = 20$.

Pro snazší případné použití naší metody uvádíme tabulku spočtených optimálních α v chybě EPP pro různé kvality JPEG komprese nezašuměných obrázků a pro různé volby maximálního počtu iterací (viz Tabulka 5.2). Tyto hodnoty byly spočteny jako průměr hodnot optimálních α přes celou sadu našich testovacích obrázků. Tato čísla lze použít jako dobré počáteční odhady optimálního parametru α .

5.2.3 Konvergenční chování algoritmů

Konvergence algoritmu APGM

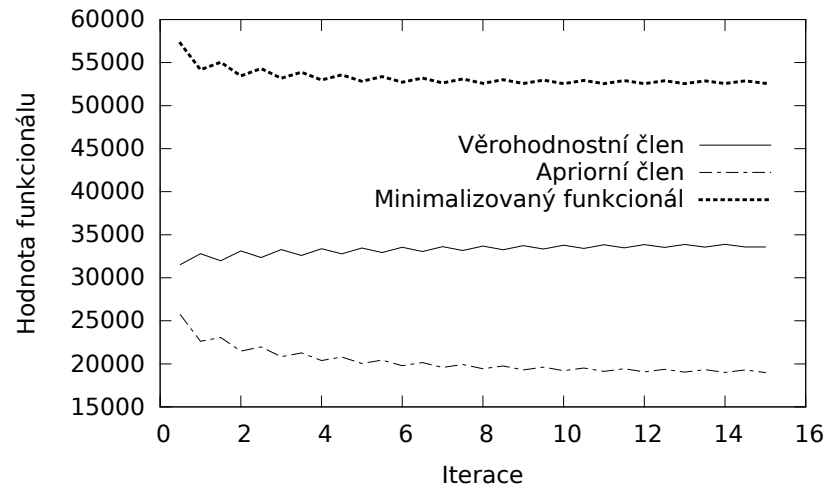
Typický vývoj funkcionálu (viz (2.7), str. 19)

$$P(X, Y_{in}, Q) = L(X, Y_{in}, Q) + \text{prior}(X).$$

v průběhu algoritmu APGM (3.11) ukazuje graf na Obrázku 5.4. Připomeňme, že v algoritmu APGM se střídá sestup po směru gradientu („půltá“ iterace) s vyčíslením proximálního operátoru („celá“ iterace).

| Optimální volba α pro EPP ($\times 10^3$) | | | | | |
|--|-------|-------|------|------|------|
| Max. počet iterací | | | | | |
| Kvalita | 1 | 3 | 5 | 10 | 15 |
| 5 | 27.10 | 11.15 | 8.32 | 6.76 | 6.01 |
| 10 | 14.06 | 7.88 | 6.76 | 6.31 | 6.09 |
| 20 | 8.10 | 6.01 | 5.86 | 5.79 | 5.57 |
| 30 | 6.31 | 5.12 | 5.12 | 5.19 | 4.90 |
| 40 | 5.12 | 4.37 | 4.37 | 4.52 | 4.30 |
| 50 | 4.45 | 3.78 | 3.93 | 3.93 | 3.93 |

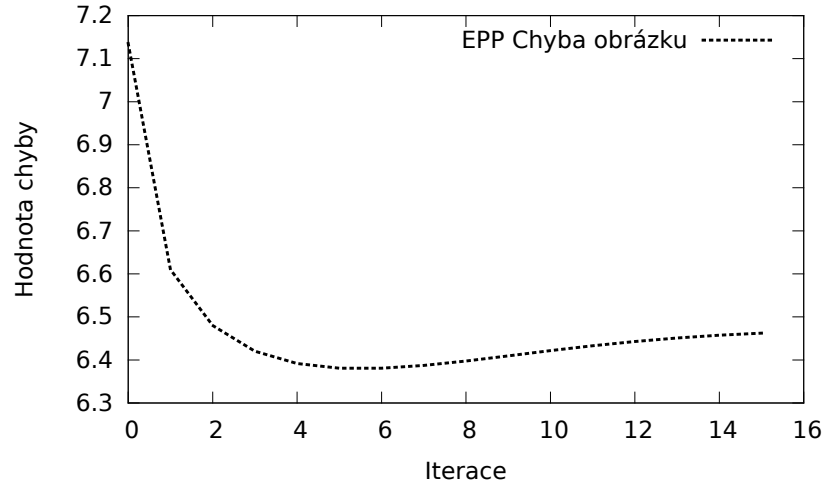
Tabulka 5.2: Tabulka průměrné optimální volby regularizačního parametru α spočtené pomocí chyby EPP pro různé JPEG kvality nezašuměného obrázku a různý počet provedených iterací algoritmu APGM.



Obrázek 5.4: Typický vývoj hodnoty funkcionálu (2.7) v průběhu algoritmu APGM. Konkrétní hodnoty: Testovací obrázek barbara, rozptyl šumu = 5, $q = 20$, $\alpha = 0,0057$.

Vidíme, že proximální člen klesá na úkor zvyšování věrohodnostního členu a naopak. Operace sestupu po směru gradientu a vyčíslení proximálního operátoru jdou tedy v jistém smyslu proti sobě. Ve výsledku se však hodnota funkcionálu zmenšuje.

Během jednotlivých kroků algoritmu můžeme zároveň sledovat vývoj chyby vůči testovacímu obrázku (viz Obrázek 5.5). Do páté iterace algoritmu klesá a pro další iterace již chyba začne mírně růst. Toto chování je pro naši metodu typické, jak uvidíme v souhrných výsledcích v sekci 5.2.4. Pravděpodobně je způsobeno tím, že apriorní člen nepopisuje dostatečně přesně ideální obrázek. Nárůst chyby je však velmi malý v porovnání s dosaženým poklesem chyby. Obrázky pro pátou a vyšší iteraci jsou navíc okem takřka nerozpoznatelné.



Obrázek 5.5: Typický vývoj EPP chyby obrázku v průběhu algoritmu APGM. Konkrétní hodnoty: Testovací obrázek barbara, rozptyl šumu = 5, $q = 20$, $\alpha = 0,0057$.

Konvergence vyčíslení proximálního operátoru

V grafu na Obrázku 5.6 je znázorněn typický vývoj funkcionálu (3.13)

$$\|\mathcal{W}(X)\|_1 + \frac{1}{2\lambda}\|X - d\|_2^2$$

v průběhu vyčíslování proximálního operátoru podle algoritmu (3.14).

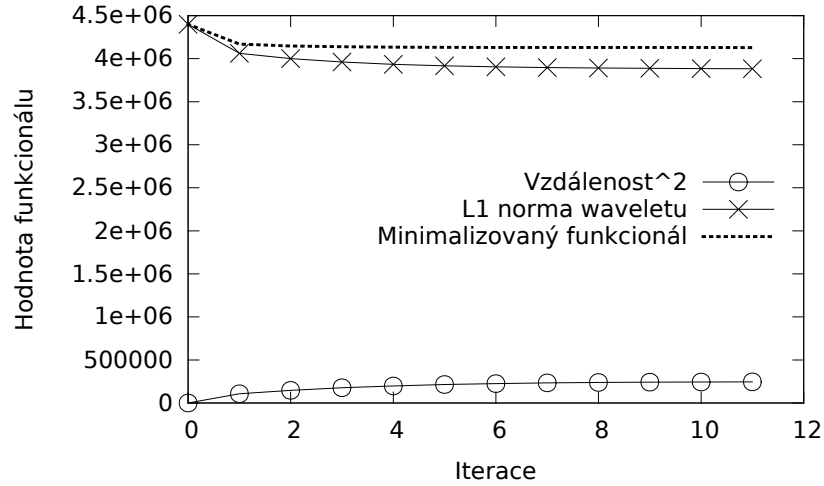
Počet provedených kroků algoritmu (3.11) pro vyčíslení proximálního operátoru při našem zastavovacím kritériu závisí na velikosti regularizačního parametru α a na pořadí kroku v algoritmu APGM. V prvním kroku APGM se počet provedených kroků algoritmu (3.11) pro spočtené optimální hodnoty α pohybuje kolem 13-ti. Pro druhý a další krok APGM již je potřeba kroků méně, přibližně 8 až 4.

5.2.4 Vlastnosti metody pro nezašuměné obrázky

Nejvíce nás zajímá chování naší metody pro běžné nezašuměné obrázky.

Redukce chyby JPEG obrázku

Nyní získáme představu o hodnotách chyby JPEG obrázku pro různé kvality komprese a o tom, jak tuto chybu zmenšuje naše metoda. Spočítali jsme EPP a SSIM chybu nezašuměných JPEG obrázků a odpovídajících výsledků naší metody (viz Tabulka 5.3 pro EPP a Tabulka 5.4 pro SSIM) pro 15 iterací APGM. Vidíme, že chyba JPEG obrázků je pro malou kvalitu komprese relativně velká a naši metodě se daří tuto chybu účinně zmenšovat. Pro vyšší kvalitu komprese už není tak velký prostor pro zlepšení a zmenšení chyby už není tolik patrné. Jasnější představu nám umožní ukazatele relativního zlepšení v dalším odstavci.



Obrázek 5.6: Typický průběh vyčíslení proximálního operátoru ze vztahu (3.11). Minimalizovaný funkcionál je součtem l_1 -normy waveletové transformace obrazu a druhé mocniny vzdálenosti od počátečního obrazu (viz vztah (3.13)). Konkrétní hodnoty: Obrázek barbara, rozptyl šumu = 0, $q = 20$, $\alpha = 0,0057$, 1. krok APGM.

| | | EPP chyba obrázku | | | | | | | |
|---------|------|-------------------|-------|---------|------|--------|-------|--------|--|
| | | lena | | barbara | | ulice | | město | |
| kvalita | jpeg | metoda | jpeg | metoda | jpeg | metoda | jpeg | metoda | |
| 5 | 8.10 | 6.73 | 12.25 | 10.80 | 8.74 | 8.32 | 10.03 | 8.93 | |
| 10 | 5.49 | 4.70 | 9.35 | 8.45 | 3.62 | 3.29 | 7.36 | 6.68 | |
| 20 | 4.08 | 3.67 | 6.89 | 6.27 | 2.87 | 2.48 | 5.71 | 5.27 | |
| 30 | 3.52 | 3.23 | 5.55 | 5.06 | 1.45 | 1.21 | 4.94 | 4.56 | |
| 40 | 3.21 | 2.99 | 4.80 | 4.42 | 1.34 | 1.12 | 4.43 | 4.10 | |
| 50 | 2.99 | 2.81 | 4.28 | 3.95 | 1.01 | 0.94 | 4.05 | 3.76 | |

Tabulka 5.3: EPP chyba nezašuměných JPEG obrázků a výsledků naší metody v závislosti na kvalitě JPEG komprese pro 15 iterací APGM.

Účinnost metody v závislosti na maximálním počtu iterací

Dále nás zajímá, jak se mění vlastnosti metody pro různé počty provedených iterací. Pro všechny čtyři testovací obrázky a pro dané kvality JPEG komprese jsme nastavili maximální počet vyčíslení proximálního operátoru **MAXITER** a popsaným způsobem našli optimální parametr α . Poté jsme spočetli průměrná zlepšení obrázků pro zadané hodnoty.

Podívejme se, jaká je účinnost naší metody v závislosti na kvalitě komprese a počtu vyčíslení proximálního operátoru (viz Tabulka 5.5). Můžeme pozorovat, že účinnost naší metody klesá s rostoucí kvalitou JPEG komprese. Dále vidíme, že pro více než 5 iterací algoritmu APGM se výsledky metody příliš nemění. Pro nižší kvalitu komprese dává dokonce 3 až 5 iterací nejlepší výsledky. Nejvíce ekonomickou volbou tedy vychází použití 5-ti iterací APGM.

| SSIM chyba obrázku | | | | | | | | |
|--------------------|------|--------|---------|--------|-------|--------|-------|--------|
| kvalita | lena | | barbara | | ulice | | město | |
| | jpeg | metoda | jpeg | metoda | jpeg | metoda | jpeg | metoda |
| 5 | .666 | .733 | .618 | .665 | .691 | .726 | .622 | .658 |
| 10 | .783 | .825 | .745 | .770 | .785 | .813 | .738 | .763 |
| 20 | .859 | .878 | .844 | .858 | .859 | .875 | .825 | .837 |
| 30 | .888 | .900 | .888 | .899 | .892 | .904 | .867 | .876 |
| 40 | .904 | .912 | .910 | .919 | .911 | .921 | .892 | .899 |
| 50 | .915 | .921 | .925 | .932 | .924 | .932 | .910 | .916 |

Tabulka 5.4: SSIM chyba nezašuměných JPEG obrázků a výsledků naší metody v závislosti na kvalitě JPEG komprese pro 15 iterací APGM.

| Kvalita | Průměrné REPP(%) | | | | | Průměrné RSSIM(%) | | | | |
|---------|--------------------|------|------|------|------|--------------------|------|------|------|------|
| | Max. počet iterací | | | | | Max. počet iterací | | | | |
| | 1 | 3 | 5 | 10 | 15 | 1 | 3 | 5 | 10 | 15 |
| 5 | 12.5 | 12.9 | 13.1 | 12.5 | 12.1 | 14.0 | 14.5 | 14.5 | 13.7 | 13.3 |
| 10 | 10.2 | 10.9 | 11.1 | 11.0 | 11.0 | 12.4 | 13.0 | 13.0 | 12.8 | 12.8 |
| 20 | 7.8 | 8.7 | 9.0 | 9.1 | 9.0 | 9.2 | 9.9 | 10.2 | 10.3 | 10.1 |
| 30 | 6.9 | 8.1 | 8.5 | 8.7 | 8.7 | 8.0 | 9.1 | 9.5 | 9.7 | 9.6 |
| 40 | 6.2 | 7.4 | 7.8 | 8.0 | 8.0 | 7.2 | 8.3 | 8.7 | 8.9 | 8.8 |
| 50 | 5.7 | 6.9 | 7.3 | 7.5 | 7.5 | 6.6 | 7.6 | 8.1 | 8.2 | 8.2 |

Tabulka 5.5: Průměrné relativní zlepšení JPEG obrázků v závislosti na kvalitě komprese a počtu provedených iterací algoritmu APGM.

5.2.5 Závislost výkonnosti metody na šumu a kvalitě

Při dalším testování naší metody budeme postupovat následovně. Vezmeme v potaz, že náš algoritmus je navržen pro obrázky vzniklé z originálních procesem zašumění a následné JPEG komprese. Tento proces tedy budeme simulovat na testovacích obrázcích. Na takto zpracované obrázky pak použijeme naši metodu pro 5 iterací APGM a podle postupu ze sekce 5.2.2 nalezneme optimální parametr α . Pro toto α pak výsledek metody srovnáme s původním testovacím obrázkem. Nakonec spočteme průměrná zlepšení oproti JPEG obrázkům (viz Tabulka 5.6).

Vidíme, že pro zašuměné obrázky průměrné zlepšení roste spolu s kvalitou JPEG komprese. To je rozdíl oproti nezašuměným obrázkům. Čím je totiž kvalita komprese vyšší, tím více se šum v obrázku zachová a tím snáze se potom odstraní minimalizací waveletových koeficientů. Pro velký rozptyl šumu je relativní zlepšení značné. To ukazuje, že naše metoda je vhodná i pro odšumování kvalitnějších JPEG obrázků.

5.2.6 Srovnání s odšuměním

Metoda, kterou používáme pro vyčíslení proximálního operátoru, se často používá k odšumění obrázku. Odšumovací metodou proto nyní budeme rozumět provedení jednoho vyčíslení proximálního operátoru s nalezeným optimálním pa-

| Kvalita | Průměrné REPP (%) | | | | Průměrné RSSIM (%) | | | |
|---------|------------------------|------|------|------|------------------------|------|------|------|
| | Rozptyl šumu v obrázku | | | | Rozptyl šumu v obrázku | | | |
| | 0 | 5 | 10 | 15 | 0 | 5 | 10 | 15 |
| 5 | 13.1 | 13.6 | 14.7 | 16.7 | 14.5 | 15.0 | 16.3 | 18.4 |
| 10 | 11.1 | 12.0 | 14.7 | 21.9 | 13.0 | 14.1 | 18.3 | 29.7 |
| 20 | 9.0 | 11.3 | 22.3 | 34.5 | 10.2 | 14.3 | 34.0 | 48.4 |
| 30 | 8.5 | 13.4 | 28.0 | 40.0 | 9.5 | 18.9 | 44.0 | 55.0 |
| 40 | 7.8 | 15.3 | 31.5 | 44.2 | 8.7 | 23.8 | 49.3 | 58.8 |
| 50 | 7.3 | 17.2 | 34.3 | 48.2 | 8.1 | 28.3 | 53.1 | 61.9 |

Tabulka 5.6: Závislost průměrného relativního zlepšení na různých parametrech šumu a kvality JPEG komprese pro MAXITER = 5. Průměr počítán přes obrázky lena, barbara, ulice, město.

parametrem α . V naší implementaci se nejedná o nic jiného, než spuštění naší metody pro MAXITER = 1.

Stejným způsobem, jako v předchozí sekci jsme otestovali naši metodu pro jednu iteraci algoritmu APGM. Výsledky jsou shrnuty v Tabulce 5.7.

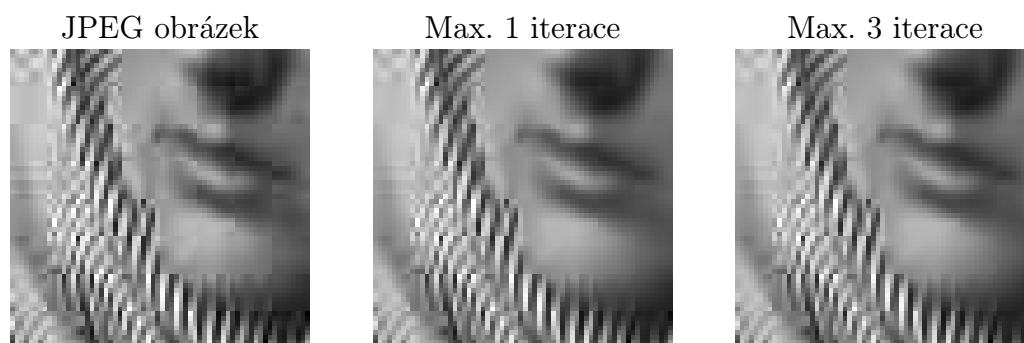
| Kvalita | Průměrné REPP (%) | | | | Průměrné RSSIM (%) | | | |
|---------|------------------------|------|------|------|------------------------|------|------|------|
| | Rozptyl šumu v obrázku | | | | Rozptyl šumu v obrázku | | | |
| | 0 | 5 | 10 | 15 | 0 | 5 | 10 | 15 |
| 5 | 12.5 | 12.9 | 14.1 | 15.7 | 14.0 | 14.3 | 15.6 | 17.6 |
| 10 | 10.2 | 11.2 | 14.1 | 21.8 | 12.4 | 13.6 | 18.1 | 30.3 |
| 20 | 7.8 | 10.3 | 21.9 | 34.5 | 9.2 | 13.4 | 34.1 | 49.1 |
| 30 | 6.9 | 12.4 | 27.7 | 39.7 | 8.0 | 18.2 | 44.3 | 54.9 |
| 40 | 6.2 | 14.3 | 31.0 | 43.3 | 7.2 | 23.1 | 49.4 | 58.0 |
| 50 | 5.7 | 16.2 | 33.5 | 46.7 | 6.6 | 27.7 | 52.6 | 60.1 |

Tabulka 5.7: Závislost průměrného relativního zlepšení na různých parametrech šumu a kvality JPEG komprese pro MAXITER = 1 (to odpovídá odšumění). Průměr počítán přes obrázky lena, barbara, ulice, město.

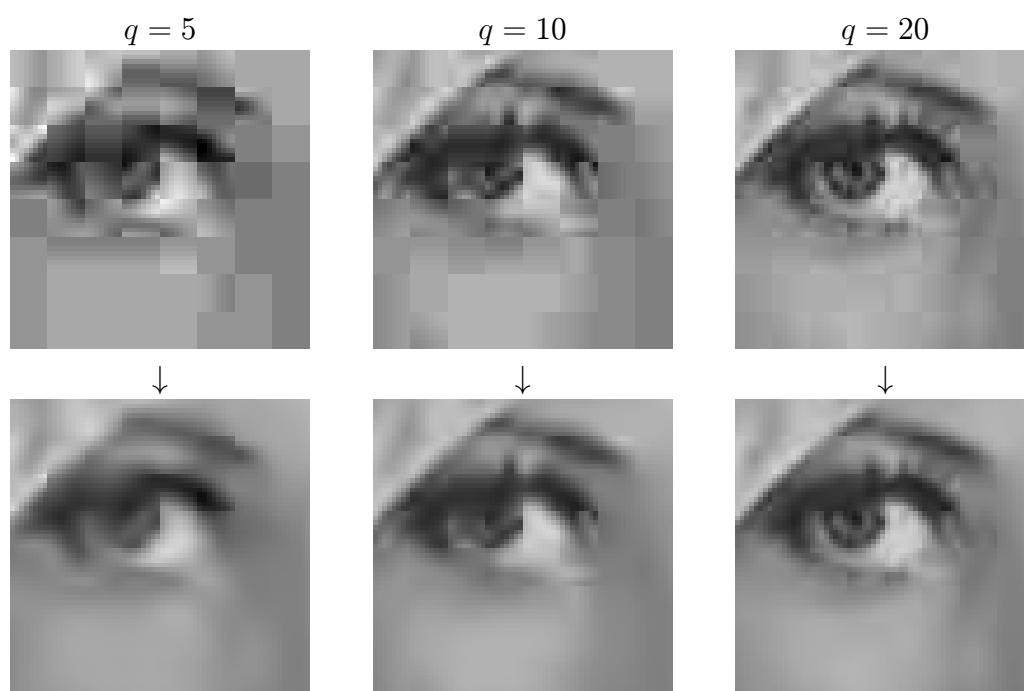
Při srovnání Tabulek 5.6 a 5.7 vidíme, že pro zašuměné obrázky jsou výsledky odšumění srovnatelné s výsledky naší metody. Naopak pro nezašuměné obrázky naše metoda dává lepší výsledky, než pouhé odšumění.

5.3 Obrazové ukázky výsledků

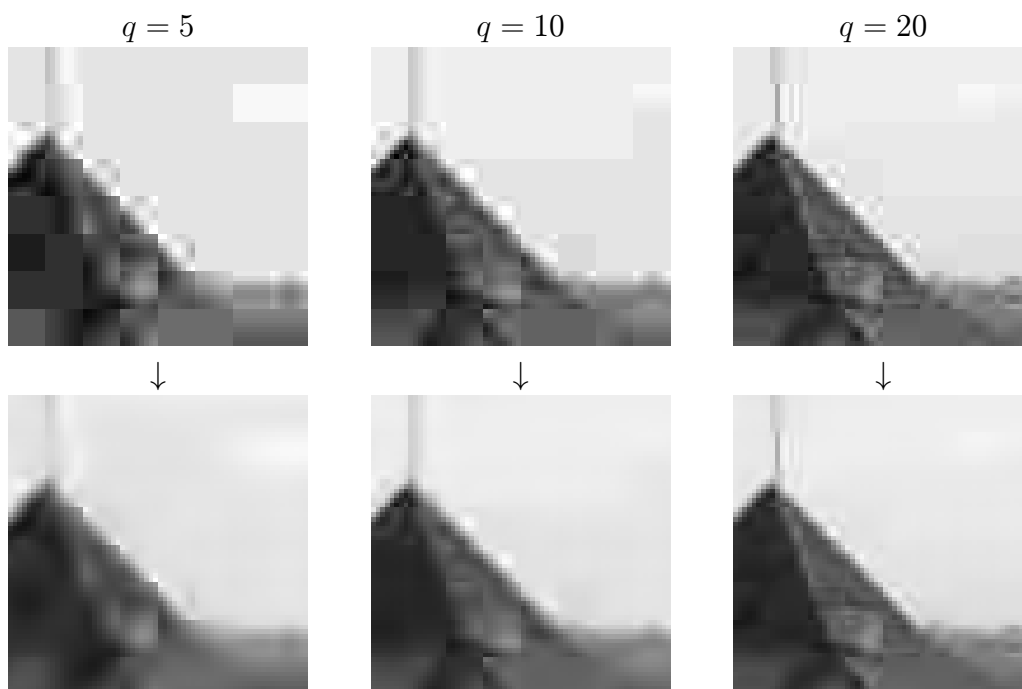
V této závěrečné sekci uvedeme několik obrazových ukázek výsledků naší metody.



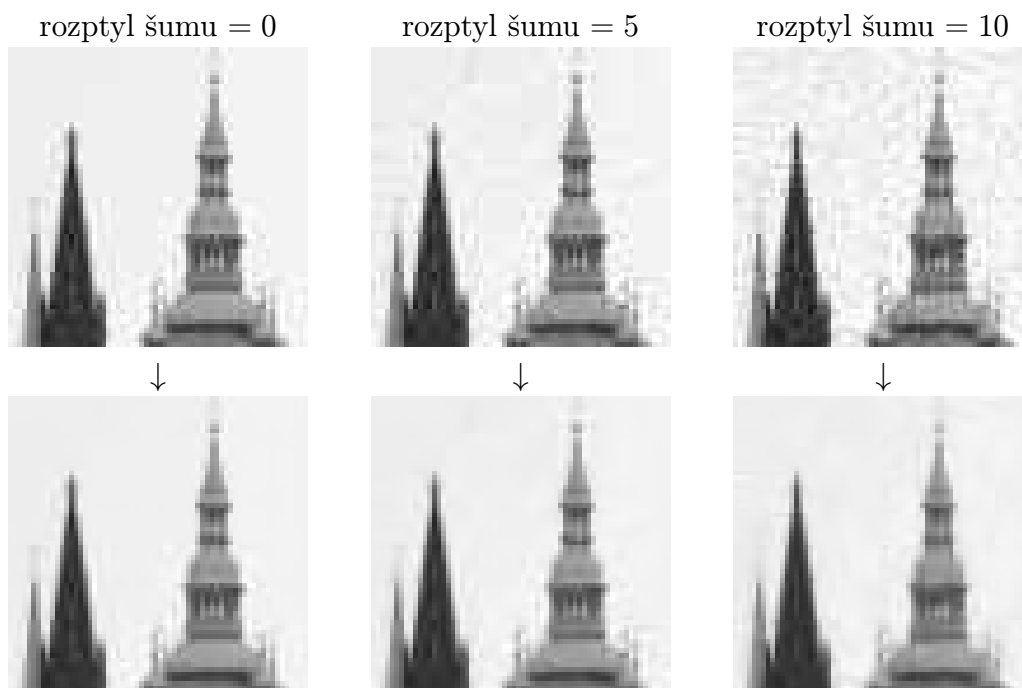
Obrázek 5.7: Srovnání detailů výsledků metody pro 1 a 3 iterace APM. Pro vyšší počet iterací vypadají výsledky metody k nerozeznání stejně. Testovací obrázek barbara při kvalitě komprese $q = 20$.



Obrázek 5.8: Srovnání výsledků metody s JPEG obrázky různé kvality v detailu. Spočteno 5 iterací APM pro optimální volby parametru α , testovací obrázek lena.



Obrázek 5.9: Srovnání detailu výsledků metody s JPEG obrázky různé kvality. Spočteno 5 iterací APM pro optimální volby parametru α , testovací obrázek město.



Obrázek 5.10: Srovnání výsledků metody na různě zašuměných obrázcích, komprimovaných JPEG kompresí kvality 40. Spočteno 15 iterací APM pro optimální volby parametru α , detail testovacího obrázku ulice.

Závěr

V této práci jsme se zabývali odstraňováním JPEG artefaktů z obrázků formátu JPEG. Narozdíl od existujících přístupů jsme do modelu vzniku JPEG komprimovaného obrázku zavedli přítomnost aditivního gaussovského šumu. Problém odstraňování JPEG artefaktů jsme formulovali jako problém minimalizace funkcionálu, vyjadřujícího pravděpodobnost, že je daný obrázek „správný“ za informací známých z dat JPEG obrázku a za jisté apriorní informace. Tento funkcionál jsme vyčíslili pomocí Bayesovy věty. Ke konstrukci apriorní informace jsme použili moderní komplexní wavelety, které umožnily charakterizaci naší představy o „správném“ obrázku. K minimalizaci funkcionálu jsme použili proximálního algoritmu, který narozdíl od běžných metod minimalizace nevyžaduje diferencovatelnost minimalizovaného funkcionálu. Samotnou implementaci metody jsme provedli v MATLABu. Navrženou metodu jsme testovali pro vícero obrázků různé kvality JPEG komprese. Získali jsme vizuální výsledky naší metody a porovnávali jsme i kvantitativní redukci chyby v závislosti na typu obrázku. Celkově se ukázalo, že naše metoda úspěšně odstraňuje JPEG artefakty z různých obrázků s různým zastoupením šumu a s různými kvalitami JPEG komprese.

Seznam použité literatury

- ALTER, F., DURAND, S. a FROMENT, J. (2004). Deblocking dct-based compressed images with weighted total variation. In *ICASSP (3)*, pages 221–224.
- ANDĚL, J. (2007). *Základy matematické statistiky*. Matfyzpress. ISBN 9788073780012.
- BAUSCHKE, H. a COMBETTES, P. (2011). *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer. ISBN 9781441994677.
- BOYD, S. a VANDENBERGHE, L. (2004). *Convex Optimization*. Berichte über verteilte messsysteme. Cambridge University Press. ISBN 9780521833783.
- BREDIES, K. a HOLLER, M. (2012). A total variation-based jpeg decompression model. *SIAM J. Img. Sci.*, **5**(1), 366–393. ISSN 1936-4954. doi: 10.1137/110833531.
- CASTAGNO, R., MARSI, S. a RAMPONI, G. (1998). A simple algorithm for the reduction of blocking artifacts in images and its implementation. *Consumer Electronics, IEEE Transactions on*, **44**(3), 1062–1070. ISSN 0098-3063. doi: 10.1109/30.713235.
- CHAMBOLLE, A. a POCK, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, **40**(1), 120–145. ISSN 0924-9907. doi: 10.1007/s10851-010-0251-1.
- COMBETTES, P. a PESQUET, J.-C. (2011). Proximal splitting methods in signal processing. In BAUSCHKE, H. H., BURACHIK, R. S., COMBETTES, P. L., ELSER, V., LUKE, D. R. a WOLKOWICZ, H., editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Springer Optimization and Its Applications, pages 185–212. Springer New York. ISBN 9781441995681. doi: 10.1007/978-1-4419-9569-8_10.
- DUPAČ, V. a HUŠKOVÁ, M. (2005). *Pravděpodobnost a matematická statistika*. Učební texty Univerzity Karlovy v Praze. Karolinum. ISBN 9788024600093.
- HASS, C. (2008). Jpeg compression quality from quantization tables. <http://www.impulseadventure.com/photo/jpeg-quantization.html>. Navštíveno 25. 7. 2014.
- HSUNG, T.-C., LUN, D. a SIU, W.-C. (1996). A deblocking technique for jpeg decoded image using wavelet transform modulus maxima representation. In

- Image Processing, 1996. Proceedings., International Conference on*, volume 1, pages 561–564 vol.2. doi: 10.1109/ICIP.1996.560920.
- JACOD, J. a PROTTER, P. (2000). *Probability essentials*. Universitext Series. Springer. ISBN 9783540664192.
- KINGSBURY, N. (2001). Complex wavelets for shift invariant analysis and filtering of signals. *Applied and Computational Harmonic Analysis*, **10**(3), 234 – 253. ISSN 1063-5203.
- LUKEŠ, J. a MALÝ, J. (2002). *Míra a integrál*. Učební texty. Karolinum. ISBN 9788024605432.
- NAJZAR, K. (2004). *Základy teorie waveletů*. Učební texty. Karolinum. ISBN 9788024609577.
- NALLAPERUMAL, K., RANJANI, J., CHRISTOPHER, S. a SAUDIA, S. (2006). Removal of blocking artifacts in jpeg compressed images using dual tree complex wavelet filters for multimedia on the web. In *Wireless and Optical Communications Networks, 2006 IFIP International Conference on*, pages 4 pp.–4. doi: 10.1109/WOCN.2006.1666679.
- PARIKH, N. a BOYD, S. (2013). *Proximal Algorithms*. Foundations and Trends(r) in Optimization. Now Publishers Incorporated. ISBN 9781601987167.
- POYNTON, C. (2003). *Digital Video and HDTV: Algorithms and Interfaces*. Electronics & Electrical. Morgan Kaufmann Publishers. ISBN 9781558607927.
- ROCKAFELLAR, R. (1997). *Convex Analysis*. Convex Analysis. Princeton University Press. ISBN 9780691015866.
- WALLACE, G. K. (1991). The jpeg still picture compression standard. *Commun. ACM*, **34**(4), 30–44. ISSN 0001-0782. doi: 10.1145/103085.103089.
- WANG, Z., BOVIK, A., SHEIKH, H. a SIMONCELLI, E. (2004). Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, **13**(4), 600–612. ISSN 1057-7149. doi: 10.1109/TIP.2003.819861.
- YANG, Y., GALATSANOS, N. a KATSAGGELOS, A. K. (1995). Projection-based spatially-adaptive reconstruction of block-transform compressed images.
- ZOU, J. J. a YAN, H. (2005). A deblocking method for bdct compressed images based on adaptive projections. *IEEE Trans. Cir. and Sys. for Video Technol.*, **15**(3), 430–435. ISSN 1051-8215. doi: 10.1109/TCSVT.2004.842610.