

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2017

Bc. Jakub Jirák



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

ALTERNATIVNÍ JPEG KODÉR/DEKODÉR

AN ALTERNATIVE JPEG CODER/DECODER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jakub Jirák

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Mgr. Pavel Rajmic, Ph.D.

BRNO 2017



Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Jakub Jirák

ID: 146030

Ročník: 2

Akademický rok: 2016/17

NÁZEV TÉMATU:

Alternativní JPEG kodér/dekodér

POKYNY PRO VYPRACOVÁNÍ:

Téma práce se týká známého obrazového kodeku JPEG. Nejprve podrobně nastudujte jak JPEG funguje a dále pochopíte teorii k tzv. řídkým reprezentacím signálů. Tuto teorii vzápětí použijte pro formulaci alternativního dekodéru, který bude schopen potlačovat známé nešvary JPEGu jako jsou blokové artefakty a zvlnění na hranách v obrazu. Po zvládnutí této základní verze rozšířte experiment tak, že upravíte jak kodér, tak dekodér. Kompresi obrazu bude po úpravě probíhat současně v obrazové i kmitočtové oblasti, což přinese další zlepšení kvality obrazu. Všechny programy implementujte v MATLABu a otestujte funkčnost a srovnajte na simulovaných a reálných obrázcích.

DOPORUČENÁ LITERATURA:

[1] Hrbáček, R., Rajmic, P., Veselý, V., Špiřík, J. Řídké reprezentace signálů: úvod do problematiky, Elektrevue, 2011. ISSN 1213-1539

[2] Špiřík, J., Rajmic, P., Veselý, V. Reprezentace signálů: od bází k framům, Elektrevue, 2010. ISSN 1213-1539

Termín zadání: 1.2.2017

Termín odevzdání: 24.5.2017

Vedoucí práce: doc. Mgr. Pavel Rajmic, Ph.D.

Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Kodek JPEG je v současné době nejrozšířenější obrázkový formát. Tato práce se zabývá návrhem a implementací alternativního JPEG kodeku využívajícího proximální algoritmy v kombinaci se zafixováním bodů z původního obrazu k potlačení artefaktů vznikajících v běžném JPEG kodéru. V rámci řešení daného problému byla nejprve využita prox_{TV} a následně Douglas-Rachford algoritmus, pro který byla odvozena speciální funkce využívající l_1 -normu k rekonstrukci obrazu. Výsledky navrženého řešení jsou velmi dobré, jelikož dokáže efektivně potlačit vzniklé artefakty a zároveň výsledek odpovídá obrazu s vyšším nastaveným kvalitativním faktorem. Navržená metoda dosahuje velmi dobrých výsledků jak pro jednoduché obrázky tak pro fotografie, avšak v případě velkých obrázků (1024×1024 px) a větších je zapotřebí velkého množství výpočetního času, proto je metoda vhodná spíše pro menší obrázky.

KLÍČOVÁ SLOVA

JPEG, MATLAB, UnLocBox, Totalní Variace, Douglas-Rachford, proximální operátor, řídká reprezentace signálů, EPP, REPP, MSE, PSNR, SSIM

ABSTRACT

The JPEG codec is currently the most widely used image format. This work deals with the design and implementation of an alternative JPEG codec using proximal algorithms in combination with the fixation of points from the original image to suppression of artifacts created in common JPEG coding. To solve the problem, the prox_{TV} and then the Douglas-Rachford algorithm were used, for which special functions using l_1 -norm for image reconstruction were derived. The results of the proposed solution are very good because they can effectively suppress the artefacts created and the result corresponds to the image with a higher set qualitative factor. The proposed method achieves very good results for both simple images and photos, but in the case of large images (1024×1024 px) and larger, a large amount of computing time is required, so the method is more suitable for smaller images.

KEYWORDS

JPEG, MATLAB, UnLocBox, Total Variation, Douglas-Rachford, sparse representation of signals, proximal operator, EPP, REPP, MSE, PSNR, SSIM

JIRÁK, Jakub. *Alternativní JPEG kodér/dekodér*. Brno, 2017, 83 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: doc. Mgr. Pavel Rajmic, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Alternativní JPEG kodér/dekodér“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno

.....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc. Mgr. Pavlu Rajmicovi, Ph.D. za odbornou pomoc, věnovaný čas a velmi cenné rady při psaní práce. Dále bych chtěl poděkovat rodině za podporu a trpělivost.

Brno

.....

podpis autora



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

PODĚKOVÁNÍ

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

Brno

.....

podpis autora



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

| | |
|--|-----------|
| Úvod | 13 |
| 1 Ztrátová komprese obrazu | 15 |
| 1.1 Klasifikace kvality | 15 |
| 1.2 Transformační kódování | 15 |
| 1.3 Metody využití k měření chyb | 16 |
| 1.3.1 Průměrná chyba na pixel | 16 |
| 1.3.2 Střední kvadratická chyba | 17 |
| 1.3.3 Špičkový poměr signálu k šumu | 18 |
| 1.3.4 Index strukturální podobnosti | 18 |
| 1.4 Způsob editace obrázků | 20 |
| 2 JPEG | 21 |
| 2.1 Struktura ztrátového JPEG kodéru | 21 |
| 2.1.1 Transformace barev | 22 |
| 2.1.2 Podvzorkování | 23 |
| 2.1.3 Dopředná DCT | 23 |
| 2.1.4 Kvantizace | 25 |
| 2.1.5 Mapování koeficientů na symboly | 26 |
| 2.1.6 Entropický kodér | 27 |
| 2.2 Struktura dekodéru | 29 |
| 2.2.1 Entropický dekodér | 30 |
| 2.2.2 Mapování symbolů na koeficienty | 31 |
| 2.2.3 Inverzní kvantizace | 31 |
| 2.2.4 Inverzní DCT | 31 |
| 2.2.5 Nadvzorkování | 31 |
| 2.2.6 Zpětná barevná transformace | 32 |
| 2.3 Obrazové artefakty | 32 |
| 2.3.1 Blokované artefakty | 32 |
| 2.3.2 Barevné zkreslení | 33 |
| 2.3.3 Artefakty v okolí ostrých přechodů | 33 |
| 2.3.4 Rozmazání | 33 |
| 2.4 Alternativní módy | 33 |
| 2.4.1 Progresivní mód | 33 |
| 2.4.2 Hierarchický mód | 34 |
| 2.4.3 Bezeztrátový mód | 34 |
| 2.5 Shrnutí | 34 |

| | | |
|----------|---|-----------|
| 2.5.1 | Výhody | 34 |
| 2.5.2 | Nevýhody | 35 |
| 3 | Řídké reprezentace signálů | 37 |
| 3.1 | Základní pojmy a značení | 37 |
| 3.2 | Hledání řídkého řešení | 40 |
| 4 | Proximální algoritmy | 44 |
| 4.1 | Omezený a neomezený tvar optimalizačních úloh | 44 |
| 4.2 | Proximální operátor | 45 |
| 4.3 | Proximální operátor l_1 -normy | 46 |
| 4.4 | Proximální operátor pro totální variaci | 46 |
| 4.5 | Douglas-Rachford algoritmus | 48 |
| 5 | Formulace řešení | 50 |
| 5.1 | Využité nástroje | 50 |
| 5.1.1 | UNLocBoX | 50 |
| 5.1.2 | Dema pro podporu výuky předmětů MGMP a BZSG | 50 |
| 5.1.3 | Externí knihovny | 51 |
| 5.2 | Vytvořené funkce a jejich činnost | 51 |
| 5.2.1 | Proximální operátor pro totální variaci | 51 |
| 5.2.2 | Douglas-Rachford algoritmus | 52 |
| 5.3 | Navrhnutý JPEG kodér/dekodér | 53 |
| 5.3.1 | Volba bodů | 54 |
| 5.3.2 | Doplnění bodů | 54 |
| 5.3.3 | Prahování | 55 |
| 6 | Shrnutí dosažených výsledků | 56 |
| 6.1 | Proximální totální variace | 56 |
| 6.2 | Douglas-Rachford algoritmus | 60 |
| 6.3 | Alternativní JPEG kodek | 62 |
| 6.4 | Demo DP | 63 |
| 6.5 | Možnosti navázání na tuto práci | 63 |
| 7 | Závěr | 65 |
| | Literatura | 67 |
| | Seznam symbolů, veličin a zkratk | 72 |
| | Seznam příloh | 74 |

| | | |
|----------|---|-----------|
| A | Rozdíl mezi obrázky | 75 |
| B | Obsah přiloženého disku | 79 |
| C | Hodnoty EPP, REPP, MSE, PSNR, SSIM | 83 |

SEZNAM OBRÁZKŮ

| | | |
|------|--|----|
| 1.1 | Sledování závislosti hodnoty EPP na zvolené kvalitě. | 17 |
| 1.2 | Sledování závislosti hodnoty MSE na zvolené kvalitě. | 18 |
| 1.3 | Sledování závislosti hodnoty PSNR na zvolené kvalitě. | 19 |
| 1.4 | Sledování závislosti hodnoty SSIM na zvolené kvalitě. | 20 |
| 1.5 | Zobrazení způsobu úpravy obrázků. | 20 |
| 2.1 | Nejběžnější implementace JPEG kodéru | 22 |
| 2.2 | JPEG kodér podle standardu | 22 |
| 2.3 | JPEG struktura souboru | 22 |
| 2.4 | Obrázky znázorňují různé typy podvzorkování | 24 |
| 2.5 | Barevná transformace z RGB do $YCbCr$ s podvzorkováním 4:2:2. | 24 |
| 2.6 | Dekompozice obrázku. | 25 |
| 2.7 | Výše jsou uvedeny doporučené kvantizační tabulky pro jasovou složku | 26 |
| 2.8 | Cik-cak vyčítání koeficientů | 30 |
| 2.9 | JPEG dekodér podle standardu | 31 |
| 2.10 | Nejběžnější implementace JPEG dekodéru | 31 |
| 2.11 | Zobrazení rozdílu mezi originálem a obrázkem s degradovanou kvalitou. | 32 |
| 2.12 | Zobrazení rozdílu mezi originálem a degradovaným obrázkem. | 36 |
| 3.1 | Znázornění jednotkových koulí | 38 |
| 3.2 | Grafické znázornění uvedeného příkladu. | 40 |
| 3.3 | Znázornění „nafukující se“ jednotkových koulí | 42 |
| 4.1 | Ukázka funkce měkkého prahování | 47 |
| 4.2 | Znázornění vlivu regularizačního parametru λ | 48 |
| 4.3 | Ukázka konvergence k výsledku u Douglas-Rachford algoritmu. | 49 |
| 5.1 | Ukázka rozdílu mezi funkcemi g_{input} a g_{inputc} | 51 |
| 5.2 | Dosažené výsledky pro navržený DR. | 53 |
| 5.3 | Alternativní JPEG kodek. | 54 |
| 5.4 | Struktura uložených bodů. | 54 |
| 5.5 | Ukázka konvergence k výsledku pro navržený DR algoritmus. | 55 |
| 6.1 | Zobrazení výsledků pro blokový diagram. | 58 |
| 6.2 | Zobrazení výsledků pro černobílou fotografii Leny. | 59 |
| 6.3 | Ukázka výsledků pro <code>demo_DR_test_manual.m</code> | 60 |
| 6.4 | Zobrazení konvergence k výsledku pro DR algoritmus. | 61 |
| 6.5 | Zobrazení vlivu parametru γ na konvergenci k výsledku pro DR algoritmus. | 61 |
| 6.6 | Dosažené výsledky pro navržený DR. | 62 |
| 6.7 | Ukázka zobrazení porovnání výsledků dílčích částí s navrženým alternativním JPEG. | 64 |

| | | |
|-----|---|----|
| A.1 | Zobrazení rozdílu mezi originálem a upraveným obrázkem pro obr. 2.12. | 75 |
| A.2 | Zobrazení rozdílu mezi originálem a upraveným obrázkem pro obr. 5.2. | 76 |
| A.3 | Zobrazení rozdílu mezi originálem a upraveným obrázkem pro obr. 5.2. | 76 |
| A.4 | Zobrazení rozdílu mezi originálem a upraveným obrázkem pro obr. 6.3. | 77 |
| A.5 | Zobrazení rozdílu mezi originálem a upraveným obrázkem pro obr. 6.6. | 78 |

SEZNAM TABULEK

| | | |
|-----|--|----|
| 1.1 | Tabulka orientační kvality komprese JPEG | 16 |
| 2.1 | Historie JPEG | 21 |
| 2.2 | Tabulka kategorií pro rozdělení DC složek při 8 bpp | 28 |
| 2.3 | Huffmanova tabulka 162 kombinací (R S) pro jas. složku při 8 bpp | 29 |
| 2.4 | Tabulka kategorií pro rozdělení AC složek při 8 bpp | 30 |
| 4.1 | Tabulka definující posunutí | 46 |
| 6.1 | Shrnutí dosažených výsledků pro blokový diagram. | 57 |
| 6.2 | Shrnutí dosažených výsledků pro fotografii. | 57 |
| 6.3 | Nastavené hodnoty parametrů | 60 |
| 6.4 | Porovnání výsledku vůči poškozenému obrázku | 60 |
| 6.5 | Porovnání hodnot pro JPEG Q=5 a rekonstruovaný obrázek | 63 |
| 6.6 | Porovnání výsledků (demo_DP.m) pro všechny dílčí metody | 63 |
| C.1 | Tabulka hodnot pro blokový diagram. | 83 |

ÚVOD

Diplomová práce se zabývá návrhem alternativního JPEG kodéru/dekodéru. Důvod pro návrh alternativního kodeku¹ je snaha potlačit artefakty vznikající při kompresi obrázků pomocí JPEG kodeku. Vznikající artefakty byly popsány v kapitole č. 2.3. V této práci jsou hledány způsoby pomoci, kterých je tyto artefakty možné potlačit případně odstranit s pomocí využití aktuálního trendu v oblasti zpracování signálů. Jde o signály s tzv. *řídkou reprezentací*. O tuto formu signálů se zajímají odborníci téměř posledních 35 let. Samotný název již napovídá jakou povahu mají signály, které budou v práci jako takové řešeny. Řídká reprezentace představuje signál, který je možné buď vyjádřit přesně, nebo aproximovat pomocí lineární kombinace velmi malého počtu vektorů pocházejících z vybraného systému.

V první kapitole je věnována pozornost ztrátové kompresi. Nejprve je uveden krátký úvod, na který je navázáno klasifikací kvality. Následuje transformační kódování, kde dojde i k definici orientační bitové kvality komprese JPEG. Závěr první kapitoly je věnován popisu metod využitých k měření chybovosti obrazu. Kapitola druhá je úvodem do problematiky kolem ztrátového JPEG, nejprve je podrobně rozebrán JPEG kodér, následně dekodér.

Dále je v této kapitole věnována pozornost obrazovým artefaktům a alternativním módům, ve kterých může JPEG pracovat. Druhá kapitola je zakončena shrnutím uvedených poznatků.

Třetí kapitola pojednává o úvodu do problematiky kolem řídkých reprezentací signálů. Je zde věnována pozornost sjednocení značení a uvedení definic nutných k pochopení problematiky kolem řídkých reprezentací signálů. V podkapitole č. 3.2 je věnována pozornost nalezení řídkého řešení.

Čtvrtá kapitola byla vymezena pro proximální algoritmy. Nejprve je vysvětlen omezený a neomezený tvar optimalizačních úloh. Načež je navázáno definicí proximálního operátoru, což je základní stavební kámen další proximální operátory. V návaznosti na proximální operátor bude popsán proximální operátor l_1 -normy a po něm proximální operátor pro totální variaci. Závěr této kapitoly byl věnován popisu Douglas-Rachford algoritmu.

Následující kapitola je věnována stanovení řešení odstranění artefaktů vznikajících při kompresi pomocí JPEG. Tato kapitola byla rozčleněna do několika částí. Nejprve budou přiblíženy využitě nástroje, poté dojde ke stručnému popisu vytvořených funkcí. Poté bude popsán navrhnutý alternativní JPEG kodek.

Shrnutí dosažených výsledků bude umístěno v kapitole č. 6. Nejprve zde bude věnována pozornost výsledkům dosažených pomocí proximální totální variace. Následně

¹Kodek – tímto slovem bude označeno spojení kodéru (coder) a dekodéru (decoder).

dosaženým výsledkům pomocí Douglas-Rachford algoritmu. Poté budou představeny výsledky, kterých dosahuje navržený alternativní JPEG kodek a na závěr této kapitoly bude uvedeno porovnání dosažených dílčích výsledků s konečným výsledkem.

Vzhledem k předpokladu nepřesnosti tisku byly některé obrázky v této práci zvýrazněny v programu Adobe Photoshop CC. U konkrétních obrázků bude tato informace uvedena a v příloze **A** bude uveden originál společně s upraveným obrázkem, aby bylo možné tyto obrázky porovnat.

1 ZTRÁTOVÁ KOMPRESSE OBRAZU

Metody zabývající se ztrátovou kompresí využívají psychovizuální redundance, aby ušetřily objem potřebných dat. Při kompresi jde jednak o odstranění informací nepostřehnutelných lidským okem a zároveň o potlačení dat, která málo přispívají k úrovni kvality reprezentace digitálního obrazu [1].

1.1 Klasifikace kvality

Kompresní algoritmy slouží k omezení množství potřebných dat, neboli k omezení kvantity na množství, při kterém neutrpí kvalita původního signálu. Kvantita je udávána pomocí kompresního poměru, což je vyjádření počtu dat výchozího snímku ku počtu dat snímku zredukovaného pomocí kompresního algoritmu. Tento poměr je uváděn ve tvaru „ $x:1$ “. Kompresní poměr může být vyjádřen například pomocí počtu bitů na jeden obrázkový bod, neboli v počtu bitů na pixel (bpp). Uvážíme-li chromatický tříkanálový obrázek, který má 24 bpp, což odpovídá 8 bpp na jednu barevnou složku. Po komprimaci má například 1,5 bpp, tedy 0,5 bpp na jednu barevnou složku. Kompresní poměr bude odpovídat 16:1. V tabulce 1.1 je znázorněno jaké kvalitě odpovídá přibližně určitý počet bpp, pak tedy pro uvedený případ jde o střední až dobrou kvalitu komprimovaného obrazu [2, 3].

Kvalitu je možné určit pomocí jednoduchých metod a to např. pomocí střední kvadratické odchylky (MSE), pomocí střední absolutní chyby (MSA), pomocí odstupe signálu od šumu (SNR, PSNR). Tyto jednoduché metody většinou nevedou k uspokojivému výsledku, který by se shodoval se subjektivním dojmem, který je při reprezentaci obrazu klíčový. Dalším možným kritériem je strukturálně-podobnostní index (structural similarity index, SSIM), který se řadí mezi vylepšená kritéria. Avšak neexistuje žádná metoda, která by byla schopna napodobit vnímání obrazu lidmi. Toto je důvod, proč jedinou možnou metodou k posouzení kvality obrazu zůstávají subjektivní testy [1, 2, 4].

1.2 Transformační kódování

Ztrátové metody většinou využívají jinou sféru, než přímo výchozí obrazovou (prostorovou). K překlenutí mezi obrazovou doménou do jiné využívají ztrátové metody různých matematických transformací, čímž docílí uskupení informací v jiné sféře. V jiné doméně může dokonce dojít ke zvýšení efektivity kódování. Nejvyužívanější transformace jsou fourierovského typu. Jsou využívány i další transformace např. waveletová (vlnková) transformace, Gaborova transformace a mnohé další. Transformace

jsou využívány kvůli svým vlastnostem [2]:

- (bi)ortogonalita,
- reverzibilita,
- separovatelnost,
- rychlost [2].

Pro opravdovou účinnost ztrátových formátů je potřeba využít důmyslné kvantování, které bude mít velkou rozlišovací schopnost v místech s výskytem významných informací a naopak malou rozlišovací schopnost všude jinde. Dále je třeba, aby byly tyto metody v konečné fázi neztrátové. Obvykle se vyskytují složitější kodéry, které mají za úkol velký stupeň komprimace za udržení odpovídající kvality. Naopak dekodéry bývají jednodušší, aby bylo možné komprimovaná data co nejrychleji zobrazit. Tato diplomová práce se dále bude zabývat kodekem JPEG, který patří mezi kodeky založené na transformacích [2, 3, 4].

Tab. 1.1: Tabulka orientační kvality komprese JPEG, pokud uvažuje výchozí obrázek RGB 24 bpp [2, 5].

| rozsahy hodnot | kvalita |
|-----------------|---|
| 0,25 – 0,50 bpp | střední až dobrá |
| 0,50 – 0,75 bpp | dobrá až velmi dobrá, vhodná pro většinu aplikací |
| 0,75 – 1,50 bpp | velmi dobrá až vynikající |
| 1,50 – 2,00 bpp | prakticky bez rozpoznatelné ztráty |

1.3 Metody využití k měření chyb

V této podkapitole bude krátce věnována pozornost využitým metodám k posuzování kvality rekonstruovaných obrázků vůči originálnímu obrázku.

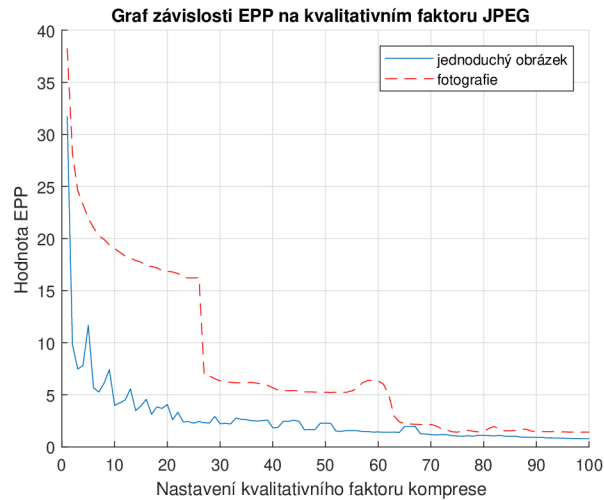
1.3.1 Průměrná chyba na pixel

Průměrná chyba na jeden pixel, představuje odchylku rekonstruovaného obrazu od originálu. Tato odchylka je vyjádřena vztahem:

$$EPP(x, y) = \frac{1}{MN} \sqrt{\sum_{m=1}^M \sum_{n=1}^N (x_{m,n} - y_{m,n})^2}, \quad (1.1)$$

kde M představuje šířku obrázku, N jeho výšku, x je bod v originálním obrázku a y je hodnota stejného bodu v rekonstruovaném obrázku. Tento vztah tedy udává průměrnou velikost odchylky od originálního obrázku [48]. Implementace této metody byla umístěna do adresáře „mereni_kvality“ pod názvem „aepp.m“. Grafické

znázornění velikosti průměrné odchylky hodnot pixelů v závislosti na zvoleném kvalitativním faktoru Q je uvedeno v obrázku 1.1.



Obr. 1.1: Sledování závislosti hodnoty EPP na zvolené kvalitě. Vytvořeno pomocí funkce s názvem „demo_EPP_graf.m“ uložené v adresáři vytvořeno.

Relativní průměrná chyba na pixel

Tato metoda vyjadřuje relativní zlepšení EPP a je definována takto:

$$REPP = 1 - \frac{EPP(\text{rekonstruovaného obrazu})}{EPP(\text{vstupního degradovaného JPEG})} \quad [48]. \quad (1.2)$$

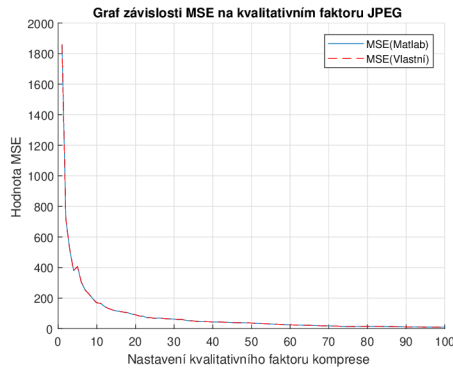
Implementace této metody byla umístěna do adresáře „mereni_kvality“ pod názvem „raepp.m“.

1.3.2 Střední kvadratická chyba

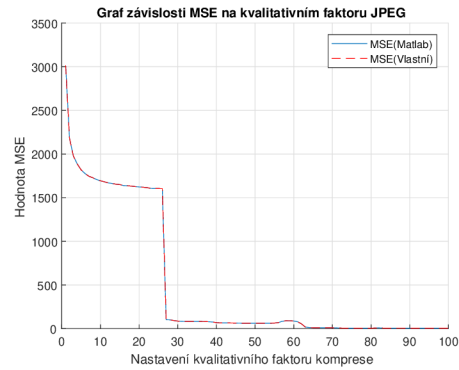
Mean squared error (MSE) udává rozdíl mezi originálem a rekonstruovaným obrázkem vůči hodnotě $(2^n - 1)^2$, kde n představuje bitovou hloubku, tedy počet bitů na jeden pixel. MSE je dáno vztahem:

$$MSE(x, y) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N (x_{m,n} - y_{m,n})^2, \quad (1.3)$$

kde M představuje šířku obrázku, N jeho výšku, x je bod v originálním obrázku a y je hodnota stejného bodu v rekonstruovaném obrázku [49]. Implementace této metody byla umístěna do adresáře „mereni_kvality“ pod názvem „mse_custom.m“, avšak ve skriptech byla využita funkce naimplementovaná přímo v matlabu, jelikož je komplexněji naimplementovaná a udává přesnější výsledky. Závislost velikosti střední kvadratické chyby na kvalitativním faktoru znázorňuje obrázek 1.2.



(a) jednoduchý obrázek



(b) fotografie

Obr. 1.2: Sledování závislosti hodnoty MSE na zvolené kvalitě. Vytvořeno pomocí funkce s názvem „demo_MSE_graf.m“ uložené v adresáři vytvořeno.

1.3.3 Špičkový poměr signálu k šumu

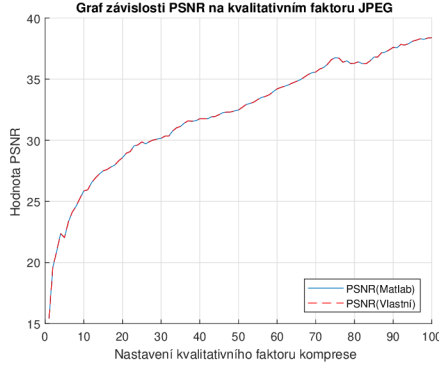
Hodnota Peak Signal to Noise Ratio (PSNR) závisí na velikosti MSE a je udávána v decibelech. Aby bylo možné určit velikost PSNR je potřeba mít k dispozici původní obraz, což nemusí být vždy pravidlem. Velká hodnota PSNR většinou odpovídá vysoké kvalitě obrazu, obdobně to platí i pro nízkou hodnotu PSNR. Kde nízká hodnota indikuje špatnou kvalitu obrazu. Výpočet PSNR odpovídá vztahu:

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} \text{ [dB]} \text{ [49]}. \quad (1.4)$$

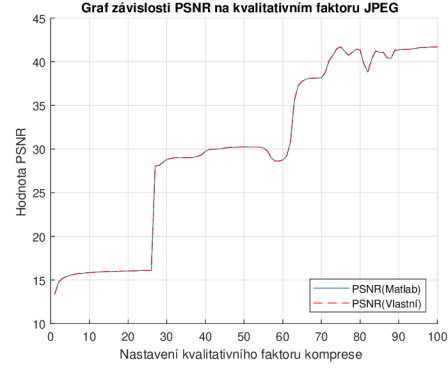
Implementace této metody byla umístěna do adresáře „mereni_kvality“ pod názvem „psnr_custom.m“, avšak ve skriptech byla využita funkce naimplementovaná přímo v matlabu, jelikož výsledky obou metod korespondují pouze při použití u jednoduchých obrázků. Závislost velikosti špičkového poměru signálu k šumu na kvalitativním faktoru znázorňuje obrázek 1.3.

1.3.4 Index strukturální podobnosti

Posledním sledovaným parametrem bude hodnota Structural Similarity Indexu. Tato metoda pracuje se strukturou dat díky čemuž se více blíží tomu, jakým způsobem obraz vnímá lidské oko. Na základě její podobnosti se blíží k jedničce. Pokud by SSIM vyšlo jedna, pak se jedná o identitu. Tvůrci této metody vycházeli z předpokladu, že nezáleží na jasu objektu a snažili se co nejvíce odstranit vliv osvětlení. SSIM může nabývat hodnot v rozmezí $\{-1; 1\}$. Metoda SSIM je vyjádřena těmito vztahy



(a) jednoduchý obrázek



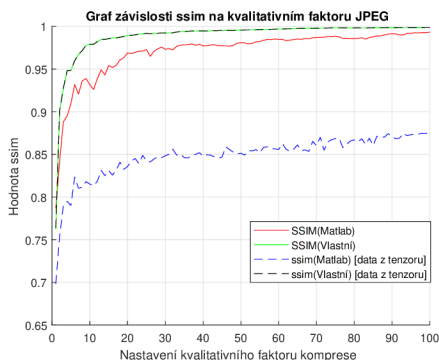
(b) fotografie

Obr. 1.3: Sledování závislosti hodnoty PSNR na zvolené kvalitě. Vytvořeno pomocí funkce s názvem „demo_PSNR_graf.m“ uložené v adresáři vytvořeno.

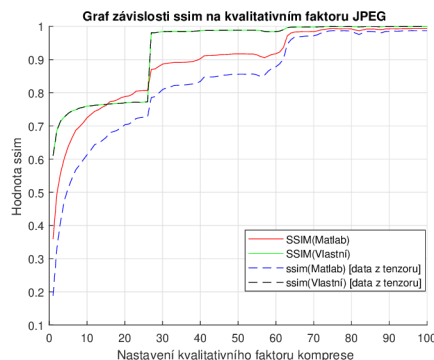
[49, 50, 51]:

$$\begin{aligned}
 SSIM(x, y) &= \frac{(2\mu_x\mu_y + C_1)(2\delta_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\delta_x^2 + \delta_y^2 + C_2)}, \\
 \mu_x &= \frac{1}{N} \sum_{i=1}^N x_i, \\
 \delta_x &= \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2}, \\
 \delta_{xy} &= \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y), \\
 C_1 &= (K_1L)^2, \\
 C_2 &= (K_2L)^2,
 \end{aligned} \tag{1.5}$$

kde x a y zastupují data z původního a rekonstruovaného obrázku, μ je průměrná hodnota jasů, δ značí střední kvadratickou odchylku, δ_{xy} představuje kovarianci. Pokud by se jmenovatel blížil k nule dojde ke stabilizaci výpočtu pomocí konstant C_1 a C_2 . Počet pixelů v obrázku je zastoupen symbolem N , x_i a y_i zastupují hodnoty konkrétních bodů v obrázku. Průměrné hodnoty jasových koeficientů pro jednotlivé obrázky zastupují koeficienty μ_x a μ_y . Rozsah hodnot jednotlivých pixelů je zastoupen symbolem L , který je při 8-bitové barevné hloubce roven 255. Konstanty K_1 a K_2 musí být mnohem menší než jedna jejich velikost bývá volena takto $K_1 = 0,01$ a $K_2 = 0,03$ [49, 50, 51]. Implementace této metody byla umístěna do adresáře „mereni_kvality“ pod názvem „ssim_custom.m“, avšak ve skriptech byla využita funkce naimplementovaná přímo v matlabu, jelikož je komplexněji naimplementovaná a udává přesnější výsledky. Závislost velikosti indexu strukturální podobnosti na kvalitativním faktoru znázorňuje obrázek 1.4.



(a) jednoduchý obrázek

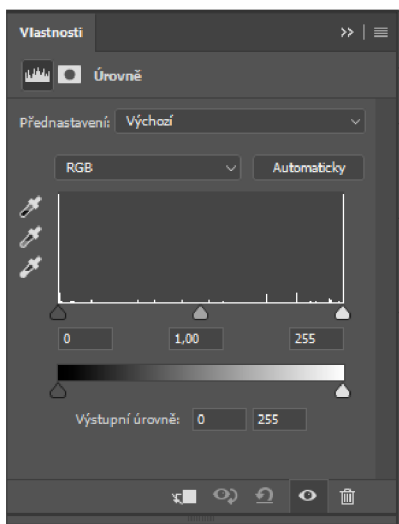


(b) fotografie

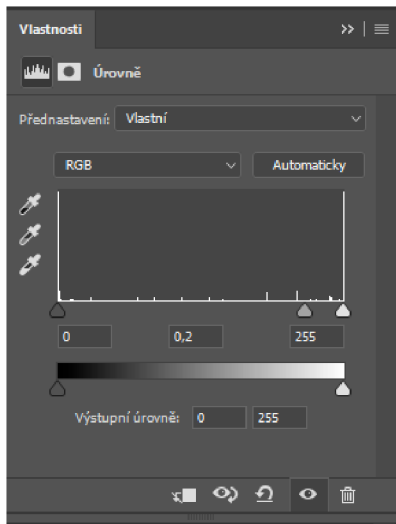
Obr. 1.4: Sledování závislosti hodnoty SSIM na zvolené kvalitě. Vytvořeno pomocí funkce s názvem „demo_SSIM_graf.m“ uložené v adresáři vytvořeno.

1.4 Způsob editace obrázků

Některé obrázky byly upraveny v programu Adobe Photoshop CC pro lepší čitelnost v tištěné verzi této práce. Původní obrázky jsou k nalezení v příloze A v porovnání s upravenými obrázky. Pro upravení zobrazení obrázků bylo využito změny nastavení úrovní v obrázku, čímž došlo k přesunutí šedých bodů do tmavší oblasti. Pro všechny upravené obrázky bylo využito stejné nastavení viz obrázek č. 1.5.



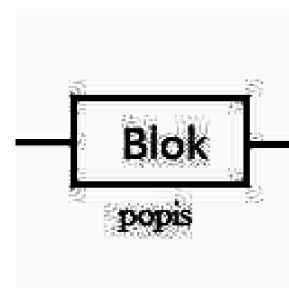
(a) prostřední posuvník nastaven na 1



(b) prostřední posuvník nastaven na 0,2



(c) originál



(d) upravený

Obr. 1.5: Zobrazení způsobu úpravy obrázků.

2 JPEG

Hlavní doménou této práce bude metoda komprese Joint Photographic Experts Group (JPEG). Tato metoda bude v této kapitole dopodrobna rozebrána. Do rodiny JPEG patří ještě JBIG a JPEG2000, někdy označovaný jako JPEG2K. JBIG byl navržen pro kompresi binárních obrázků, byl využíván ve faxech. JPEG2000 pracuje na principu využití vlnkové transformace. Zkratkou JFIF (JPEG File Interchange Format) je označen standard zabývající se uložením dat komprimovaných pomocí JPEG do souboru. Možné přípony jsou .jpg, .jpeg, .jpe, .jif, .jfif a .jfi. Algoritmus kodéru je stanoven v doporučení ITU-T Rec. T.81:1992. V tomto doporučení je však definován pouze kodér, jak u těchto standardů bývá zvykem implementace dekodéru není definována [2, 6].

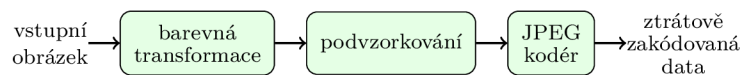
Tab. 2.1: Historie JPEG [1, 2, 5]

| | | |
|------|--------|--|
| 1982 |● | Kmenování týmu odborníků na fotografie pod organizací ISO. |
| 1987 |● | Navrženo 12 metod komprese. |
| 1989 |● | Definování základního schématu procesu kódování: DCT → Huffmanovo kódování. |
| 1990 |● | Dokončena první verze JPEG rev. 8. |
| 1992 |● | Konečné schválení IS 10918-1. |
| 1994 |● | Dokončení návrhu IS 10918-3. |
| 1996 |● | Definovány možnosti rozšíření. |
| 1998 |● | Přidány JPEG profily. |
| 2004 |● | Nový obrazový formát JPEG2000 - využívající vlnkovou transformaci. |
| 2011 |● | JPEG File Interchange Format (JFIF) IS 10918-5. |
| 2012 |● | Specifikace nástrojů a vlastností pro tisk IS 10918-6. |

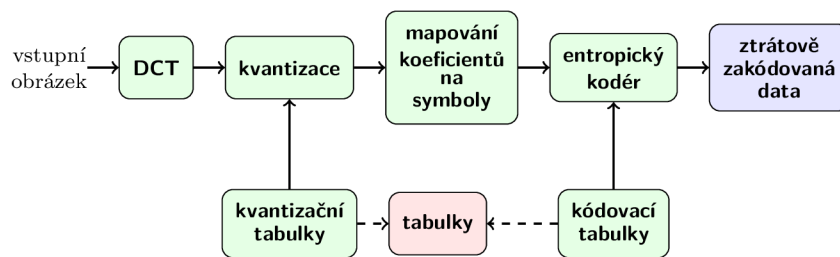
2.1 Struktura ztrátového JPEG kodéru

Zde bude do detailu rozebrán kodér ztrátového JPEG. Na obrázku 2.1 byla zobrazena nejběžnější realizace JPEG kodéru. Tato implementace zahrnuje barevnou transformaci z RGB do $YCbCr$ a není zde opomenuto ani podvzorkování chrominancních složek obrázku. Výsledek těchto operací vstupuje do JPEG kodéru jehož

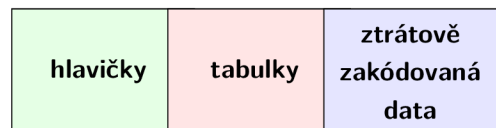
strukturu znázorňuje obrázek 2.2. Nejprve je provedeno rozdělení jednotlivých složek na subbloky o velikosti 8×8 px. Tyto bloky jsou poté podrobeny dopředné 2D-DCT transformaci, tyto subbloky jsou dále znásobeny s kvantizačními maticemi pro příslušné složky a nastavení kvality. Dále je provedeno mapování koeficientů na symboly. Výstup z tohoto bloku je dále zpracován v entropickém kodéru, kde dojde k zakódování vstupní informace. Výsledkem průchodu všech těchto bloků jsou ztrátově zakódovaná data. Strukturu JPEG souboru naznačuje obrázek 2.3, ze kterého je patrné, že společně s přenášenými ztrátově zakódovanými daty jsou přenášeny, jak kvantizační, tak Huffmanovy tabulky. V následujícím textu budou podrobně popsány jednotlivé bloky.



Obr. 2.1: Nejběžnější implementace JPEG kodéru [2]



Obr. 2.2: JPEG kodér podle standardu [2, 3, 4]



Obr. 2.3: JPEG struktura souboru [2]

2.1.1 Transformace barev

V tomto bloku nejčastěji dochází k převodu z RGB do $YCbCr$ dle vztahu [2, 3]:

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

Pro převod z YC_bC_r do RGB [3, 7]:

$$\begin{aligned}R &= Y + 1,402 \cdot (C_r - 128) \\G &= Y - 0,34414 \cdot (C_b - 128) - 0,71414 \cdot (C_r - 128) \\B &= Y + 1,772 \cdot (C_b - 128)\end{aligned}\tag{2.2}$$

Pomocí výše zmíněných převodních vztahů je docíleno toho, že se hodnoty budou nacházet v rozmezí 0–255. Pokud bude uvažován pouze samotný převod, tak jde o bezetrátový proces. V praktické aplikaci však dochází k malým zaokrouhlovacím chybám, které je možno rozdělit na chyby vzniklé vlivem aritmetických výpočtů a dále nesmí být opomenuty ani chyby vzniklé zaokrouhlením výsledků přepočtu na celá čísla. V samotné transformaci je naloženo jinak s chrominančními složkami C_b a C_r a jinak se složkou luminanční neboli Y [2, 3].

2.1.2 Podvzorkování

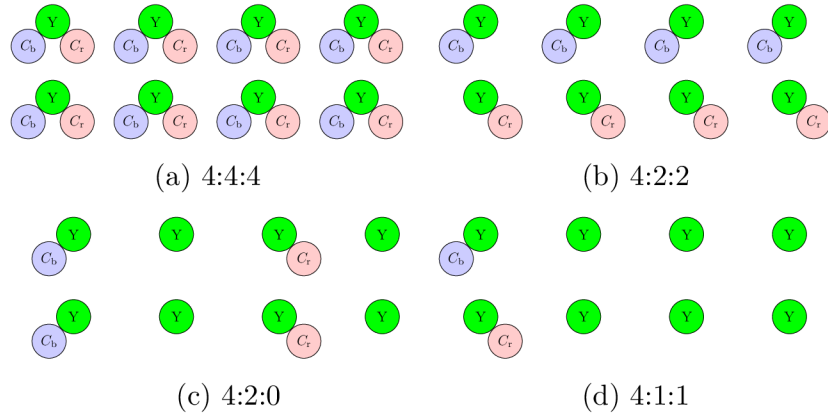
Z vlastností lidského oka vyplývá, že je mnohem citlivější na změnu jasu, protože tyčinky jsou mnohem citlivější než čípky. Naopak méně citlivé je lidské oko na barevnost, proto může být určitá část zredukována [1]. Toho docílíme tím, že část barevné informace vypustíme. Odstraněním části barevných složek nedochází k výraznému snížení kvality výsledného obrazu. Možností jakým způsobem toho docílit existuje mnoho [8].

- **Model 4:4:4** – u tohoto modelu k žádným změnám nedochází.
- **Model 4:2:2** – jasová složka zůstane nezměněná, avšak dojde k podvzorkování horizontálních chrominančních složek na polovinu, tzn. v řádku je vždy sdílena jedna barevná hodnota.
- **Model 4:2:0** – zde je opět luminanční složka bez změny, chrominanční složky jsou podvzorkovány ve vertikálním i horizontálním rozlišení na polovinu, tzn. 4 jasové složky sdílí jednu chrominanční hodnotu, střídá se C_b a C_r .
- **Model 4:1:1** – luminanční složka zůstává beze změny a barvonosné složky jsou podvzorkovány na čtvrtinu v řádcích znamená to, že 4 po sobě jdoucí jasové složky sdílí jednu chrominanční hodnotu [9].

2.1.3 Dopředná DCT

Hodnoty jasových i barevných složek jsou sníženy o 128 pro vzorky o velikosti 8 bitů, pokud se jedná o obrázek v 12bitové barevné hloubce, pak jsou jednotlivé vzorky zmenšeny o hodnotu 2048 [2].

Luminanční složka obrazu je rozdělena na dílky o velikosti 8×8 px, které jsou následně zpracovány samostatně. Chrominanční složky mají velikosti dílků různé



Obr. 2.4: Obrázky znázorňují různé typy podvzorkování [9].



Obr. 2.5: Barevná transformace z RGB do YC_bC_r s podvzorkováním 4:2:2. Vlevo se nachází původní obrázek 2.5a, následovaný luminanční složkou 2.5b a složkami chrominančními 2.5c a 2.5d.

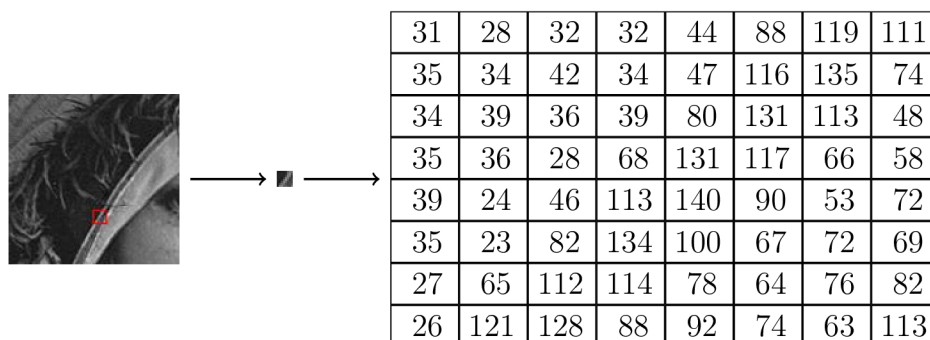
v závislosti na použitém podvzorkování. Například při režimu 4:2:0 jsou oblasti velké 8×8 px, avšak jedná se o suboblasti, jelikož ve skutečnosti reprezentují blok o velikosti 16×16 px z původního obrázku. Při použití podvzorkování 4:2:2 jsou bloky velké 16×8 px. Pokud velikost obrazu neodpovídá násobkům 8 je okraj doplněn tak, aby nebyla narušena hodnota stejnosměrné složky. Tento problém je řešen buď opakováním poslední hodnoty nebo pomocí ozrcadlení obrazových bodů. S každým z takto vytvořených bloků je následně provedena 2D dopředná DCT [2, 3]:

$$DCT[m, n] = \frac{C[i]C[j]}{\sqrt{2N}} \sum_{i=0}^7 \sum_{j=0}^7 pixel[i, j] \cos \frac{(2m+1)i\pi}{2N} \cos \frac{(2n+1)j\pi}{2N}, \quad (2.3)$$

kde m značí řádek, n značí sloupec, $N = 8$, $0 \leq i, j \leq 7$ a pro $C[a]$ platí [3]:

$$C[a] = \begin{cases} \frac{1}{\sqrt{2}}, & a = 0, \\ 1, & 1 \leq a \leq 7. \end{cases} \quad (2.4)$$

Výsledkem této operace je získání 1 DC elementu a 63 AC elementů. Tyto členy jsou uspořádané v bloku 8×8 px. Všechny dílčí bloky jsou uspořádány do matice D, dle oblastí odkud byly vytvořeny [2, 3, 4, 10].



Obr. 2.6: Dekompozice obrázku. Vlevo se nachází výřez jasové složky obrázku, uprostřed je blok velký 8×8 px ze složky Y a vpravo jsou hodnoty jednotlivých koeficientů tohoto konkrétního bloku získané pomocí matlabu.

2.1.4 Kvantizace

Při kvantizaci jsou hodnoty, které se nacházejí v matici D složku po složce vyděleny koeficienty z matice Q a následně zaokrouhleny na celá čísla [2]:

$$d^Q[k_x, k_y] = \text{round}\left(\frac{d[k_x, k_y]}{q[k_x, k_y]}\right); k_x, k_y \in \{1, \dots, 8\}, \quad (2.5)$$

tímto je dosaženo potlačení/vynulování nevýznamných koeficientů a ponechání koeficientů významných. Čím větší je hodnota koeficientu $q[k_x, k_y]$, tím větší je možnost, že bude $d^Q[k_x, k_y] = 0$. Nastavení kvantizačních matic je provedeno pomocí volby úrovně kvality zvolením hodnoty koeficientu $Q \in \{1, \dots, 100\}$, kdy dojde buď k přepočtu výchozích matic nebo k výběru jiných. Nastavení kvality na hodnotu 0 zde není uvažováno, jelikož je využíváno metody škálování kvantizačních matic a v případě, že by byla hodnota kvalitativního faktoru nastavena na nulu, došlo by při změně rozlišení kvantizačních matic k dělení nulou. Kvantizační matice jsou různé pro lumenanční složku a chrominanční složky. Kvantizační tabulky jsou součástí zkomprimovaného souboru, díky tomu je zde volnost volby tabulek. Tabulky dané standardem byly odvozeny empiricky a jsou určeny komisí JPEG [2, 3, 4].

Kvantizační tabulky

Kvantizační tabulky jsou různé pro jasovou a pro barvonosné složky obrazu [3]. Kvantizační tabulka pro lumenanční složku se nachází na obrázku 2.7a a pro chromi-

nanční složky je uvedena na obrázku 2.7b. Je to způsobeno vlastnostmi lidského oka zmíněnými v podkapitole 2.1.2.

| | | | | | | | | | | | | | | | |
|----|----|----|----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|
| 16 | 11 | 10 | 16 | 24 | 40 | 51 | 61 | 17 | 18 | 24 | 47 | 99 | 99 | 99 | 99 |
| 12 | 12 | 14 | 19 | 26 | 58 | 60 | 55 | 18 | 21 | 26 | 66 | 99 | 99 | 99 | 99 |
| 14 | 13 | 16 | 24 | 40 | 57 | 69 | 56 | 24 | 26 | 56 | 99 | 99 | 99 | 99 | 99 |
| 14 | 17 | 22 | 29 | 51 | 87 | 80 | 62 | 47 | 66 | 99 | 99 | 99 | 99 | 99 | 99 |
| 18 | 22 | 37 | 56 | 68 | 109 | 103 | 77 | 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |
| 24 | 35 | 55 | 64 | 81 | 104 | 113 | 92 | 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |
| 49 | 64 | 78 | 87 | 103 | 121 | 120 | 101 | 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |
| 72 | 92 | 95 | 98 | 112 | 100 | 103 | 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |

(a) Luminanční složka

(b) Chrominanční složka

Obr. 2.7: Výše jsou uvedeny doporučené kvantizační tabulky pro jasovou složku 2.7a a pro barvosné složky 2.7b [3, 10, 11].

Návrh kvantizačních tabulek

Se ztrátovou kompresí, je spojeno i zkreslení v obraze, které podstatně ovlivňuje počet bitů použitých k zakódování. Čím více bitů je využito, tím menší je výsledné zkreslení. Při využití ztrátové komprese jde přirozeně o snahu použít co nejmenší počet bitů s požadavkem zachovat co nejlepší kvalitu obrazu a nízkého zkreslení. Podobně to platí i pro zvolenou úroveň zkreslení (kvality). Klademe důraz na zakódování obrazu s co nejmenším možným bitovým tokem za udržení co nejlepší možné kvality. Proto byla do JPEG standardu zakomponována možnost zvolení požadované kvality výsledného obrazu, kde dochází k přepočtu kvantizačních tabulek, podle zvolené úrovně kvality dle následujících vzorců [3]:

$$\text{Rozlišovací faktor} = \begin{cases} \frac{5000}{Q} & \text{pro } 1 \leq Q \leq 50, \\ 200 - 2 \cdot Q & \text{pro } 50 \leq Q \leq 99, \\ 1 & \text{pro } Q = 100. \end{cases} \quad (2.6)$$

U těchto výpočtů je důležité poznamenat, že může nastat situace, kdy bude výsledek nulový, což je však pro další využití kvantizačních tabulek nežádoucí, jelikož by došlo ve vztahu 2.5 k dělení nulou. Tento problém lze jednoduše ošetřit například podmínkou pokud je hodnota rovna 0, dojde ke změně na 1, čímž daný problém odstraníme [1, 2, 3].

2.1.5 Mapování koeficientů na symboly

Tato jednotka většinou bývá přímo součástí entropického kodéru, zde byla vyjmuta, aby byla zdůrazněna její existence a důležitost. Zkvantované koeficienty po DCT

jsou namapovány na nové symboly k usnadnění práce entropického kodéru. AC koeficienty jsou efektivně reprezentovány pomocí shluku nul a jsou ukončeny nenulovým ukončovacím koeficientem. DC koeficienty jsou definovány tím, že je vypočten rozdíl mezi DC koeficientem v aktuálním bloku a v bloku předešlém [3].

2.1.6 Entropický kodér

Tato jednotka přidává kódové slovo k symbolům, které se objeví na jejím vstupu. Z kódových slov je následně vytvořen bitový tok (bitstream). Tento bitový tok je následně buď přenášen dál nebo uložen. V tomto bloku je nejčastěji využíváno Huffmanovo kódování s proměnnou délkou (variable length coding) kódového slova. Je zde možné využít i aritmetické kódování [3].

Kódování DC koeficientů

Využívá se vlastnosti, že přilehlé bloky bývají vysoce korelované. Této vlastnosti je využíváno při rozdílovém kódování sousedních bloků. Necht' je $qX_i[0,0]$ a bod v v sousedním bloku $qX_{i-1}[0,0]$ popisují stejnosměrný koeficient uložený v bloku. Rozdíl mezi hodnotami koeficientů je vypočten pomocí [3]:

$$\delta_i = qX_i[0,0] - qX_{i-1}[0,0]. \quad (2.7)$$

DC koeficienty δ_i mohou nabývat hodnot $[-2047, 2047]$, pokud budeme uvažovat přesnost 8 bitů na pixel pro každou obrazovou složku. Nejběžněji je využito Huffmanovo kódování, které může dosahovat velmi rozsáhlých kódovacích tabulek, proto byly kódovací tabulky omezeny. Omezení bylo docíleno rozdělením hodnot do 12 skupin. Tyto skupiny jsou označeny číslem 0 až 11. Tyto kategorie jsou zakódovány pomocí Huffmanova kódování. K rozlišení hodnot z různých kategorií je využito k přídavných bitů. Tyto bity jsou využity k reprezentaci jedné z 2^k hodnot ze symbolů v rámci dané kategorie. Velikost hodnot δ_i může být v rámci kategorie $\{2^{k-1} \leq \delta_i \leq 2^k - 1\}$. Jde tedy o jejich binární vyjádření. Na druhé straně je velikost dána $\{-2^k - 1 \leq \delta_i \leq -2^{k-1}\}$. Velikost je tedy dána doplňkem k absolutní hodnotě $|\delta_i|$, nebo pomocí reprezentace $\delta_i + 2^k - 1$ [3].

Kódování AC koeficientů

Nejvíce AC koeficientů je nulových. Pomocí cik-cak vyčítání je z bloku 8×8 px vyčteno 63 koeficientů obsahujících mnoho po sobě jdoucích nul v angličtině je tento jev označován „runs of zeros“. Tento jev činí AC koeficienty vhodné k úsekovému kódování (run-length coding). Symboly jsou v tomto případě definovány jako úseky nul a nenulové ukončovací hodnoty, které mohou být entropicky kódovány. Rozsah hodnot

Tab. 2.2: Tabulka kategorií pro rozdělení DC složek při 8 bpp [2, 12, 13]

| rozsahy hodnot | kategorie | kód kat. v Huff. tabulce |
|-----------------------------|-----------|--------------------------|
| 0 | 0 | 00 |
| -1;1 | 1 | 010 |
| -3, -2;2, 3 | 2 | 011 |
| -7... -4;4... 7 | 3 | 100 |
| -15... -8;8... 15 | 4 | 101 |
| -31... -16;16... 31 | 5 | 110 |
| -63... -32;32... 63 | 6 | 1110 |
| -127... -64;64... 127 | 7 | 11110 |
| -255... -128;128... 255 | 8 | 111110 |
| -511... -256;256... 511 | 9 | 1111110 |
| -1023... -512;512... 1023 | 10 | 11111110 |
| -2047... -1024;1024... 2047 | 11 | 111111110 |

AC koeficientů pro 8 bpp je v rozmezí $[-1023, 1023]$. Možná velikost AC koeficientů s sebou nese potřebu obdobné optimalizace velikosti kódovacích tabulek jako byla provedena u DC koeficientů. Kategorie byly vytvořeny pro vhodně seskupené hodnoty. Využívá se tedy 8 bitů na symbol, který je složený ze dvou částí. Dynamický rozsah pro 8bitové vzorky je až 11 bitů. První 4 bity definují počet po sobě jdoucích nul a jsou označeny R (run) a druhá čtveřice označuje kategorii koeficientu a označuje se S (size), pak následuje označení jednotkového doplňku. Nenulové prvky jsou kódovány pomocí dvojice „(R|S)“. Pokud je hodnota délky posloupnosti nulových hodnot nastavena na 15. Symbol pak je nastaven na (15|0), což znamená 15 nul následovaných nulou tento stav se označuje zero run length. Tyto symboly mohou být skládány do kaskády, aby složily specifickou velikost posloupnosti nul. Pokud jsou za nenulovým koeficientem zbylé koeficienty nulové, pak následuje speciální symbol (0|0), který označuje konec bloku – end of block. Celkem je k dispozici 162 kombinací (R|S), jelikož R může nabývat 16 hodnot, S 10 hodnot a pak zde jsou ještě 2 speciální symboly EOB a ZRL. Náhled na standardní Huffmanovu tabulku naleznete v tabulce 2.3. V JPEG mohou být použity, jak standardní Huffmanovy tabulky, tak i tabulky přizpůsobené speciálně na míru danému obrázku – pak se jedná o „optimalizovaný JPEG“. Toto je možné udělat jelikož jsou tabulky přenášeny společně s daty [2, 3].

Tab. 2.3: Huffmanova tabulka 162 kombinací (R|S) pro jas. složku při 8 bpp [2]

| R S kombinace | kód v Huffmanově tab. |
|---------------|-----------------------|
| (0 1) | 00 |
| (0 2) | 01 |
| (0 3) | 100 |
| (0 0) | 1010 (EOB) |
| (0 4) | 1011 |
| (1 1) | 1100 |
| (0 5) | 11010 |
| (1 2) | 11011 |
| (2 1) | 11100 |
| (3 1) | 111010 |
| (4 1) | 111011 |
| (0 6) | 1111000 |
| (1 3) | 1111001 |
| (5 1) | 1111010 |
| : | |
| (13 1) | 11111111000 |
| (15 0) | 11111111001 (ZRL) |
| (2 4) | 111111110100 |
| : | |
| (15 9) | 111111111111101 |
| (15 10) | 111111111111110 |

2.2 Struktura dekodéru

Dekodér je možné stejně jako kodér rozdělit na část definovanou standardem viz obrázek 2.9 a na nejčastější implementační řešení, které je znázorněno na obrázku 2.10. Jak je možné vidět na blokových schématech, tak dekodér JPEG je složen z entropického dekodéru, který má za úkol dekodovat vstupní data a přidělit konkrétní shluky symbolů zpátky k koeficientům. Výsledek této operace je znásoben s kvantizačními tabulkami, následně je provedena inverzní 2D-DCT, čímž dojde k rekonstrukci dat v daném bloku, avšak obnovená data jsou již se zkreslením vlivem kvantizace. Na výstup z této části je aplikováno nadvzorkování chrominančních složek, pokud byly předtím podvzorkovány. Jakmile jsou všechny složky obrazu v původní velikosti provede se baravná transformace zpět do původního barevného prostoru. V námi uvažovaném případě jde o transformaci z $YCbCr$ zpátky do RGB. V následujícím

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 1 | 2 | 6 | 7 | 15 | 16 | 28 | 29 |
| 3 | 5 | 8 | 14 | 17 | 27 | 30 | 43 |
| 4 | 9 | 13 | 18 | 26 | 31 | 42 | 44 |
| 10 | 12 | 19 | 25 | 32 | 41 | 45 | 54 |
| 11 | 20 | 24 | 33 | 40 | 46 | 53 | 55 |
| 21 | 23 | 34 | 39 | 47 | 52 | 56 | 61 |
| 22 | 35 | 38 | 48 | 51 | 57 | 60 | 62 |
| 36 | 37 | 49 | 50 | 58 | 59 | 63 | 64 |

Obr. 2.8: Cik-cak vyčítání koeficientů [2, 3, 10]

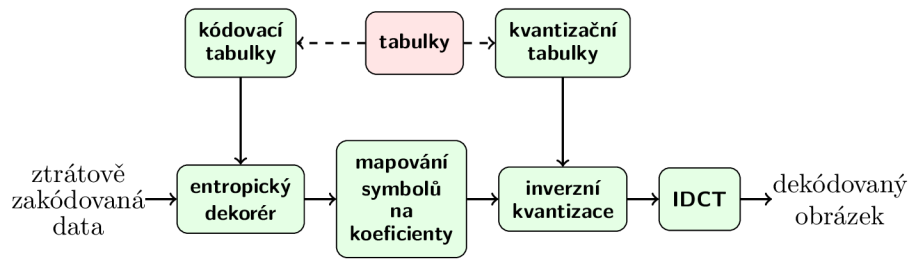
Tab. 2.4: Tabulka kategorií pro rozdělení AC složek při 8 bpp [2, 12, 13]

| rozsahy hodnot | kategorie |
|--------------------------|-----------|
| -1;1 | 1 |
| -3, -2;2, 3 | 2 |
| -7... -4;4...7 | 3 |
| -15... -8;8...15 | 4 |
| -31... -16;16...31 | 5 |
| -63... -32;32...63 | 6 |
| -127... -64;64...127 | 7 |
| -255... -128;128...255 | 8 |
| -511... -256;256...511 | 9 |
| -1023... -512;512...1023 | 10 |

textu budou popsány jednotlivé části JPEG dekodéru.

2.2.1 Entropický dekodér

V dekodéru je provedena inverzní operace vzhledem ke kodéru. Zakódovaný bitový tok obsahuje kódovací a kvantizační tabulky, které jsou prvně vyjmuty. Zakódovaná data následně vstoupí do dekodéru, kde dojde k rozpoznání zakódovaných symbolů. Symboly jsou poté namapovány do matice kvantizovaných DCT koeficientů, které jsou následně „dekvantizovány“. Je toho docíleno tak, že každý koeficient je znásoben s odpovídajícím koeficientem v kvantizační tabulce. Dekódovaný obrázek je poté získán aplikací inverzní dvojrozměrné DCT na matici obnovených DCT koeficientů pro každý blok obrázku [3].



Obr. 2.9: JPEG dekodér podle standardu [2, 3, 4]



Obr. 2.10: Nejběžnější implementace JPEG dekodéru [2]

2.2.2 Mapování symbolů na koeficienty

Tento blok bývá velmi často přímo součástí entropického dekodéru je zde uveden pro zdůraznění potřeby převodu binární hodnoty zpět na nějakou dekadickou. V praxi se však nikdy nedostaneme zpět k původní hodnotě, ale například vlivem kvantizace zde dostáváme mírně zaokrouhlenou hodnotu. To však pro lidské oči nehraje roli, jelikož drobné niance nejsme schopni rozlišit [3].

2.2.3 Inverzní kvantizace

V této části dochází k násobení bloků 8×8 px s příslušnými kvantizačními maticemi, tento proces je naznačen rovnicí 2.8 [3].

$$d^Q[k_x, k_y] = d[k_x, k_y]q[k_x, k_y]; k_x, k_y \in \{1, \dots, 8\} \quad (2.8)$$

2.2.4 Inverzní DCT

„Originální“ obrázek je sestaven z bloků 8×8 px, na které je aplikována inverzní DCT, která má následující výpočet [3]:

$$pixel[m, n] = \sum_{i=0}^7 \sum_{j=0}^7 \frac{C[i]C[j]}{\sqrt{2N}} DCT[i, j] \cos \frac{(2m+1)i\pi}{2N} \cos \frac{(2n+1)j\pi}{2N}, \quad (2.9)$$

kde m značí řádek, n značí sloupec, $N = 8$, $0 \leq i, j \leq 7$ [3].

2.2.5 Nadvzorkování

V této části je provedeno nadvzorkování chrominančních složek na původní velikost v případě, že byly předtím podvzorkovány.

2.2.6 Zpětná barevná transformace

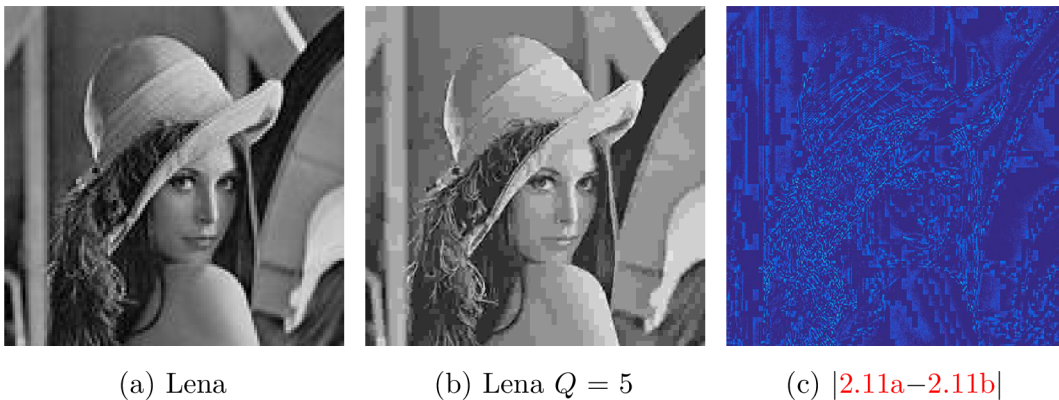
V tomto bloku dochází ke zpětné barvené transformaci zpátky do původního barevného modelu. Pro nejběžnější implementace to tedy znamená převod z $YCbCr$ do RGB. K tomuto by byly využity již dříve zmíněné vztahy 2.2.

2.3 Obrazové artefakty

Tato podkapitola se zabývá vybranými zápornými vlastnostmi JPEG komprese. Nejprve budou rozebrány blokové artefakty, na ty bude navázáno barevným zkreslením. Dále budou uvedeny artefakty oblastí a posledním uvedeným jevem bude rozmazání.

2.3.1 Blokové artefakty

Tyto artefakty vznikají, protože základní JPEG může pracovat pouze s jedním subblokem velikosti 8×8 px ve stejnou chvíli, což způsobuje artefakty v zakódovaných obrázcích. Artefakty mají tvar čtverečků a mohou být po celém rekonstruovaném obrázku. K tomuto narušení obrazu dochází hlavně při nastavení extrémně nízké kvality obrazu, a to vlivem ztráty harmonických složek z původního signálu. Ukázkou blokových artefaktů naleznete na obrázku 2.11 [14].



Obr. 2.11: Zobrazení rozdílu mezi originálem a obrázkem s degradovanou kvalitou. Na obrázku 2.11a je zobrazen originální obrázek, obrázek 2.11b obsahuje originál zakódovaný pomocí JPEG s $Q = 5$. Poslední obrázek 2.11c reprezentuje rozdíl v absolutní hodnotě mezi originálem a komprimovaným obrázkem, aby lépe vynikly vzniklé rozdíly byla pro vykreslení využita funkce `imagesc` z `matlabu`.

2.3.2 Barevné zkreslení

Jelikož jsou lidské oči nejvíce citlivé na jas, tak je pro ně velice obtížné postřehnout, že došlo k drobné barevné odchylce. K těmto zkreslením dochází vlivem podvzorkování chrominančních složek obrazu. V závislosti na velikosti podvzorkování je pak výsledný obrázek barevně zkreslený [14]. Tento jev je hezky viditelný na příkladech uvedených v [9, 14].

2.3.3 Artefakty v okolí ostrých přechodů

Protože JPEG pracuje ve spektrální oblasti, což znamená, že je pracováno s obrázkem jako se sumou hladkých oscilujících vlnek. Právě spektrální oblast je vhodná k reprezentaci jemných barevných přechodů, ale není vhodná k reprezentaci hran. Proto dochází v oblastech s rychlými přechody k artefaktům. Je to způsobeno vlivem aliasingu. Vhodným příkladem kde dochází k těmto jevům je obraz s ostrými hranami, obzvláště se tento jev projevuje u textu, blokových diagramů a čar. Ukázka uvedeného jevu se nachází na obrázku 2.12. V anglické literatuře jsou tyto artefakty označovány jako „ringing“ [14].

2.3.4 Rozmazání

Nežádáný efekt rozmazání znamená, že dekódovaný obrázek není tak ostrý jako byl obrázek původní. Tento problém vzniká vlivem snižování kvality, kdy jsou postupně odstraňovány vysoko frekvenční složky v obraze, čímž dojde ke ztrátě informace o hranách. Zajímavé je, že tohoto jevu si lidé všimnou narozdíl od barevného zkreslení snadno [14].

2.4 Alternativní módy

V předcházejícím textu byl popsán JPEG v sekvenčním DCT módu, který je zároveň nejrozšířenějším. Avšak tento mód není jediným možným, standard JPEG definuje i progresivní DCT mód, bezztrátový mód a hierarchický mód. V následujícím textu budou tyto módy představeny.

2.4.1 Progresivní mód

Výhodou progresivního módu je to, že není potřeba projít celý obrázek např. pro získání jeho náhledu. U JPEG v progresivním módu stačí už cca jedna desetina celkové velikosti souboru k zobrazení slušné úrovně kvality náhledu. Nevýhodou tohoto módu je menší komprimační poměr. Další nevýhodou byla potřeba rychlé

operační paměti s dostatečnou kapacitou, avšak toto již bude v dnešní době přežitkem, jelikož osobní počítače mívají standardně 4 GB RAM a více. Mezi další nevýhody je možné zahrnout fakt, že obrázek musí být dekodován několikrát po sobě, aby byl získán obrázek v plném rozlišení. Je to způsobeno tím, že nejprve jsou z obrázku vyčteny DC koeficienty dílčích oblastí, následně dojde k vyčtení dalších detailů. Tento proces se může opakovat až 8krát [3, 15].

2.4.2 Hierarchický mód

Tento mód je ve své podstatě jinak definovaný progresivní mód. Obrázek je zde rozložen pomocí pyramidové struktury, kde nejvyšší úroveň představuje nejmenší rozlišení obrázku (náhledu – teoreticky 1 px) a nejspodnější úroveň představuje obrázek v plném rozlišení. Dochází zde k dělení vertikálního a horizontálního rozlišení, mezi úrovněmi pyramidy. Hierarchický mód je vhodný k zobrazení obrázku v různých rozlišeních, například v chytrém telefonu s nízkým rozlišením displeje není nutné stahovat obrázky v plné kvalitě, ale je možné využít jen určitou postačující kvalitu. Této výhody lze využít např. u webových stránek s responzivním designem, kde jsou vytvořeny kvalitní textury pro plné rozlišení a pokud je rozlišení menší, tak je zobrazena právě jedna z nižších úrovní hierarchické struktury [3].

2.4.3 Bezeztrátový mód

JPEG umožňuje pracovat i v bezeztrátovém módu, pokud jde o JPEG v tomto módu bývá označován též JPEG-LS. Tento mód je definován v ISO-14495-1/ITU-T.87. Bezeztrátový mód umožňuje pracovat i v „téměř“ bezeztrátovém módu. Jelikož se tato práce věnuje ztrátovému JPEG, nebude tento mód dále rozebírán, protože jde o velice komplexní problematiku, která svým rozsahem spadá mimo oblast této práce [2, 16].

2.5 Shrnutí

V této kapitole byl podrobně rozebrán ztrátový JPEG a to jak z pohledu kodéru tak dekodéru. Byly zde zdůrazněny některé artefakty, které při kompresi pomocí JPEG vznikají. Nebyly opomenuty ani alternativní módy JPEG.

2.5.1 Výhody

Mezi hlavní výhody JPEG patří to, že ukládá barvenou informaci v rozsahu 24 bpp. Nejčastěji se využívá k ukládání fotografií s hladkým barevným přechodem. Dosahuje

většího kompresního poměru než paletové formáty GIF a PNG. Umožňuje nastavení kvality, čímž se ovlivní i kompresní poměr. Dovoluje nastavení podvzorkování chrominančních složek. Jde o světově rozšířený standard.

2.5.2 Nevýhody

Za hlavní nevýhodu lze považovat to, že při každém přeuložení obrázku dochází k degradaci jeho kvality vlivem kvantizačních tabulek. JPEG není vhodný ke kompresi obrázků obsahujících ostré hrany. Na rozdíl od uvedeného paletového formátu PNG, JPEG neumožňuje využívat alfa kanál, tedy nedovoluje u obrázků průhlednost. Další nevýhody, které JPEG zahrnuje byly zmíněny v podkapitole [2.3](#).



Obr. 2.12: Zobrazení rozdílu mezi originálem a degradovaným obrázkem. V prvním sloupečku lze nalézt originální obrázky, ve druhém sloupci se nachází originální obrázky zakódované pomocí JPEG s nastavenou kvalitou Q na hodnotu 50. Poslední sloupec reprezentuje rozdíl v absolutní hodnotě mezi originálem a komprimovaným obrázkem. Pro lepší viditelnost rozdílů byla pro vykreslení využita funkce `imagesc` z `matlabu`. Vytvořeno pomocí funkce s názvem „demo_zobrazeni_rozdilu_mezi_originalem_a_degradovanim.m“ uložené v adresáři vytvořeno. Obrázky 2.12b, 2.12e a 2.12h byly upraveny v aplikaci Adobe Photoshop pro lepší viditelnost vzniklých artefaktů v tištěné verzi této práce. Porovnání původního obrázku s upraveným znázorňuje obrázek A.1.

3 ŘÍDKÉ REPREZENTACE SIGNÁLŮ

Nejlepší definice co to vlastně je řídká reprezentace signálů byla stanovena v [17] a zní „Název řídká reprezentace znamená, že daný signál je možné vyjádřit přesně či velmi dobře aproximovat lineární kombinací velmi malého počtu vektorů ze zvoleného reprezentačního systému.“.

3.1 Základní pojmy a značení

Kardinalita¹ bude značena pomocí absolutní hodnoty tj. $|\{-2, -1, 5, 9, 12\}| = 5$. Skalární veličiny budou v textu označeny kurzívou např. a , B . K rozlišení komplexních čísel bude využít proužek nad označením, pak tedy pokud c bude komplexní číslo v textu bude k nalezení označené takto \bar{c} . Vektory v textu budou rozlišeny pomocí tučného řezu písma malého písmena, jako \mathbf{a} , φ atd. Dále jsou vektory uvažovány jako sloupcové konečnorozměrné, pokud v textu nebude uvedeno jinak. Matice budou v textu označovány rovněž tučným řezem písma, ale pomocí kapitálek např. \mathbf{M} , Φ . K jednotlivým prvkům matic poté bude přistupováno pomocí příslušného malého písmene shodného s označením matice, např. $m_{i,j}$, kde i označuje zvolený řádek a j označuje příslušný sloupec. Konkrétní řádek i matice \mathbf{M} bude zastoupen symbolem $m_{i,:}$. Obdobně to bude platit i pro konkrétní j sloupec $m_{:,j}$. Pokud bude matice označena symbolem $*$, pak se bude jednat o tzv. *hermitovskou transpozici*, která vznikla pomocí transpozice matice \mathbf{M} a komplexním sdružením všech jejích prvků [17, 18].

Definice 3.1. Nosič vektoru představuje množinu indexů tohoto vektoru, u kterých má vektor nenulové hodnoty. Tato množina se označuje $\text{supp}(\mathbf{x})$, pak tedy $\text{supp}(\mathbf{x}) = \{i | x_i \neq 0\}$ [17].

Uvážíme-li například signál $\mathbf{x} = [x_1, \dots, x_8]^T = [1, 0, 4, 0, 5, 2, 0, 7]^T$. Pak tedy hodnota $\text{supp}(\mathbf{x})$ bude odpovídat $\text{supp}(\mathbf{x}) = \{1, 3, 5, 6, 8\}$ a kardinalita nosiče vektoru je rovna $|\text{supp}(\mathbf{x})| = 5$. Pro hermitovské matice platí $(\mathbf{A}\mathbf{B}^*) = \mathbf{A}^*\mathbf{B}^*$. Jádro lineárního zobrazení bude označováno $\ker \mathbf{A}$ a hodnota matice bude označena $\text{rank } \mathbf{A}$. Pro přeskládání matic do sloupcového tvaru bude sloužit operátor $\text{vec}(\cdot)$, tento operátor způsobí přeskládání sloupců matice postupně podsebe. Obdobně to bude platit i při aplikaci na vícerozměrný tenzor [17].

Pomocí tohoto řezu písma \mathbb{V} , \mathbb{R}^n , \mathbb{C}^n budou v textu označeny vektorové prostory. Jakmile bude někde uveden vektorový prostor bude uvažován jako neprázdný. Generátory vektorových prostorů včetně bází budou označeny pomocí tučných kapitálek

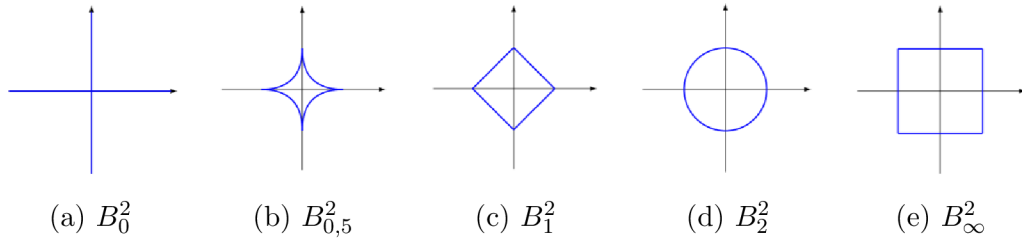
¹Kardinalita – počet prvků množiny.

A, B, C atd., jednotková matice bude značena vždy pomocí **I**. Norma vektoru je hodnota, která pomocí jediného čísla vyjadřuje velikost vektoru [17, 18].

Definice 3.2. Stanovení l_p -normy vektoru $\mathbf{x} \in \mathbb{C}^N$ je formulováno [17]:

$$\begin{aligned}\|\mathbf{x}\|_p &:= \left(\sum_{i=0}^N |x_i|^p \right)^{\frac{1}{p}} \quad \text{pro } 1 \leq p < \infty, \\ \|\mathbf{x}\|_p &:= \sum_{i=0}^N |x_i|^p \quad \text{pro } 0 < p < 1, \\ \|\mathbf{x}\|_\infty &:= \max_i |x_i|, \\ \|\mathbf{x}\|_0 &:= |\text{supp}(\mathbf{x})|.\end{aligned}\tag{3.1}$$

O normu se z matematického hlediska jedná pouze v případě, že $1 \leq p < \infty$, avšak pro zjednodušení bude použito pro všechna p stejné označení tedy l_p -norma. Sumu absolutních hodnot jednotlivých prvků vektoru vyjadřuje l_1 -norma $\|\cdot\|_1 = \|\mathbf{x}\|_1 = \sum_i |x_i|$. Norma l_0 vyjadřuje počet prvků vektoru různých od nuly. Jestliže se v textu objeví $\|\cdot\|$ bude se jednat o tzv. *euklidovskou normu* $\|\mathbf{x}\|_2 = \sqrt{\sum_i |x_i|^2}$. Pomocí zobrazení jednotkových koulí v jednotlivých normách bude snadnější porozumět činnosti norm. Na obrázku 3.1 jsou znázorněny hranice jednotkových koulí pro různé normy [17].



Obr. 3.1: Znázornění jednotkových koulí [17].

Definice 3.3. Jednotková koule B_p^N je v normě l_p definována jako [17]:

$$B_p^N := \{\mathbf{x} \in \mathbb{C}^N \mid \|\mathbf{x}\|_p \leq 1\}.\tag{3.2}$$

Definice 3.4. O řídký vektor se jedná právě tehdy, když pro vektor $\mathbf{x} \in \mathbb{C}^N$ platí [17]:

$$\|\mathbf{x}\|_0 \leq k.\tag{3.3}$$

Pak tedy pokud bude uvažován k -řídký (k -sparse) vektor, což je takový vektor, který má maximálně k nenulových prvků. Poměrem $\frac{k}{N}$ bude vyjádřena relativní řídkost vektoru \mathbf{x} o délce N . Množina všech k -řídkých vektorů délky N bude označena pomocí $\Sigma_k := \Sigma_k^N := \{\mathbf{x} \in \mathbb{C}^N \mid \|\mathbf{x}\|_0 \leq k\}$. Jde tedy o to nalézt takové řešení, kde bude nejvíce neznámých současně nulových, pak se jedná o tzv. *řídké řešení*.

Vzhledem k tomu, že skutečné signály nebývají řídké striktně, protože místo nulových hodnot jednotlivých nevýznamných složek jsou jejich hodnoty pouze blízké nule. Proto je pro výpočty výhodné stanovit chybu aproximace [17, 19].

Definice 3.5. Odchylka nejlepší aproximace vektoru $\mathbf{x} \in \mathbb{C}^N$ k -řídkým vektorem v normě l_p je stanovena takto [17]:

$$\sigma_k(\mathbf{x})_p := \sigma_k^N(\mathbf{x})_p := \inf_{\mathbf{z} \in \Sigma_k^N} \|\mathbf{x} - \mathbf{z}\|_p. \quad (3.4)$$

Z výše uvedené definice vyplývá, že chyba nemůže klesnout pod stanovenou hodnotu.

Pro vektor odvozený z $\mathbf{x} \in \mathbb{C}^N$ přesunutím nenulových prvků do nové množiny T , platí $T \subset \{1, \dots, N\}$. Doplněk této množiny bude v následujícím textu označen $T^c = \{1, \dots, N\} \setminus T$ [17].

Chybu aproximace lze vyjádřit jako p -normu vektoru, vzniklého z \mathbf{x} odebráním k složek s největší velikostí. Toto je možné definovat $\sigma_k(\mathbf{x})_p = \min_{T \subset \{1, \dots, N\}, |T| \leq k} \|\mathbf{x}_{T^c}\|_p$. Určení chyby se vždy odvíjí od konkrétního vektoru \mathbf{x} , proto se dá považovat tato činnost za adaptivní záležitost. Nyní bude stanovena definice diskrétní totální variace [17].

Definice 3.6. Pro jednodimenzionální vektor $\|x\|_{TV}$ je stanovena diskrétní totální variace jako [17]:

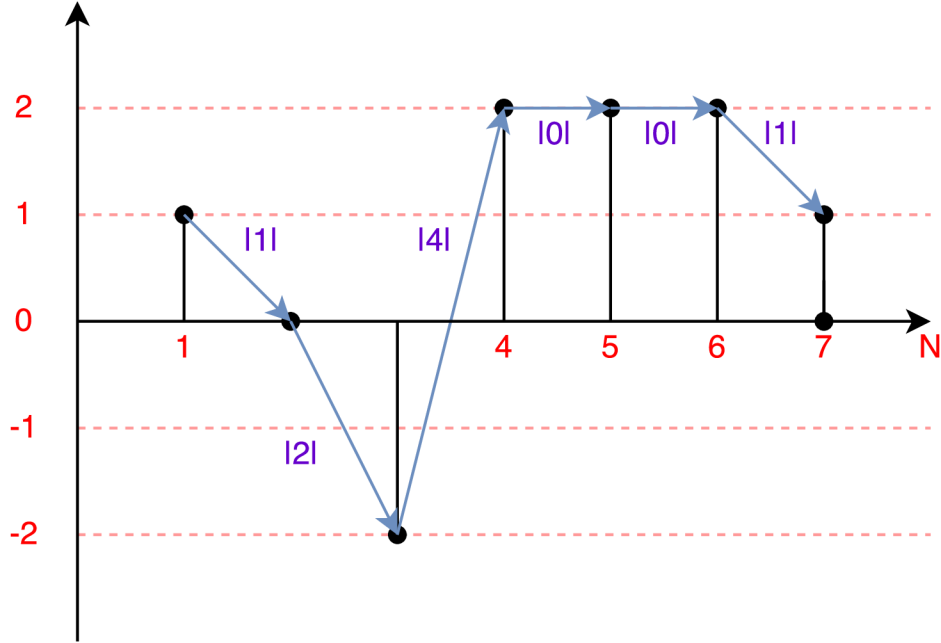
$$\|x\|_{TV} := \sum_i = 1^{n-1} |x_{i+1} - x_i|. \quad (3.5)$$

Samotná diskrétní TV sice nesplňuje jeden z axiomů normy, avšak v literatuře je často uváděna TV-norma. Vzorec 3.5 se dá představit jako celková dráha promítnutá na horizontální ose. Jde o dráhu, kterou by uběhl signál v čase [17]. Pro lepší představu toho jak TV-norma pracuje využijeme následující příklad. Např. bude uvažováno následující $\|[1, 0, -2, 2, 2, 2, 1]\|_{TV} = 1 + 2 + 4 + 0 + 0 + 1 = 8$, tento příklad je znázorněn graficky na obrázku 3.2.

TV-norma je definována i pro obrazy, pak jde o dvoudimenzionální TV-normu. Podle [17] se snadněji pracuje s tzv. *izotropní TV*, avšak lepší vlastnosti vykazuje tzv. *anizotropní TV-norma*. Obě normy mají podobný zápis jako již uvedená TV-norma. Totální variace v obraze představuje míru proměnlivosti obrazu [17].

- Konstantní obraz – nulová TV.
- Bílý šum, případně signál měnící znaménka – vysoká hodnota TV [17].

Normy je možné aplikovat i na matice, kde se pro usnadnění operací s maticemi využívá převod matice do jednoho vektoru a na výsledný vektor jsou aplikovány normy určené pro vektory. Tohoto přeskládání je možné docílit například pomocí již dříve uvedeného operátoru $\text{vec}(\cdot)$. Pokud bude v následujícím textu aplikována vektorová norma na matici, pak bude uvažován vektor vytvořený pomocí $\|\mathbf{X}\| := \|\text{vec}\mathbf{X}\|$. Předpokládáme pouze případ $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$ nebo $\mathbf{X} \in \mathbb{C}^{n_1 \times n_2}$. Jde tedy o matice, které mají n_1 řádků a n_2 sloupců [17].



Obr. 3.2: Grafické znázornění uvedeného příkladu.

Definice 3.7. Nejběžnější maticovou normou je tzv. *Frobeniova norma* $\|\cdot\|_F$, tato norma je určena jako [17]:

$$\|\mathbf{X}\|_F := \|\text{vec}\mathbf{X}\|_2 = \sqrt{\sum_i \sum_j |x_{ij}|^2}, \quad (3.6)$$

tato norma tedy vyjadřuje celkovou energii prvků matice [17].

3.2 Hledání řídkého řešení

Hledání řídkého řešení vlastně představuje řešení soustavy lineárních rovnic $\mathbf{Ax} = \mathbf{y}$. V tomto případě však jde o soustavu, která má nekonečně mnoho řešení a hledaným řešením je právě to nejřidší. Jako nejřidší je označeno takové řešení, které má nejvíce nulových složek nebo alespoň nejvíce nejmenších složek v nastavené toleranci. Hledání řídkého řešení tedy představuje úlohu [17]:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ vzhledem k } \mathbf{Ax} = \mathbf{y}, \quad (3.7)$$

kde známe hodnoty vektoru $\mathbf{y} \in \mathbb{C}^n$ a hodnoty matice $\mathbf{A} \in \mathbb{C}^{n \times N}$. Uvažovány budou pouze případy $n < N$ případně $n \ll N$ a \mathbf{A} je uvažována plnořádkové hodnoty. Matice \mathbf{A} bývá obvykle v souvislosti s řídkými reprezentacemi signálů označována jako slovník (dictionary). Sloupce této matice jsou pak označovány jako atomy (atoms). Řídké řešení bývá velmi často zatížené šumem, který způsobuje niance od přesného

řešení. Z tohoto důvodu bude ve vztahu $\mathbf{Ax} = \mathbf{y}$ povolena drobná odchylka, která je definována takto [17]:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ vzhledem k } \|\mathbf{Ax} - \mathbf{y}\|_p \leq \delta. \quad (3.8)$$

Určování řídkého řešení není jednoznačnou záležitostí. Na slovník jsou kladeny určité požadavky, aby bylo možné určit zda vůbec řídké řešení existuje. Případně zda nalezené řešení je právě to nejřidší a zda se jedná o jediné možné řešení nebo takových řešení existuje více. To jestli řešení existuje a jaké má vlastnosti udává číslo spark. Toto číslo představuje nejmenší počet lineárně závislých sloupců matice. Číslo spark tedy může být stanoveno jako [17, 19]:

$$\text{spark}(\mathbf{A}) = \min_{\mathbf{z} \in \ker \mathbf{A}, \mathbf{z} \neq \mathbf{0}} \|\mathbf{z}\|_0. \quad (3.9)$$

Z této definice vyplývá, že pokud bude matice \mathbf{A} nenulová, pak pro ní platí $\mathbf{A} \in \mathbb{C}^{n \times N}$, kde $n < N$, pro spark platí $\text{spark}(\mathbf{A}) \in \{2, \dots, n+1\}$. Pro řešení problému 3.7 může být spark využit takto [17]:

$$\|\mathbf{x}\|_0 < \frac{\text{spark}(\mathbf{A})}{2}, \quad (3.10)$$

pokud nalezené řešení bude splňovat tuto 3.10 podmínku, pak je s jistotou právě nejřidší možné a žádné další s takovou kardinalitou neexistuje. Číslo spark je možné využít i k zjišťování vlastností řešení to má však jednu podstatnou nevýhodu. Výpočetní náročnost nalezení $\text{spark}(\mathbf{A})$ je srovnatelná s nalezením samotného řešení problému 3.7, proto se v praxi této vlastnosti nevyužívá. Z tohoto důvodu byla zavedena další vlastnost matic a to vzájemná koherence (mutual coherence). Tato vlastnost je určena jako [17, 19]:

$$\mu(\mathbf{A}) = \max_{1 \leq j, k \leq N, j \neq k} \frac{|\mathbf{a}_j^T \mathbf{a}_k|}{\|\mathbf{a}_j\|_2 \cdot \|\mathbf{a}_k\|_2}, \quad (3.11)$$

kde j -tý sloupec matice \mathbf{A} je označen \mathbf{a}_j . Vztah mezi vzájemnou koherencí matice a číslem spark platí pro libovolnou matici. Tento vztah byl stanoven takto [17]:

$$\text{spark}(\mathbf{A}) \leq 1 + \frac{1}{\mu(\mathbf{A})}. \quad (3.12)$$

Jestliže je pro soustavu $\mathbf{Ax} = \mathbf{y}$ nalezeno řešení \mathbf{x} splňující [17]:

$$\|\mathbf{x}\|_0 < \frac{1}{2} \left(1 + \frac{1}{\mu(\mathbf{A})} \right), \quad (3.13)$$

pokud je tato podmínka splněna, pak je právě \mathbf{x} nejřidší možné řešení a žádné další takové neexistuje. Takové řešení je možné nalézt l_1 -minimalizací, která bude uvedena

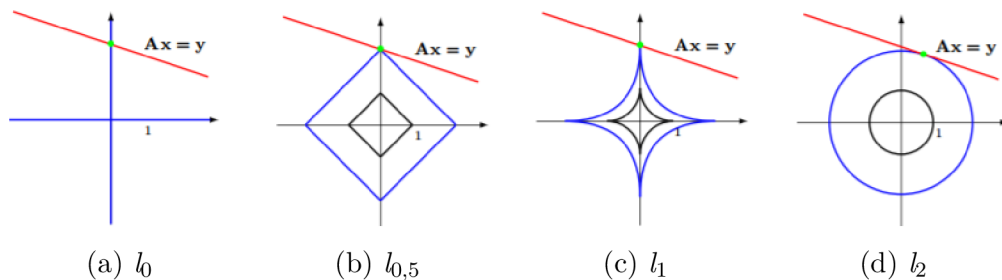
v zápětí. Pro získání nejřidšího možného řešení je tedy nutné využít maximálně nekoherentní slovníky [17].

V případě, že by bylo požadováno nalezení nejpřesnějšího řešení, nastává problém. Muselo by totiž být nalezeno takové řešení, které odpovídá \mathbf{x} s minimální l_0 -normou. Pro nalezení k_0 -řidkého řešení by bylo nutné projít všechny možné varianty z $\binom{N}{k_0}$. Bylo by tedy nutné projít všechny možné kombinace, které vzniknou v podmnožinách sloupců matice. Toto řešení odpovídá složitosti NP (nedeterministická polynomiální). To z praktického hlediska není přijatelné, protože N nabývá velkých hodnot. Z tohoto důvodu je využita aproximační metoda. Nekonvexní norma l_0 je nahrazena konvexní l_1 -normou. Pro tuto aproximační úlohu platí [17]:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ vzhledem k } \mathbf{Ax} = \mathbf{y}. \quad (3.14)$$

Možnost získat řešení v polynomiálním čase je vykoupena určitou mírou nepřesnosti [17].

Vztah 3.14 nezaručuje, že jde o nejřidší možné řešení, avšak je možné dokázat, že ve většině případů je řešení l_1 -relaxace shodné s l_0 -minimalizací. Řešení nalezených pomocí l_1 -minimalizace může být v některých případech nekonečně mnoho, avšak ta nejřidší budou výhradně ta krajní. Pro znázornění ověření shodnosti řešení může posloužit obrázek 3.3, na kterém jsou zobrazeny „nafukující se“ jednotkové koule v různých normách, kde je jejich dotyk s nadrovinou určen soustavou $\mathbf{Ax} = \mathbf{y}$. Jestliže by nadrovina procházela koulí l_1 znamenalo by to, že l_1 -relaxace má právě nekonečně mnoho řešení, nejřidší z nich by byly právě průsečíky s osou x_1 a x_2 . Na obrázku je rovněž vidět i fakt, že jednotková koule v normě l_0 se nemění, jelikož ji neovlivňuje velikost nenulových prvků [17, 19].



Obr. 3.3: Znázornění „nafukující se“ jednotkových koulí [17].

Ekvivalenci řešení l_0 - a l_1 -minimalizace je možné zajistit pomocí několika podmínek. V praxi však nenalezli příliš velké uplatnění, proto zde nebudou podrobně rozebírány. Jde tedy o tyto podmínky [19].

1. **Vlastnost nulového prostoru** (NSP – Null Space Property). Pokud je tato podmínka splněna pro jakýkoliv vektor z jádra $\ker \mathbf{A}$, pak to pro tento vektor

znamená, že je v něm koncentrována norma ve velmi malém počtu prvků. Proto je možné tvrdit, že NSP zajistí k -řádké řešení jednoznačně a toto řešení je nalezitelné pomocí l_1 -minimalizace. Toto pravidlo platí i obráceně. Jestliže je možné obnovit všechny k -řádké vektory \mathbf{x} ze soustavy $\mathbf{Ax} = \mathbf{y}$ s využitím l_1 -minimalizace, pak je o \mathbf{A} možné tvrdit, že splňuje podmínku NSP [17, 19].

2. **Vlastnost zeslabené izometrie** (RIP – Restricted Isometry Property). Tato podmínka není tak výpočetně náročná jako NSP. Navíc oproti NSP vykazuje stabilitu i pod vlivem šumu. Podstatnou nevýhodou této metody je, že se zatím nikomu nepovedlo vytvořit deterministickou matici, která by tuto podmínku splňovala s předem stanovenými parametry. Bylo však zjištěno, že RIP splňují matice, které byly vytvořeny pomocí náhody, tedy např. zvolením n řádků. V praxi jsou využívány nejvíce [17, 19].

- **Gausovské matice** – prvky těchto matic jsou generovány nezávisle z normálního rozdělení.
- **Bernoulliiovské matice** – jednotlivé prvky nabývají náhodnou hodnotu $\pm \frac{1}{\sqrt{m}}$ se stejnou pravděpodobností [17, 19].

Tyto matice jsou využívány při komprimovaném snímání [20].

4 PROXIMÁLNÍ ALGORITMY

Proximální algoritmy obsahují velké množství konvexních úloh mezi, které patří i relaxované úlohy s l_1 -normou a dalšími normami. Tyto algoritmy byly nalezeny ještě před řídkými reprezentacemi [21, 22], avšak svůj věhlas získaly až právě díky řídkým reprezentacím signálů. Optimalizace vychází z předem známých podmínek konvergence, za kterých algoritmus směřuje k tíženému výsledku. Je zde využito postupné minimalizace funkcionálu. Tato kapitola se zabývá řešením tzv. *neomezené konvexní úlohy*

$$\arg \min_{\mathbf{x}} g(\mathbf{x}) + h(\mathbf{x}) \quad [17]. \quad (4.1)$$

Podmínky, které funkce g a h musí splňovat budou zmíněny až později v této kapitole.

4.1 Omezený a neomezený tvar optimalizačních úloh

Úprava rovnice 4.1 do tvaru

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\| \text{ vzhledem k } \mathbf{Ax} = \mathbf{y} \quad [17], \quad (4.2)$$

představuje minimalizační úlohu v tzv. *omezeném tvaru*. Tento tvar neodpovídá 4.1, avšak do tohoto tvaru lze převést pomocí indikátorové funkce.

Definice 4.1. Necht \mathbb{R}^n má neprázdnou podmnožinu C . Charakteristická (indikátorová) funkce množiny C je následující[17]:

$$i_C : \mathbf{x} \mapsto \begin{cases} 0 & \text{pro } \mathbf{x} \in C, \\ \infty & \text{jinak.} \end{cases} \quad (4.3)$$

Poté je místo úlohy 4.1 řešena úloha nová ve tvaru

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 + i_{\{\mathbf{x}:\mathbf{Ax}=\mathbf{y}\}} \quad [17]. \quad (4.4)$$

Rovnici 4.2 je v žádaném tvaru. Druhý ze sčítanců zajišťuje, že hledané řešení bude náležet do množiny vhodných řešení. Analogicky pro další typ omezené úlohy

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ vzhledem k } \|\mathbf{Ax} - \mathbf{y}\|_2 \leq \delta, \quad (4.5)$$

což lze upravit do tvaru

$$\arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 + i_{\{\mathbf{x}:\|\mathbf{Ax}-\mathbf{y}\|_2 \leq \delta\}}. \quad (4.6)$$

Posledním typem úloh, se kterými se lze setkat se nazývá tzv. *regularizovaná lineární inverzní úloha*, jelikož rozšiřující člen omezuje počet vhodných řešení z nekonečně mnoha na menší množinu. Díky čemuž dojde k regularizování řešené úlohy [17, 23].

4.2 Proximální operátor

Proximální operátor je základní částí algoritmů řešících úlohy neomezeného typu, proto mu je zde nejprve věnována pozornost. Samotný proximální operátor je vlastně rozšíření projekčního operátoru na konvexní množinu.

Definice 4.2. Za předpokladu, že je funkce $f(\mathbf{x})$ zdola polospojité na podmnožině u metrického prostoru, pokud pro volitelné $\alpha \in \mathbb{R}$ je množina $\{\mathbf{x}: f(\mathbf{x}) > \alpha\}$ otevřená [17].

Definice 4.3. Subdiferenciál je definován jako konvexní zdola polospojité funkce, která má neprázdný definiční obor. Prvky \mathbf{u} subdiferenciálu se nazývají subgradients. Tato funkce bude označena $f(\mathbf{x})$, pak pro ni platí $f: \mathbb{R}^n \rightarrow (-\infty, \infty)$. Subdiferenciál funkce f představuje operátor $\partial f: \mathbb{R}^n \rightarrow 2^{\mathbb{R}^n}$ definovaný takto [17, 21]:

$$\mathbf{x} \mapsto \left\{ \mathbf{u} \in \mathbb{R}^n: (\forall \tilde{\mathbf{x}} \in \mathbb{R}^n) (\tilde{\mathbf{x}} - \mathbf{x})^\top \mathbf{u} + f(\mathbf{x}) \leq f(\tilde{\mathbf{x}}) \right\}. \quad (4.7)$$

Operátor subgradientu ∂f představuje zavedený pojem pro gradienty nediferencovatelných funkcí. Tam kde je f diferencovatelná jsou si pojmy gradient a subgradient rovny. Gradient je definován jako $\nabla f = \left[\frac{\nabla f}{\nabla x_1}, \dots, \frac{\nabla f}{\nabla x_n} \right]$. Operátor subgradientu slouží k přiřazení množiny k bodu funkce [17, 23, 24, 25, 26]. Velmi pěkné grafické znázornění subdiferenciální funkce je uvedeno v [17, 24, 26].

Definice 4.4. Necht funkce f je konvexní zdola polospojité funkce s neprázdným definičním oborem, pro kterou platí $f: \mathbb{R}^n \rightarrow (-\infty, \infty)$ má následující minimalizační problém pro všechny $\mathbf{x} \in \mathbb{R}^n$:

$$\arg \min_{\mathbf{y} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2 + f(\mathbf{y}), \quad (4.8)$$

který má jednoznačné řešení. Toto řešení bude značeno $\text{prox}_f \mathbf{x}$. Vzniklý operátor $\text{prox}_f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ se nazývá proximální operátor funkce f [17, 21, 23]. Vztah 4.8 představuje z pohledu zpracování signálů regularizované potlačování šumu. Pokud signál \mathbf{x} vznikl pomocí aditivního zašumění původního signálu, který má být rekonstruován zpět, pak $\|\mathbf{x} - \mathbf{y}\|_2$ dbá na to, aby se výsledek příliš nevzdálil od původního signálu. Zatímco člen $f(\mathbf{y})$ zajišťuje pomyslnou váhu, která slouží k regulaci hledané hodnoty \mathbf{y} pomocí konvexní funkce. Pomocí této regulace jsou vynuceny některé předem známé vlastnosti \mathbf{y} . Například může jít o tyto vlastnosti:

- energie $f(\mathbf{y}) = \|\mathbf{y}\|_2$,
- relaxovaná řídkost $f(\mathbf{y}) = \|\mathbf{y}\|_1$,
- totální variace $f(\mathbf{y}) = \|\mathbf{y}\|_{\text{TV}}$ [17].

Tvrzení 4.5. Jestliže φ je konvexní zdola polospojité funkce, která má neprázdný definiční obor, pak pro φ platí: $\varphi: \mathbb{R}^n \rightarrow (-\infty, \infty)$. Pak tedy pro změnu měřítka, resp. posunutí platí vztah uvedený v tabulce 4.1 [17, 21].

Tab. 4.1: Tabulka definující posunutí [21]

| $f(\mathbf{x})$ | $prox_f(\mathbf{x})$ |
|---|--|
| $\varphi(\mathbf{x} - \mathbf{z}), \mathbf{z} \in \mathbb{R}^N$ | $\mathbf{z} + prox_\varphi(\mathbf{x} - \mathbf{z})$ |

4.3 Proximální operátor l_1 -normy

Tento operátor je zde uveden právě, protože v kapitole 5 bude využit k formulaci konkrétního řešení Douglas-Rachford algoritmu. Proximální operátor této normy je velmi významný při hledání řídkých reprezentací signálů. Vztah pro relaxovanou řídkost, který byl zmíněn v podkapitole 4.2 bude doplněn o *regularizační parametr* λ . Pak tedy vztah pro proximální operátor l_1 -normy bude ve tvaru $f(\mathbf{x}) = \lambda \|\mathbf{y}\|_1$. Proximální operátor l_1 -normy bude tedy roven:

$$\begin{aligned} \text{prox}_{\lambda \|\cdot\|_1}(\mathbf{x}) &= \arg_{\mathbf{y}} \min \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{y}\|_1 \\ &= \arg_{\mathbf{y}} \min \frac{1}{2} \sum_{i=1}^n (x_i - y_i)^2 + \lambda \sum_{i=1}^n |y_i|. \end{aligned} \quad (4.9)$$

Pro vektor \mathbf{x} platí, že pro něj proximální operátor vrací takovou množinu $\{y_i\}$, která je minimální. Po úpravách je získána rovnice, která platí pro všechna nenulová x_i , ve tvaru:

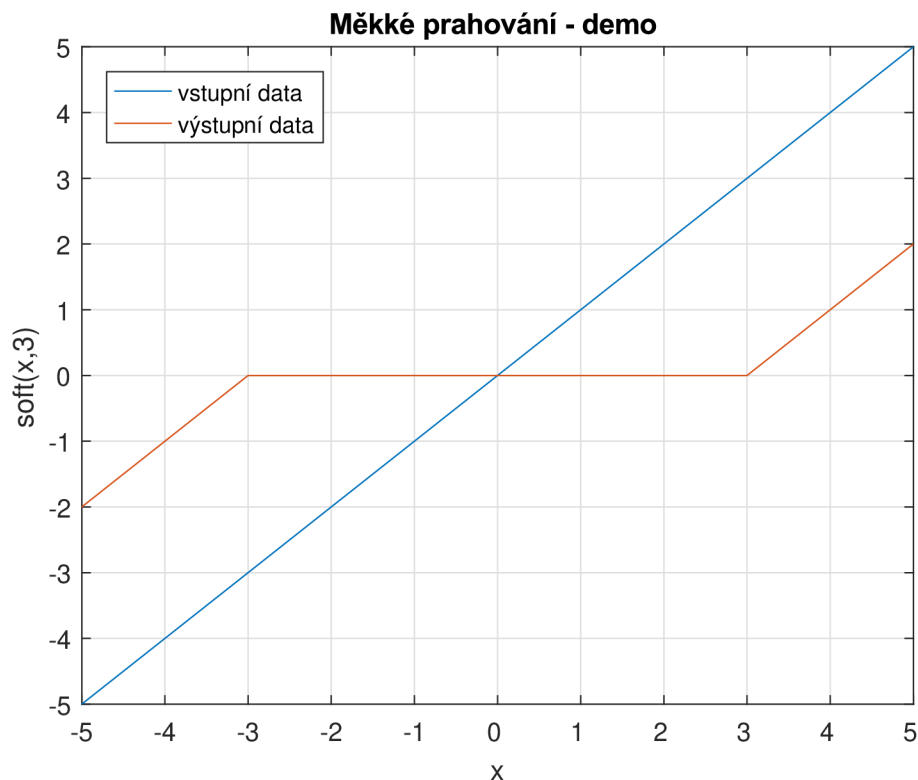
$$y_i = \frac{x_i}{|x_i|} \max(|x_i| - \lambda, 0). \quad (4.10)$$

Funkce 4.10 je označovaná jako *měkké prahování* [17, 27]. Funkce měkkého prahování je znázorněna na obrázku 4.1, tento obrázek byl vytvořen pomocí funkce „demo_mekke_prahovani.m“, která je umístěna v adresáři vytvořeno. Měkké prahování bude dále značeno $y_i = \text{soft}_\lambda(x_i)$ nebo vektorově $\mathbf{y} = \text{soft}_\lambda(\mathbf{x})$, kde jsou prahovány jednotlivé složky vektoru \mathbf{x} samostatně.

4.4 Proximální operátor pro totální variaci

Proximální operátor totální variace pro jednorozměrné signály je definován takto:

$$\begin{aligned} \text{prox}_{\lambda \|\cdot\|_{\text{TV}}}(x) &= \arg_{\mathbf{y}} \min \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{y}\|_{\text{TV}} \\ &= \arg_{\mathbf{y}} \min \frac{1}{2} \sum_{i=1}^n (x_i - y_i)^2 + \lambda \sum_{i=1}^{n-1} |y_{i+1} - y_i|. \end{aligned} \quad (4.11)$$



Obr. 4.1: Ukázka funkce měkkého prahování pro skalární veličinu. Vytvořeno pomocí funkce s názvem „demo_mekke_prahovani.m“ uložené v adresáři vytvořeno.

Hodnotu proximálního operátoru totální variace není možné analyticky určit. K výpočtu tohoto algoritmu jsou využívány převážně iterativní algoritmy, avšak v článku [28] z roku 2013 je uveden algoritmus, který má výpočetní čas lineárně úměrný počtu vzorků. TV-norma představuje l_1 -normu diferencí signálů, což vzhledem ke konvexní relaxaci znamená, že TV-norma bude směřovat k signálům s nízkým gradientem, tedy k signálům po částech konstantním [17].

Vlivy regularizačního parametru

V této části bude v krátkosti věnována pozornost vlivu regularizačního parametru λ na výsledek optimalizační úlohy. Z principu regularizační parametr funguje následovně. Pokud je jeho hodnota velká, pak dochází k velké penalizaci neřídých řešení, avšak pokud dojde k nastavení příliš velké hodnoty regularizačního parametru, pak může dojít k odchýlení od dat. Naopak při malé hodnotě regularizačního parametru dochází pouze k nepatrnému odfiltrování neřídých řešení. Nastavení regularizačního parametru není možné předem určit je nutné pro každou úlohu zjistit konkrétní nejlepší nastavení experimentálně [17]. Vlivy regularizačního parametru λ jsou znázorněny na obrázku 4.2. Obrázek 4.2a představuje poškozený obrázek, který byl rekonstruován s takovým to nastavením parametrů. Parametr tolerance byl nastaven

na hodnotu 5×10^{-15} , maximální počet iterací byl roven 3000 a parametr λ nabýval postupně těchto hodnot 4.2b $\lambda = 1$, 4.2c $\lambda = 50$ a 4.2d $\lambda = 100$. Obrázek 4.2 rovněž ukazuje způsob jakým pracuje totální variace, jde tedy o to, že se snaží co nejdéle udržet konstantní hodnotu, což demonstrují části 4.2c a 4.2d.



Obr. 4.2: Znárodnění vlivu regularizačního parametru λ na rekonstrukci obrazu. Výsledky získané pomocí vytvořené funkce s názvem „demo_vliv_lambda.m“, která je umístěna v adresáři vytvořeno.

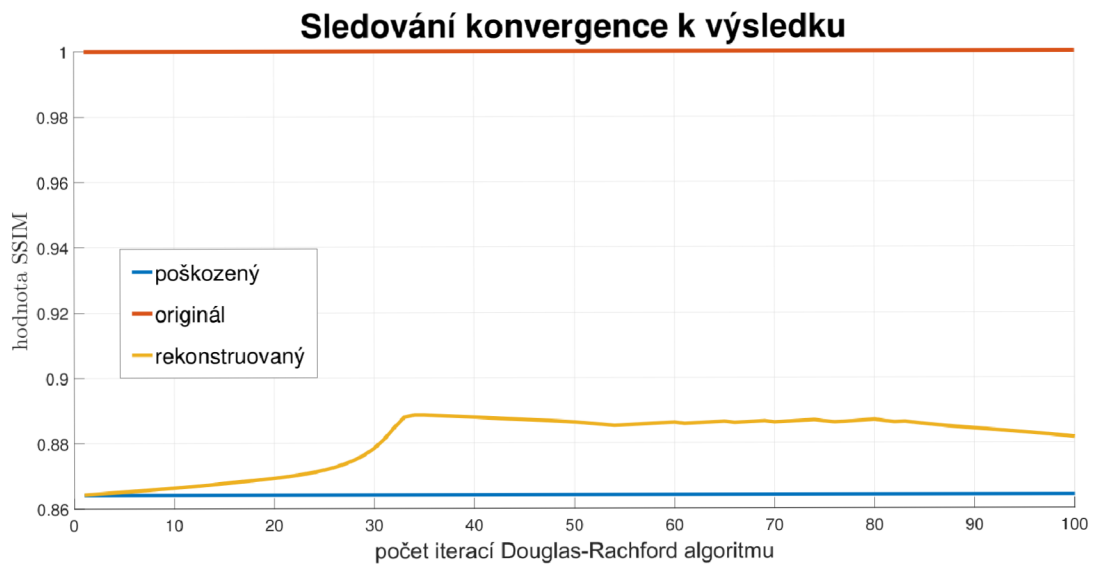
4.5 Douglas-Rachford algoritmus

Podle [21] je současná definice Douglas-Rachford algoritmu (dále v textu bude uváděno pouze DR algoritmus) nejbližší původní definici, která sloužila k řešení matic ve tvaru $\mathbf{y} = \mathbf{Ax} + \mathbf{Bx}$ [29], kde \mathbf{A} a \mathbf{B} jsou pozitivně-definitní matice [30]. Tato metoda byla později upravena tak, aby mohla být použita k řešení nelineárních problémů [21, 23]. Následně došlo k mnoha dalším modifikacím, které některé z nich jsou zmíněny v [31, 32, 33, 34, 35]. DR algoritmus je tedy speciální případ zpětně-dopředného algoritmu více k této problematice je k nalezení například zde [21, 23]. Tyto podmínky vedly ke stanovení struktury DR algoritmu, která je následující:

$$\begin{aligned}
 &\text{Fix } \epsilon \in [0, 1], \gamma > 0, y_0 \in \mathbb{R}^N \\
 &\text{For } n = 0, 1, \dots \\
 &\quad \mathbf{x}_n = \text{prox}_{\gamma f_2}(\mathbf{y}_n) \\
 &\quad \lambda_n \in [\epsilon, 2 - \epsilon] \\
 &\quad \mathbf{y}_{n+1} = \mathbf{y}_n + \lambda_n(\text{prox}_{\gamma f_1}(2\mathbf{x}_n - \mathbf{y}_n) - \mathbf{x}_n).
 \end{aligned}$$

Každé řešení získaných pomocí algoritmu 4.5 vedou k nalezení řešení [36]. Tak jako u dopředně-zpětných algoritmů jde i v DR algoritmu o využití rozdělení problému mezi více funkcí. Konkrétně v DR algoritmu jde o oddělené řešení pro dvě funkce, které jsou řešeny pomocí jednodušších funkcí než u dopředně-zpětného algoritmu. Není zde totiž požadavek, aby jedna z funkcí měla Lipschitzův spojitý gradient

[21]. Aplikace DR algoritmu ve spojitosti se zpracováním signálů a obrazu naleznete například v těchto zdrojích [36, 37, 38, 39, 40, 41, 42]. Obrázek 4.3 představuje ukázkou konvergence Douglas-Rachford algoritmu.



Obr. 4.3: Ukázka konvergence k výsledku u Douglas-Rachford algoritmu.

5 FORMULACE ŘEŠENÍ

V této kapitole budou nejprve představeny využitě doplňky pro matlab. Následně bude přiblíženo jakým způsobem bylo postupováno při řešení odstranění artefaktů v JPEG.

5.1 Využitě nástroje

Tato podkapitola se věnuje popisu využitých nástrojů při realizaci diplomové práce. Nejprve zde bude věnována pozornost UNLocBoXu. Následně ukázkovým souborům vytvořeným pro předmět MGMP a BZSG, které jsou volně šiřitelné pod licencí Creative Commons BY-NC.

5.1.1 UNLocBoX

UNLocBoX je doplněk matlabu sloužící pro konvexní optimalizaci [43]. Z tohoto nástroje byla využívána funkce „prox_tv“, která slouží k nalezení řešení této rovnice [44]:

$$\text{řešení} = \arg \min_{\mathbf{z}} \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \gamma \|\mathbf{z}\|_{\text{TV}}. \quad (5.1)$$

U této funkce byly nastavovány tyto parametry [44]:

- γ – přípustná odchylka od zadaného řešení,
- param.tol – ukončovací podmínka z hlediska tolerance,
- param.verbose – nastavovala se hodnota 1 pro stručný výpis (rychlejší výpočet) a 2 pro podrobný výpis,
- param.maxit – ukončovací podmínka z hlediska maximálního počtu iterací.

Samotné nastavení výpočtu pak bylo provedeno pomocí [44]:

$$\text{řešení} = \text{prox}_{\text{TV}}(\text{degradovaný obrázek}, \gamma, \text{struktura parametrů}). \quad (5.2)$$

V rámci této práce byl využíván UNLocBoX ve verzi 1.7.3 [43]. Tato verze je dostupná na github z <https://github.com/epfl-lts2/UNLocBoX>.

5.1.2 Dema pro podporu výuky předmětů MGMP a BZSG

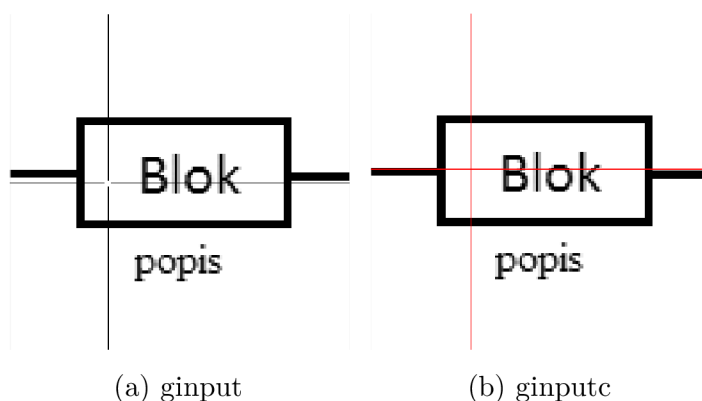
V práci byly využity ukázkové soubory, které slouží pro podporu výuky předmětů Moderní počítačová grafika a Základy počítačové sazby a grafiky. Oba předměty jsou vedeny autorem této knihovny docentem Pavlem Rajmicem. Knihovna je dostupná z této webové adresy <https://goo.gl/VvPyRi>. Z uvedené knihovny byla využita část týkající se JPEG a inicializační soubor. Pomocí této knihovny byly vytvořeny testovací obrázky. K tomuto byla využita vytvořená funkce „vytvor_testovaci_obrazky.m“,

kteřá volala upravenou funkci „demo_jpeg_uprav“. Vytvořené obrázky byly již v průběhu vytváření uloženy i do tenzoru. Obrázky byly ukládány do tenzoru v pořadí odpovídající nastavenému kvalitativnímu faktoru Q $[1, \dots, 100]$. Část [101] byla vyhrazena pro původní obrázek. Tento tenzor nese data v datovém typu double, aby nedocházelo ke zkreslení výpočtů vlivem zaokrouhlování.

5.1.3 Externí knihovny

V této podkapitole bude věnována pozornost popisu využitých externích knihoven.

- **columnlegend** – tato knihovna slouží k lepšímu popisu dat [45],
- **ginputc** – knihovna využitá pro vytvoření barevného kříže sloužícího pro výběr bodů ve vytvořené funkci s příhodným názvem „vyber_bodu.m“ [46]. Důvod této volby je ukázán na obrázku 5.1,
- **image2animation** – knihovna slouží k vytvoření animace ze sekvence obrázků, je uložena v adresáři externí [47].



Obr. 5.1: Ukázka rozdílu mezi funkcemi ginput a ginputc.

5.2 Vytvořené funkce a jejich činnost

Vytvořené funkce jsou stručně popsány v příloze B. Tato příloha obsahuje obsah příloženého disku.

5.2.1 Proximální operátor pro totální variaci

Vytvořené funkce sloužící k určení kvality obrazu byly rozděleny pomocí prefixů podle způsobu určení kvality obrazu:

- **SUBJ** – soubory s tímto prefixem zastupují funkce, které sloužily k nalezení nejlepšího subjektivního dojmu,

- **SSIM** – jak již tento prefix napovídá, tak zde byla porovnávána strukturální podobnost s originálem a hledal se nejpodobnější obrázek, ke zjištění hodnoty SSIM byla využita funkce implementovaná přímo v matlabu,
- **PSNR** – v těchto funkcích docházelo k hledání nejmenšího rozdílu právě pomocí funkce PSNR implementované v matlabu.

5.2.2 Douglas-Rachford algoritmus

Využitý DR algoritmus pracuje s osmi parametry. Nejdříve je potřeba nastavit zvolenou kvalitu JPEG, následně nastavit balanční (regularizační) parametr τ . Potom je nutno nastavit parametry pro vnitřní prox_{TV} , pro kterou je potřeba nastavit první ukončovací podmínku a to toleranci. Dále parametr verbose, který určuje četnost výpisů a druhou podmínku ukončení a to maximální počet iterací. Jelikož se jedná o vnitřní výpočty v rámci DR algoritmu není nutné, aby byly velmi přesné, proto byly voleny hodnoty, které vedou k maximálně 50 iteracím. Jakmile jsou tyto parametry nastaveny zbývá už jen nastavit parametry týkající se přímo DR algoritmu a to jsou γ , λ a maxit. V rámci diplomové práce je řešen tento minimalizační problém:

$$\arg \min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x}), \quad (5.3)$$

kde jsou jednotlivé funkce vyjádřeny jako:

$$f(\mathbf{x}) = \|\mathbf{x} - \mathbf{y}\|_1 = \text{prox}_{\|\cdot - \mathbf{y}\|_1}(\mathbf{x}), \quad (5.4)$$

$$g(\mathbf{x}) = \text{prox}_g(\mathbf{x}) = \text{prox}_{\tau\text{TV}}(\mathbf{x}). \quad (5.5)$$

Po dosazení vztahů 5.4 (výsledný tvar této funkce bude odvozen v rovnici č. 5.7) a č. 5.5 do rovnice 5.3 je získán minimalizační problém řešený v alternativním JPEG kodeku, který je ve tvaru:

$$\arg \min_{\mathbf{x}} \|\mathbf{x} + \mathbf{y}\|_1 - \text{prox}_{\tau\text{TV}}(\mathbf{x}). \quad (5.6)$$

Minimalizační problém udávaný vztahem 5.6 je následně řešen pomocí DR algoritmu. Pro DR algoritmus je nejprve nutné odvodit vztah pro \mathbf{y}_{n+1} . Pro odvození konkrétního \mathbf{y}_{n+1} v DR algoritmu bylo využito vlastnosti pro posunutí uvedené v tabulce č. 4.1 společně s odvozeným vztahem č. 4.10 definujícím l_1 -normu. Samotné odvození vypadá takto:

$$\begin{aligned} \mathbf{y}_{n+1} &= \mathbf{y}_n + \lambda \left[\text{prox}_{\gamma f_1}(2\mathbf{x}_n - \mathbf{y}_n) - \mathbf{x}_n \right] \\ &= \mathbf{y}_n + \lambda \left[\text{prox}_{\gamma \|\cdot - \mathbf{y}\|_1}(2\mathbf{x}_n - \mathbf{y}_n) - \mathbf{x}_n \right] \\ &= \mathbf{y}_n + \lambda \left[\mathbf{y} + \text{soft}_{\gamma}(2\mathbf{x}_n - \mathbf{y}_n - \mathbf{y}) - \mathbf{x}_n \right]. \end{aligned} \quad (5.7)$$

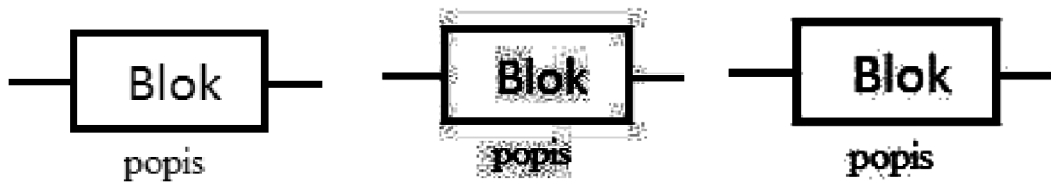
Konkrétní DR algoritmus je tedy ve tvaru:

For $n = 0, 1, \dots$

$$\mathbf{x} = \text{prox}_{r \times \gamma}(\mathbf{y}_{old})$$

$$\mathbf{y}_{new} = \mathbf{y}_{old} + \lambda [\mathbf{y}_0 + \text{soft}_{\gamma}(2\mathbf{x} - \mathbf{y}_{old} - \mathbf{y}_0) - \mathbf{x}].$$

Vytvořené funkce související s DR algoritmem mají v názvu „DR“.



(a) originál

(b) poškozený (Q = 5)

(c) restraurovaný

Obr. 5.2: Dosažené výsledky pro navržený DR. Obrázky 5.2b a 5.2c byly upraveny v programu Adobe Photoshop CC, pro lepší viditelnost vzniklých artefaktů v tištěné verzi. Porovnání původního obrázku s upraveným znázorňuje obrázek A.2.

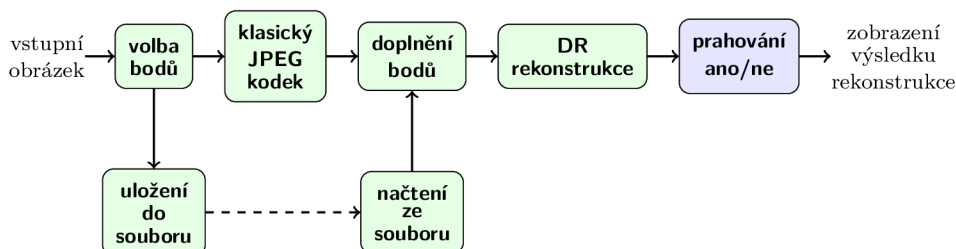
5.3 Navrhnutý JPEG kodér/dekodér

Alternativní JPEG kodek, který byl v průběhu řešení této práce navrhnout má následující strukturu:

1. nastavení parametrů,
2. zavolání funkce alternativního JPEG kodeku,
 - (a) zavolání upraveného dema JPEG z knihovny JPEG,
 - (b) zvolení bodů z původního obrazu,
 - (c) uložení bodů do souboru,
 - (d) zakódování obrazu pomocí klasického JPEG kodéru,
 - (e) dekodování obrazu pomocí klasického JPEG dekodéru,
 - (f) načtení uložených bodů,
 - (g) doplnění uložených bodů do dekodovaného obrazu,
 - (h) volba zda bude při rekonstrukci využito prahování,
 - (i) rekonstrukce pomocí Douglas-Rachford algoritmu,
 - (j) předání výsledku,
3. zobrazení výsledku,

4. vyhodnocení výsledků.

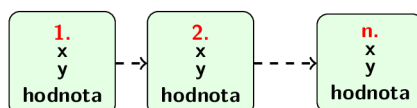
Pro lepší představu navrhnutého alternativního JPEG kodeku je na obrázku č. 5.3 uvedeno blokové schéma. Nyní budou popsány dílčí části navrhnutého řešení. Nyní budou popsány dílčí části alternativního JPEG kodeku.



Obr. 5.3: Alternativní JPEG kodek.

5.3.1 Volba bodů

Tato funkce slouží k výběru bodů z originálního obrázku. Tyto body jsou následně uloženy do souboru, který je možné přenášet společně s obrázkem, který je zakódován pomocí běžného JPEG kodéru. Pro lepší lokalizaci vybíraného pixelu byla v této funkci využita knihovna ginputc, která umožňuje rozsáhlejší nastavení kurzoru než metoda přímo implementovaná v matlabu s názvem ginput. Rozdíl těchto funkcí je znázorněn na obrázku 5.1. Struktura uloženého souboru odpovídá obrázku 5.4, kde červené číslo je pouze pro názornost, jelikož pořadí je určeno přímo samotným pořadím v poli hodnot. Souřadnice x a y jsou zaokrouhleny na celá čísla, aby byl zvolený bod jednoznačně definován.



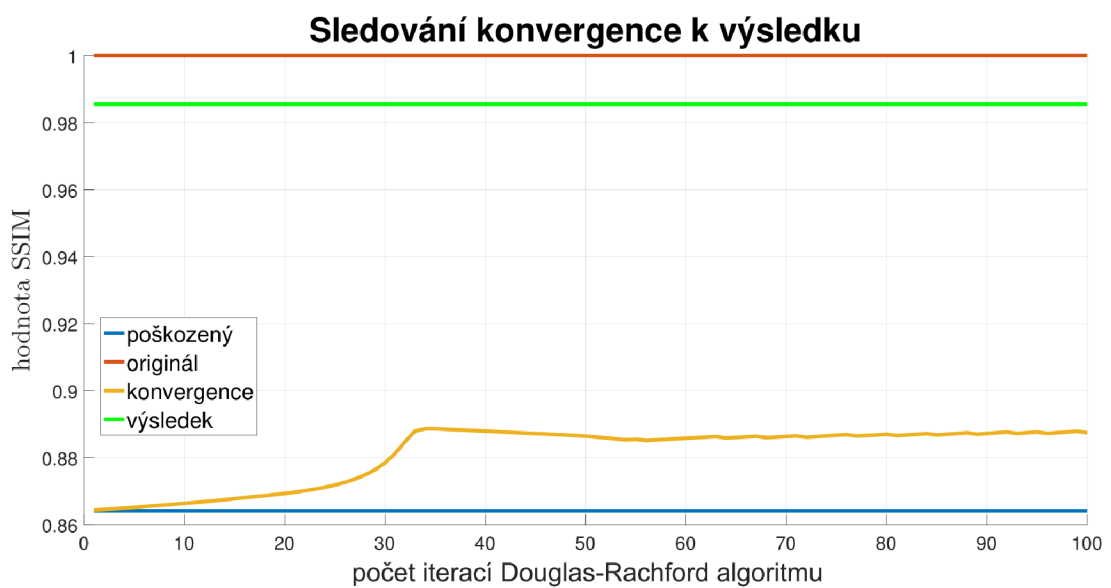
Obr. 5.4: Struktura uložených bodů.

5.3.2 Doplnění bodů

Slouží k doplnění uložených bodů z originálu do dekódovaného obrázku pomocí klasického JPEG dekodéru. Vytvořená metoda postupně projde všechny zvolené body a jejich hodnotu uloží na jejich souřadnice v dekódovaném obrázku pomocí běžného JPEG dekodéru. Následně je tento nově vzniklý obrázek dále zpracováván.

5.3.3 Prahování

Navrhnutá metoda je vhodná jak pro jednoduché obrázky, tak i pro fotografie. Prahování funguje pouze při zvolení tmavých bodů v obrázku, jelikož při takovém nastavení dochází k získání lepších výsledků pomocí DR algoritmu. V rámci experimentů byla zkoušena i metoda, která využívala adaptivní nastavení prahování v závislosti na výsledcích DR algoritmu, avšak její výsledky nebyly ani zdaleka tak dobré, jako u zvolené metody. Navržená metoda pracuje s body, které byly předtím zvoleny v původním obrázku. K prahování dojde právě tehdy, když je nějaký z bodů nulový. Pokud je takových bodů víc tak se prahování opakuje. Navrhnutý DR algoritmus dosahuje velmi dobrých výsledků, jak je možné vidět na obrázku č. 5.5.



Obr. 5.5: Ukázka konvergence k výsledku pro navrhnutý DR algoritmus. Výsledek je získán pomocí doplnění prahování Douglas-Rachford výsledku.

6 SHRNU TÍ DOSAŽENÝCH VÝSLEDKŮ

Pomocí již dříve zmíněných funkcí bylo dosaženo potlačení JPEG artefaktů. V této podkapitole budou shrnuty jednotlivé postupy při získávání výsledků. Tato podkapitola je dále členěna pro lepší rozlišení čeho se daný postup týká. Nejprve bude věnována pozornost metodám týkajících se proximální totální variace, následně metodám spojeným s Douglas-Rachford algoritmem a nakonec bude popsán alternativní JPEG kodek.

6.1 Proximální totální variace

Srovnání výsledků dosažených pomocí různých kritérií bylo shrnuto v tabulkách 6.1 a 6.2. Jaké sady obrázků se daná tabulka týká je uvedeno v levém horním rohu tabulky. Z výsledků uvedených v tabulkách 6.1 a 6.2 vyplývá, že nejlepší subjektivní řešení leží blíže k řešení vyhodnocenému pomocí zohlednění parametru SSIM. Ukázka rozdílnosti výsledků je uvedena na obrázcích 6.1 a 6.2. Tyto obrázky byly získány pomocí přednastavených funkcí, které mají prefix „zobrazeni_nejlepsich_vysledku“. Hodnota přednastavených parametrů byla zjištěna experimentálně. Hodnoty parametrů γ , ukončovací podmínky param.tol a maximálního počtu iterací, což je rovněž ukončovací podmínka, byly pro subjektivní dojem zjišťovány náhodným nastavením parametrů, poté byl spuštěn kód a vyhodnoceny výsledky. Během hledání nejlepšího subjektivního výsledku byl neprve napevno nastaven parametr ukončovací podmínky tolerance na hodnotu $5 \cdot 10^{-10}$ a maximální počet iterací na hodnotu 3000, pro manuální hledání a 50000 pro hledání pomocí cyklu. Následně byl vytvořen cyklus for, který měnil hodnoty parametru γ . Pro každé dílčí řešení byl vykreslen obrázek. Z dané sady byl zvolen obrázek, který vypadal nejlépe. Od tohoto obrázku následně byla převzata hodnota parametru γ , následně bylo voleno rozmezí $[0,5\gamma; 1,5\gamma]$ s jemnějším krokem. Poté znovu proběhlo vyhodnocení dílčích výsledků a následné zvolení nejlepšího z nich. Takto bylo pokračováno dále až do kroku o velikosti $1 \cdot 10^{-4}$. Jakmile byl nalezen optimální parametr γ přišla na řadu optimalizace ostatních parametrů, tedy hlavně parametru param.tol, který slouží jako ukončovací podmínka, u optimalizace tohoto parametru bylo přistupováno obdobně jako u parametru γ . Pro získání optimálního nastavení proběhlo vyzkoušení nejprve v rozsahu $a \cdot 10^{-b}$. Parametr a byl volen z hodnot $[1, \dots, 9]$ a parametr b byl volen z hodnot $[1, \dots, 25]$. Jakmile bylo nalezeno optimální řešení, byl tento experiment ukončen a hodnoty parametrů zapsány do příslušných souborů. Obdobně bylo postupováno i při hledání nejlepší hodnoty SSIM a PSNR. Zde ovšem neprobíhalo vyhodnocování subjektivním dojemem, ale výpočtem hodnot SSIM a PSNR funkcí z prostředí Matlab. Nalezení nejlepší hodnoty tedy probíhalo tak, že byl vytvořen nastavovací cyklus pro nastavení hodnoty

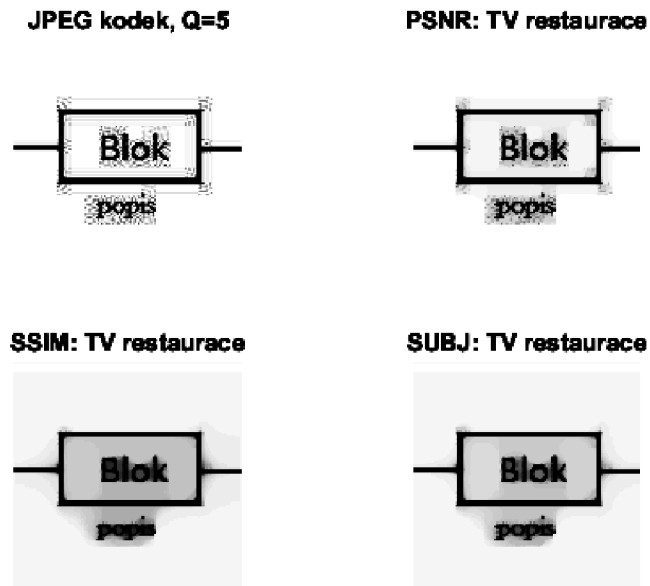
param.tol a v tomto cyklu byla vnořena další smyčka, která nastavovala parametr γ . Uvnitř vnitřní smyčky probíhal výpočet, kterému se postupně propočítávaly hodnoty s různými parametry a výsledek byl porovnáván vůči originálnímu obrázku jednak pomocí SSIM a také pomocí PSNR. Pokud v dané iteraci byla nalezena lepší shoda pro některý z parametrů, došlo k přeložení optimálních hodnot nastavení v paměti a výpočet pokračoval dál, dokud neproběhli veškeré iterace. Tímto způsobem byly nejprve získány hrubé hodnoty parametrů, které následně sloužily jako vstupní nastavení pro dohledání přesnějších hodnot těchto parametrů. Tento postup byl několikrát opakován dokud nebyla nalezena optimální hodnota. V tabulkách 6.1 a 6.2 naleznete shrnutí dosažených výsledků. Sloupec SSIM představuje nejlepší zjištěné nastavení a výsledky při zohlednění SSIM, obdobně tomu je i pro sloupec SUBJ, kde je optimální nastavení a výsledky pro subjektivní vyhodnocení. Sloupec s názvem PSNR, představuje optimální nastavení a výsledky pro PSNR. Poslední sloupec s názvem JPEG $Q=5$ ilustruje rozdílnost degradovaného obrázku vůči originálu. Jak je vidět z hodnot v tabulce ve všech případech došlo ke zlepšení kvality vůči degradovanému obrázku, což je mimo jiné vidět i na obrázcích 6.1 a 6.2.

Tab. 6.1: Shrnutí dosažených výsledků pro blokový diagram.

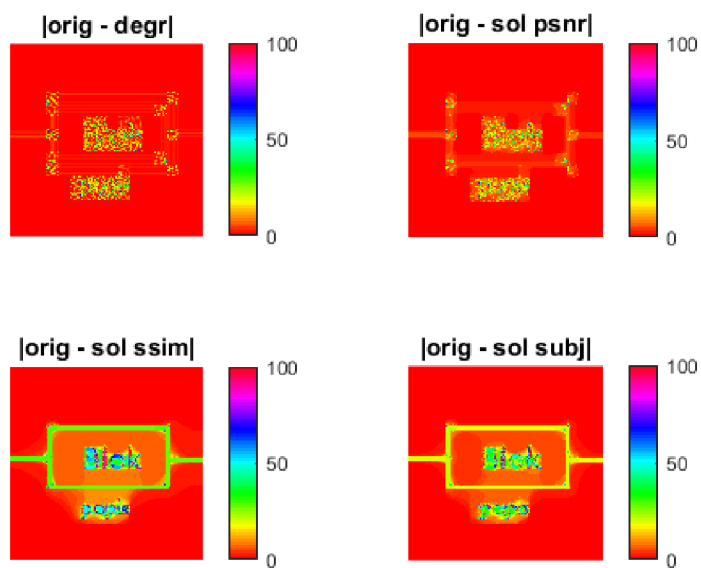
| blokový diagram | SSIM | SUBJ | PSNR | JPEG $Q=5$ |
|------------------|--------------------|--------------------|--------------------|------------|
| param.tol | $1 \cdot 10^{-11}$ | $5 \cdot 10^{-12}$ | $5 \cdot 10^{-15}$ | - |
| gamma | 38 | 25 | 3,5270 | - |
| iterací | 1739 | 6030 | 16464 | - |
| SSIM | 0,8721 | 0,8583 | 0,8680 | 0,8541 |
| PSNR | -22,1060 | -19,4376 | -14,5020 | -15,1189 |
| Frobeniova norma | 1631,2054 | 1199,7465 | 679,6906 | 729,7178 |

Tab. 6.2: Shrnutí dosažených výsledků pro fotografii.

| Lena | SSIM | SUBJ | PSNR | JPEG $Q=5$ |
|------------------|--------------------|--------------------|--------------------|------------|
| param.tol | $5 \cdot 10^{-15}$ | $2 \cdot 10^{-12}$ | $4 \cdot 10^{-15}$ | - |
| gamma | 0,99 | 0,01 | 1 | - |
| iterací | 1117 | 491 | 1170 | - |
| SSIM | 0,8625477 | 0,85621108 | 0,8625473 | 0,85606798 |
| PSNR | -17,6452 | -17,8947 | -17,6437 | -17,8984 |
| Frobeniova norma | 976,0478 | 1004,4930 | 975,8801 | 1004,9189 |



(a) porovnání kvality zobrazení

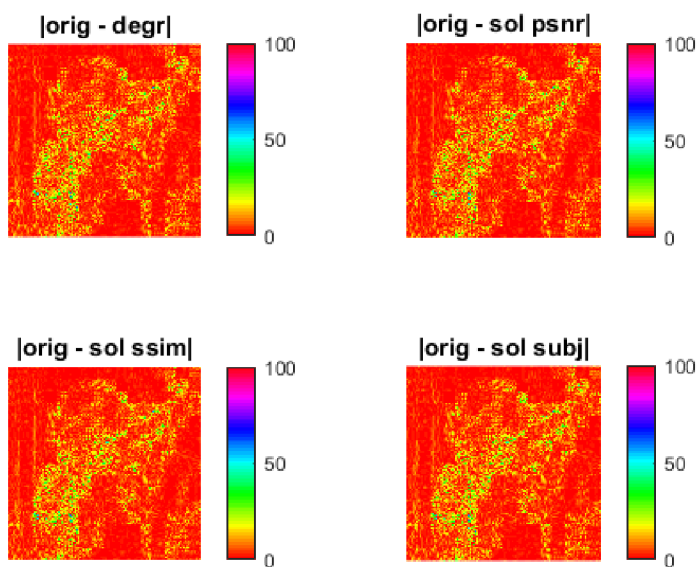


(b) porovnání rozdílů vůči originálu

Obr. 6.1: Zobrazení výsledků pro blokový diagram. Na upraveném obrázku 6.1a pro tištěnou verzi práce se nalézá zobrazení degradovaného obrázku a jeho restaurací pomocí zohlednění PSNR, SSIM a subjektivního dojmu. Porovnání původního obrázku s upraveným představuje obrázek A.4. Druhý obrázek ukazuje grafické znázornění rozdílů mezi obrázky uvedenými na obrázku 6.1a a originálem.



(a) porovnání kvality zobrazení



(b) porovnání rozdílů vůči originálu

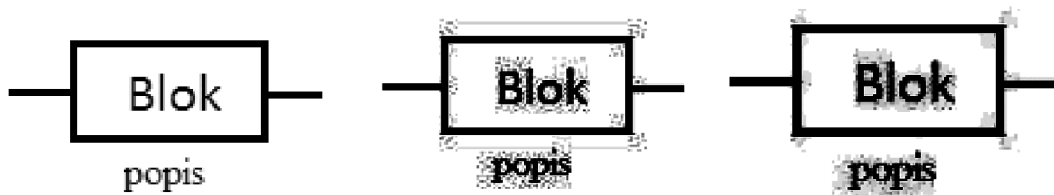
Obr. 6.2: Zobrazení výsledků pro černobílou fotografii Leny. Na obrázku 6.2a se nalézá zobrazení degradovaného obrázku a jeho restaurací pomocí zohlednění PSNR, SSIM a subjektivního dojmu. Druhý obrázek ukazuje grafické znázornění rozdílů mezi obrázky uvedenými na obrázku 6.1a a originálem.

6.2 Douglas-Rachford algoritmus

Nejprve byla definována funkce s názvem „demo_DR_test_manual.m“, která slouží k manuálnímu nastavování šestice parametrů díky, kterým bylo nalezeno uvedené řešení. Nastavované parametry včetně jejich hodnoty jsou uvedeny v tabulce č. 6.3. Pomocí takto nastaveného DR algoritmu bylo dosaženo výsledku uvedeného v obrázku č. 6.3. Hodnoty rozdílnosti jsou uvedeny v tabulce č. 6.4 a znázornění konvergence, jak rekonstruovaný obrázek směřoval k dosaženému výsledku představuje obrázek č. 6.4. Poté byla vytvořena sofistikovanější metoda, která sloužila ke sledování

Tab. 6.3: Nastavené hodnoty parametrů

| | parametr | hodnota | | parametr | hodnota |
|----------|----------|---------|----------|----------|---------|
| param_TV | tol | 1e-6 | param_DR | gamma | 0,01 |
| | verbose | 0 | | lambda | 0,9999 |
| | maxit | 50 | | maxit | 250 |



(a) originál

(b) poškozený ($Q = 5$)

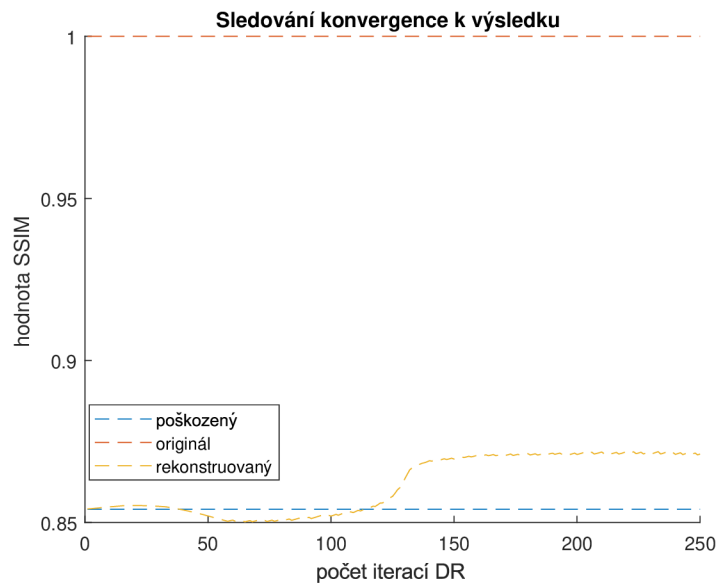
(c) restraurovaný

Obr. 6.3: Ukázka výsledků pro demo_DR_test_manual.m. Obrázky s označením 6.3b a 6.3c byly upraveny v programu Adobe Photoshop CC, pro lepší viditelnost vzniklých artefaktů v tištěné verzi. Porovnání původního obrázku s upraveným znázorňují obrázky A.2 a A.3.

Tab. 6.4: Porovnání výsledku vůči poškozenému obrázku

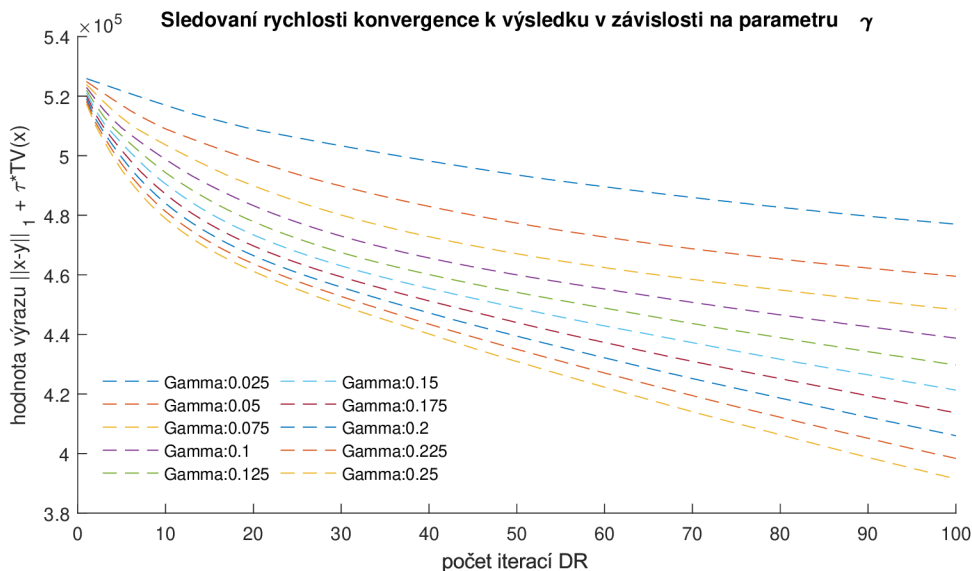
| obrázek | EPP | REPP | MSE | PSNR | SSIM |
|----------------|--------|--------|--------|----------|--------|
| poškozený | 1,5027 | -0,079 | 32,5 | -15,1189 | 0,8541 |
| rekonstruovaný | 1,6214 | | 27,545 | -14,4 | 0,8711 |

konvergence při změně parametrů. Jednalo se o metodu „demo_DR_chart_data.m“.



Obr. 6.4: Zobrazení konvergence k výsledku pro DR algoritmus. Vytvořeno pomocí funkce s názvem „demo_DR_test_manual.m“ uložené v adresáři vytvořeno.

Tato metoda může být velice snadno upravena pro sledování vlivu, kteréhokoliv parametru na výslednou konvergenci. Obrázek č. 6.5 znázorňuje vliv parametru γ na konvergenci. Z funkce „demo_DR_chart_data.m“ byla následně vytvořena



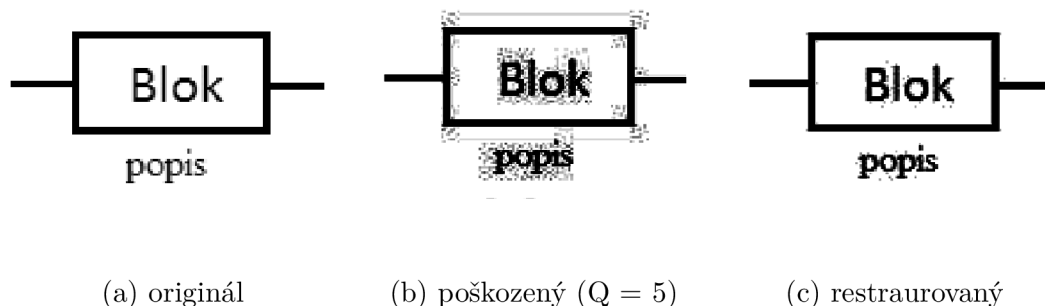
Obr. 6.5: Zobrazení vlivu parametru γ na konvergenci k výsledku pro DR algoritmus. Vytvořeno pomocí funkce s názvem „demo_DR_chart_data.m“ uložené v adresáři vytvořeno.

lepší funkce, která sloužila ke snadnějšímu nalezení nastavení parametrů pomocí

sledování konvergence celého minimalizačního problému. Vylepšená funkce nese název „demo_DR_nalezeni_nejlepsiho.m“. V této funkci jde o nastavení škálování pro zvolený parametr a následné procházení výsledků pomocí šipek. Poté se převezmou hodnoty nastavené pro nejlepší výsledek ze sady a může se pokračovat dále ve zvyšování přesnosti daného výsledku. Ve zdrojovém kódu byla ponechána ukázka pro parametr γ , avšak velmi snadno lze tento zdrojový kód upravit pro další z parametrů. V návaznosti na funkci „demo_DR_nalezeni_nejlepsiho.m“ byly vytvořeny funkce „demo_jpeg_DP.m“ a „demo_jpeg_DP_2.m“, které vedly k vytvoření „demo_jpeg_alternativni.m“. Tato funkce bude podrobně popsána v následující podkapitole.

6.3 Alternativní JPEG kodek

Navrhnuté řešení bylo složeno jednak z upravené knihovny JPEG, která je součástí knihovny vytvořené pro podporu výuky předmětů BZSG a MGMP, ale rovněž byly využity i vytvořené funkce. Navrhnuté řešení se ukázalo jako optimální, jak při využití u jednoduchých obrázků, tak i u fotografií. Nejhorších výsledků navrhnuté řešení dosahuje u rekonstrukce textu. Shrnutí dosažených výsledků včetně ukázky jednotlivých zdrojových kódů je umístěno v adresáři videa, kde jsou jednotlivé zdrojové kódy okomentovány včetně ukázky jejich funkčnosti. Ukázka dosažených výsledků pro blokový diagram je umístěna v obrázku č. 6.6. Porovnání dosažených výsledků je umístěno v tabulce č. 6.5.



Obr. 6.6: Dosažené výsledky pro navrhnutý DR. Obrázky 6.6b a 6.6c byly upraveny v programu Adobe Photoshop CC, pro lepší viditelnost vzniklých artefaktů v tištěné verzi. Porovnání původního obrázku s upraveným znázorňuje obrázek A.2.

Tab. 6.5: Porovnání hodnot pro JPEG Q=5 a rekonstruovaný obrázek

| obrázek | EPP | REPP | MSE | PSNR | SSIM |
|----------------|--------|--------|---------|----------|--------|
| JPEG Q=5 | 1,5027 | 0,6513 | 32,5005 | -15,1189 | 0,8541 |
| rekonstruovaný | 0,5240 | | 13,0553 | -11,1579 | 0,9855 |

6.4 Demo DP

V této podkapitole bude popsána vytvořená funkce, která slouží k porovnání dílčích výsledků, kterých bylo v této práci dosaženo. Komentovaná ukázka této funkce je k dispozici v adresáři videa.

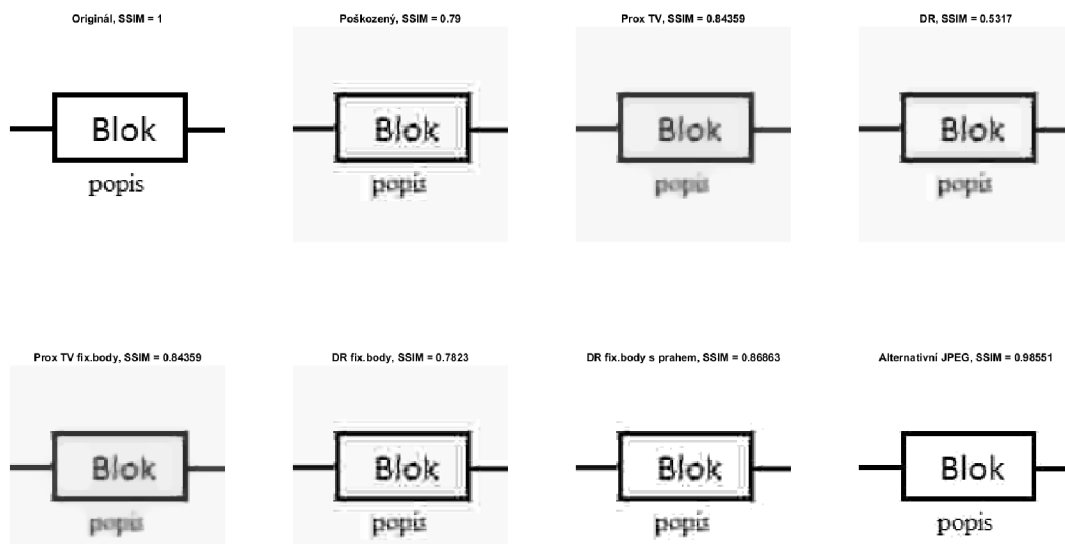
Tato funkce je uložena v adresáři vytvořeno pod názvem „demo_DP.m“. Po jejím spuštění dojde nejprve k výpočtu řešení pomocí prox_{TV} , následně je nalezeno řešení pomocí DR algoritmu. Poté je uživatel vyzván k volbě bodů v obrázku. Zvolené body jsou následně doplněny do rekonstruovaného obrázku pomocí běžného JPEG dekodéru a s obrázkem s doplněnými body je poté provedena prox_{TV} , následně řešení pomocí DR algoritmu bez prahování, poté s prahováním a předání alternativnímu JPEG. Po dokončení všech výpočtů dojde k zobrazení výsledků a jejich vyhodnocení v tabulce. Ukázka dosažených výsledků pro blokový diagram je v tabulce č. 6.6. Obrázek č. 6.7 představuje ukázkou stylu zobrazení porovnání výsledků pro dílčí metody.

Tab. 6.6: Porovnání výsledků (demo_DP.m) pro všechny dílčí metody

| obrázek | EPP | REPP | MSE | PSNR | SSIM |
|---------------------------|---------------|---------------|----------------|-----------------|---------------|
| Poškozený (JPEG Q=5) | 11.67 | – | 406,6736 | –26,0925 | 0,7900 |
| prox_{TV} | 15,5215 | –0,3300 | 566,8956 | –27,5350 | 0,8436 |
| DR bez prahu | 12,9008 | –0,1055 | 423,0356 | –26,2638 | 0,5317 |
| prox_{TV} | 15,5162 | –0,3296 | 566,3723 | –27,5310 | 0,8436 |
| DR bez prahu | 11,7958 | –0,0108 | 399,2195 | –26,0121 | 0,7830 |
| DR s prahem | 5,5929 | 0,5208 | 346,2726 | –25,3942 | 0,8687 |
| Alternativní JPEG | 0,5249 | 0,9559 | 13,0661 | –11,1615 | 0,9855 |

6.5 Možnosti navázání na tuto práci

Na tuto práci by bylo možné navázat jednak rozšířením funkcionality základního JPEG. Zdrojové kódy by mohly být rozšířeny o funkcionality s barevnými obrázky bez omezení jejich rozměrem. Dále by mohlo být navržené řešení rozšířeno například



Obr. 6.7: Ukázka zobrazení porovnání výsledků dílčích částí s navrhnutým alternativním JPEG. Vytvořeno pomocí funkce s názvem „demo_DP.m“ uložené v adresáři vytvořeno.

o implementaci paralelního Douglas-Rachford algoritmu, kde by byl sledován vliv této implementace na konvergenci k výsledku a jeho kvalitu. Další možností jak tuto práci rozšířit by bylo zkoumání vlivu alternativních kvantizačních tabulek, jelikož dle standardu JPEG je možno využívat i alternativní kvantizační tabulky, jinak řečeno je možné vytvořit kvantizační tabulky na míru danému obrázku nebo obrázkovému typu. Běžně využívané kvantizační tabulky jsou vhodné pro fotografie, avšak u textu při nastavení nižšího kvalitativního faktoru již nejsou úplně nejvhodnější. Tuto práci by bylo možno rozšířit i o využití grafické akcelerace, jelikož přímo unlocbbox podporuje u některých funkcí akceleraci pomocí cuda jader, které obsahují grafické karty od společnosti NVIDIA, avšak k tomuto řešení je nutné poznamenat, že v případě práce s obrázky s malým rozlišením (např. 128×128 px) bude trvat přesun obrázku do a z paměti grafické karty delší dobu než samotný výpočet. Může se tedy stát, že při využití grafické akcelerace dojde ke zpomalení samotného výpočtu. Pro využití CUDA akcelerace ve spojení s UNLocBoXem je tedy potřeba grafická karta od společnosti NVIDIA, dále mít v počítači nainstalovaný „CUDA toolkit“ a nastavit pro knihovnu UNLocBoX globální proměnnou „GLOBAL_useGPU“ na jedničku. Následně všude tam, kde je podporována CUDA akcelerace dojde k jejímu využití.

7 ZÁVĚR

Tato práce se zabývá problematikou odstranění artefaktů v obraze způsobených ztrátovým JPEG. K odstranění těchto artefaktů je využito proximálních algoritmů. Nejdříve je využita prox_{TV} následně byl využit **Douglas-Rachford algoritmus** s navrhnutou funkcí, která byla uvedena v rovnici 5.7. Dále byla využita metoda se zafixováním bodů z původního obrázku. Dosažené výsledky jsou demonstrovány v podkapitole 6. Vytvořené funkce byly stručně popsány v příloze B.

Diplomová práce mimo jiné popisuje problematiku ztrátové komprese obrazu, jejíž pochopení je nutné ke správnému porozumění ztrátovému JPEG, který je právě na ztrátové kompresi založený. Druhá kapitola detailně popisuje ztrátový JPEG kodek. Jsou zde zmíněny i obrazové artefakty, ke kterým u JPEG právě vlivem komprese dochází. Dále byla věnována pozornost alternativním módům, které JPEG nabízí. Třetí kapitola byla věnována řídkým reprezentacím signálů, kde došlo k ujednocení pojmů a značení. Do této kapitoly bylo zahrnuto i hledání řídkého řešení. Kapitola čtvrtá byla věnována proximálním algoritmům, které byly v následující kapitole využity k formulaci řešení. Výsledky navrhovaného řešení byly shrnuty v kapitole č. 6. V rámci demonstrace navrhovaného řešení byly vytvořeny komentované ukázky pro zdrojové kódy s prefixem demo ze složky vytvořeno, tyto ukázky jsou k nalezení na přiloženém disku v adresáři videa. Video s demonstrací výsledků jsou dále dostupné dostupné na URL: <https://goo.gl/bR98Wd>.

V průběhu zpracování této práce bylo nejprve využito **proximální totální variace** k rekonstrukci obrazu ve zvolené kvalitě JPEG. Metodika získávání výsledků a výsledky jako takové jsou umístěny v kapitole č. 6.1. Následně byl využit **Douglas-Rachford algoritmus** jehož odvození obsahuje podkapitola č. 5.2.2. Shrnutí dosažených výsledků obsahuje podkapitola č. 6.2. Z dílčí části věnované Douglas-Rachford algoritmu byl následně odvozen alternativní JPEG kodek jehož blokové schéma je znázorněno na obrázku č. 5.3, výsledky dosažené s tímto kodekem jsou stručně shrnuty v podkapitole č. 6.3. Podrobná ukázka konkrétní implementace a popis funkcí je umístěn v podkapitole č. 5.3. Podrobný popis vytvořené funkce je umístěn na sociální síti YouTube v komentovaném videu. Závěr kapitoly č. 6 je věnován porovnání dílčích dosažených výsledků s navrhnutým alternativním JPEG kodekem. I pro tuto část práce bylo vytvořeno komentované video.

V rámci této práce byl navrhnut alternativní JPEG kodek, který funguje jako nadstavba na běžně dostupný JPEG kodek. Tato struktura byla zvolena právě proto, aby mohl být obrázek dekodován i pomocí běžně dostupného dekodéru.

Tato práce byla zrealizována v prostředí programu **Matlab** ve verzi **2016a**. Vyzkoušení všech zdrojových kódů proběhlo i ve verzi **2017a**. Výsledků uvedených v podkapitole 6 bylo dosaženo pomocí nastavování parametru gamma, tolerance

pro ukončení a maximálního počtu iterací. Obrázek 6.1 ukazuje fakt, že využití řešení vedlo ke zlepšení kvality poškozeného obrazu vlivem JPEG kodeku. Navrhnuté řešení je vhodné použít jak pro jednoduché obrázky, tak pro složité obrázky. Možná rozšíření této práce byla uvedena v podkapitole 6.5.

LITERATURA

- [1] SHI, Yun Q. a Huifang SUN. *Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards*. 2nd ed. Boca Raton: CRC Press, c2008. ISBN 08-493-7364-6.
- [2] Ztrátová komprese obrazu. RAJMIC, Pavel a Jiří SCHIMMEL. *Moderní počítačová grafika* [online]. Aktualizované. Brno, 2016, s. 105-116 [cit. 2016-11-03]. ISBN 978-80-214-4906-0.
- [3] Lossy Image Compression: JPEG and JPEG2000 Standards. BOVIK, Al. *Handbook of image and video processing*. 2nd ed. Boston, MA: Elsevier Academic Press, 2005, s. 709-731. ISBN 0121197921.
- [4] Image Compression. GONZALEZ, Rafael C. a Richard E. WOODS. *Digital image processing*. 3rd ed. Upper Saddle River, N.J.: Prentice Hall, c2008, s. 525-626. ISBN 013168728X.
- [5] TIŠNOVSKÝ, Pavel. JPEG - král rastrových grafických formátů? *ROOT.cz* [online]. [cit. 2016-11-19]. Dostupné z: <https://goo.gl/r1A7Ew>
- [6] Image Coding Recommendations. *International Telecommunication Union* [online]. c2006, 23.11.2006 [cit. 2016-11-09]. Dostupné z: <https://goo.gl/AneDdt>
- [7] HAMILTON, Eric. *JPEG File Interchange Format* [online]. 1992, , 4 [cit. 2016-11-24]. Dostupné z: <https://goo.gl/fx4Px1>
- [8] ŽÁRA, Jiří, B. BENEŠ, J. SOCHOR a aj. *Moderní počítačová grafika*. Druhé vydání. Praha: Computer Press, 2005, 609 s. ISBN 80-251-0454-0.
- [9] *CHROMA SUBSAMPLING TECHNIQUES* [online]. RED.COM, INC., c2016 [cit. 2016-11-09]. Dostupné z: <https://goo.gl/zKYvuf>
- [10] Image Processing Basics. TAN, Li. *Digital signal processing: Fundamentals and Applications*. Boston: Academic Press, c2008, s. 617-698. ISBN 978-0-12-374090-8.
- [11] HASS, Calvin. *JPEG Compression Quality from Quantization Tables* [online]. ©2016 [cit. 2016-11-08]. Dostupné z: <https://goo.gl/EAxzca>
- [12] TIŠNOVSKÝ, Pavel. Programujeme JPEG: Huffmanovo kódování kvantovaných DCT složek. *ROOT.cz* [online]. [cit. 2016-11-18]. Dostupné z: <https://goo.gl/ALQn1f>

- [13] JPEG Huffman Coding Tutorial. HASS, Calvin. *Impulse Adventure* [online]. 2016 [cit. 2016-11-18]. Dostupné z: <https://goo.gl/EtCvAd>
- [14] JAKULIN, Aleks. *Baseline JPEG and JPEG2000 Artifacts Illustrated* [online]. 2004 [cit. 2016-11-19]. Dostupné z: <https://goo.gl/Q3scxq>
- [15] TIŠNOVSKÝ, Pavel. Programujeme JPEG: Progresivní JPEG a informace EXIF. *ROOT.cz* [online]. [cit. 2016-11-20]. Dostupné z: <https://goo.gl/H1cZ3v>
- [16] The JPEG Lossless Image Compression Standards. BOVIK, Al. *Handbook of image and video processing*. 2nd ed. Boston, MA: Elsevier Academic Press, 2005, s. 733-745. ISBN 0121197921.
- [17] RAJMIC, Pavel. *Řídké a nízkohodnostní reprezentace signálů s aplikacemi: Sparse and low-rank representations of signals with applications* [online]. Brno, 2014, s. 17-61 [cit. 2016-11-26].
- [18] ŠPIŘÍK, Jan, Pavel RAJMIC a Vítězslav VESELÝ. Reprezentace signálů: od báží k framům. *Elektrorevue* [online]. 2010, **2010**(6) [cit. 2016-10-30]. ISSN 1213-1539. Dostupné z: <https://goo.gl/6H77Lx>
- [19] HRBÁČEK, Radek, Pavel RAJMIC, Vítězslav VESELÝ a Jan ŠPIŘÍK. Řídké reprezentace signálů: úvod do problematiky. *Elektrorevue* [online]. 2011, **2011**(5) [cit. 2016-10-30]. ISSN 1213-1539. Dostupné z: <https://goo.gl/9xVRvE>
- [20] HRBÁČEK, Radek, Pavel RAJMIC, Vítězslav VESELÝ a Jan ŠPIŘÍK. Řídké reprezentace signálů: komprimované snímání. *Elektrorevue* [online]. 2011, **2011**(6) [cit. 2016-12-12]. ISSN 1213-1539. Dostupné z: <https://goo.gl/DQHvf2>
- [21] COMBETTES, Patrick L. a Jean-Christophe PESQUET. Proximal Splitting Methods in Signal Processing. BAUSCHKE, H. H., R. BURACHIK, P. L. COMBETTES, V. ELSER, D. R. LUKE a H. WOLKOWICZ. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering* [online]. New York: Springer, 2011, s. 185-212 [cit. 2017-03-26]. DOI: 10.1007/978-1-4419-9569-8_10. ISBN 978-1-4419-9569-8. Dostupné z: http://link.springer.com/10.1007/978-1-4419-9569-8_10.
- [22] BOYD, Stephen P. a Lieven. VANDENBERGHE. *Convex optimization*. New York: Cambridge University Press, 2004. ISBN 05-218-3378-7.
- [23] POLSON, Nicholas G., James G. SCOTT a Brandon T. WILLARD. *Proximal Algorithms in Statistics and Machine Learning* [online]. 2015 [cit. 2017-04-16]. Dostupné z: <https://arxiv.org/pdf/1502.03175.pdf>. ArXiv:1502.03175v3.

- [24] DAŇKOVÁ, M. *Komprimované snímání v perfuzním zobrazování pomocí magnetické rezonance*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014. 56 s. Vedoucí Mgr. Pavel Rajmic, Ph.D.
- [25] *Encyclopedia of Math: Subdifferential* [online]. [cit. 2017-04-15]. Dostupné z: <https://goo.gl/xP8cgf>
- [26] BOYD, S., J. DUCHI a L. VANDENBERGHE. *Subgradients* [online]. 2014 [cit. 2017-04-15]. Dostupné z: <https://goo.gl/87tviQ>. Notes for EE364b. Stanford University.
- [27] DONOHO, D.L. De-noising by soft-thresholding. *Information Theory, IEEE Transactions on* [online]. USA: IEEE, 9505, **41**(3), 613-627 [cit. 2017-04-16]. DOI: 10.1109/18.382009. ISSN 00189448. Dostupné z: <http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/document/382009/?reload=true>
- [28] CONDAT, Laurent. A Direct Algorithm for 1-D Total Variation Denoising. *Signal Processing Letters, IEEE* [online]. USA: IEEE, 1311, **20**(11), 1054-1057 [cit. 2017-04-16]. DOI: 10.1109/LSP.2013.2278339. ISSN 10709908. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6579659>
- [29] DOUGLAS, Jim a H. H. RACHFORD. On the Numerical Solution of Heat Conduction Problems in Two and Three Space Variables. *Transactions of the American Mathematical Society* [online]. 1956, **82**(2), 421- [cit. 2017-04-08]. DOI: 10.2307/1993056. ISSN 00029947. Dostupné z: <http://www.jstor.org/stable/1993056?origin=crossref>
- [30] VARGA, Richard S. *Matrix iterative analysis*. 2nd rev. and expand. ed. Berlin: Springer, 2009. ISBN 978-3-642-05154-8.
- [31] COMBETTES, Patrick L. Solving monotone inclusions via compositions of nonexpansive averaged operators. *Optimization* [online]. 2004, **53**(5-6), 475-504 [cit. 2017-04-08]. DOI: 10.1080/02331930412331327157. ISSN 0233-1934. Dostupné z: <http://www.tandfonline.com/doi/abs/10.1080/02331930412331327157>
- [32] ECKSTEIN, Jonathan a Dimitri P. BERTSEKAS. On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming* [online]. 1992, **55**(1-3), 293-318 [cit. 2017-04-08]. DOI: 10.1007/BF01581204. ISSN 0025-5610. Dostupné z: <http://link.springer.com/10.1007/BF01581204>

- [33] LIONS, P. L. a B. MERCIER. Splitting Algorithms for the Sum of Two Nonlinear Operators. *SIAM Journal on Numerical Analysis* [online]. 1979, **16**(6), 964-979 [cit. 2017-04-08]. DOI: 10.1137/0716071. ISSN 0036-1429. Dostupné z: <http://epubs.siam.org/doi/10.1137/0716071>
- [34] BORWEIN, Jonathan M., Guoyin LI a Matthew K. TAM. *Convergence rate analysis for averaged fixed point iterations in the presence of H^s older regularity* [online]. 2015 [cit. 2017-04-16]. Dostupné z: <https://arxiv.org/pdf/1510.06823.pdf>
- [35] The Douglas-Rachford Algorithm in the Absence of Convexity. *Fixed-point algorithms for inverse problems in science and engineering*. New York: Springer, c2011, s. 93-109. ISBN 978-1-4419-9569-8.
- [36] COMBETTES, Patrick L. a Jean-Christophe PESQUET. A Douglas–Rachford Splitting Approach to Nonsmooth Convex Variational Signal Recovery. *IEEE Journal of Selected Topics in Signal Processing* [online]. 2007, **1**(4), 564-574 [cit. 2017-04-08]. DOI: 10.1109/JSTSP.2007.910264. ISSN 1932-4553. Dostupné z: <http://ieeexplore.ieee.org/document/4407760/>
- [37] STEIDL, G. a T. TEUBER. Removing Multiplicative Noise by Douglas-Rachford Splitting Methods. *Journal of Mathematical Imaging and Vision* [online]. 2010, **36**(2), 168-184 [cit. 2017-04-08]. DOI: 10.1007/s10851-009-0179-5. ISSN 0924-9907. Dostupné z: <http://link.springer.com/10.1007/s10851-009-0179-5>
- [38] SETZER, Simon. *Split Bregman Algorithm, Douglas-Rachford Splitting and Frame Shrinkage* [online]. , 464 [cit. 2017-04-08]. DOI: 10.1007/978-3-642-02256-2_39. Dostupné z: http://link.springer.com/10.1007/978-3-642-02256-2_39
- [39] DURAND, Sylvain, Jalal FADILI a Mila NIKOLOVA. Multiplicative Noise Removal Using L1 Fidelity on Frame Coefficients. *Journal of Mathematical Imaging and Vision* [online]. 2010, **36**(3), 201-226 [cit. 2017-04-08]. DOI: 10.1007/s10851-009-0180-z. ISSN 0924-9907. Dostupné z: <http://link.springer.com/10.1007/s10851-009-0180-z>
- [40] DUPE, F.-X., J.M. FADILI a J.-L. STARCK. A Proximal Iteration for Deconvolving Poisson Noisy Images Using Sparse Representations. *IEEE Transactions on Image Processing* [online]. 2009, **18**(2), 310-321 [cit. 2017-04-08]. DOI: 10.1109/TIP.2008.2008223. ISSN 1057-7149. Dostupné z: <http://ieeexplore.ieee.org/document/4738431/>
- [41] CHAUX, Caroline, Jean-Christophe PESQUET a Nelly PUSTELNIK. Nested Iterative Algorithms for Convex Constrained Image Recovery Problems. *SIAM Journal on Imaging Sciences* [online]. 2009, **2**(2), 730-762

- [cit. 2017-04-08]. DOI: 10.1137/080727749. ISSN 1936-4954. Dostupné z: <http://epubs.siam.org/doi/10.1137/080727749>
- [42] CHEN, Peijun, Jianguo HUANG a Xiaoqun ZHANG. A primal–dual fixed point algorithm for convex separable minimization with applications to image restoration. *Inverse Problems* [online]. 2013, **29**(2), 025011- [cit. 2017-04-16]. DOI: 10.1088/0266-5611/29/2/025011. ISSN 02665611. Dostupné z: <https://goo.gl/dO02nS>
- [43] UNLocBoX: Matlab convex optimization toolbox. *SIGNAL PROCESSING LABORATORY LTS2* [online]. 2015 [cit. 2016-12-12]. Dostupné z: <https://lts2.epfl.ch/unlocbox/>
- [44] PROX_TV: Total variation proximal operator. *SIGNAL PROCESSING LABORATORY LTS2* [online]. 2015 [cit. 2016-12-13]. Dostupné z: <https://goo.gl/LhXRKj>
- [45] HENIN, Simon. Simonhenin/columnlegend. *MathWorks* [online]. The Mathworks, 1994-2017 [cit. 2017-04-08]. Dostupné z: <https://goo.gl/K99oqH>
- [46] DOKE, Jiro. Custom GINPUT. *MathWorks* [online]. The MathWorks, 1994-2017 [cit. 2017-04-08]. Dostupné z: <https://goo.gl/nsj6Pd>
- [47] LINDNER, Moshe. Animated-gif creator. *MathWorks* [online]. The MathWorks, 1994-2017 [cit. 2017-04-08]. Dostupné z: <https://goo.gl/4tYEhI>
- [48] LOPATA, Jan. *Odstraňování artefaktů JPEG komprese obrazových dat*. Praha, 2014. Diplomová práce. Univerzita Karlova. Vedoucí práce RNDr. Václav Kučera, Ph.D.
- [49] ČÍKA, Petr. *Multimédia: cvičení*. Brno, 2014. Skripta. VUTBR.
- [50] WANG, Zhou, Alan C. BOVIK, Hamid R. SHEIKH a Eero P. SIMONCELLI. *The SSIM Index for Image Quality Assessment* [online]. [cit. 2017-04-22]. Dostupné z: <http://www.cns.nyu.edu/~lcv/ssim/>
- [51] ZHOU WANG, A.C. BOVIK, H.R. SHEIKH a E.P. SIMONCELLI. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on* [online]. USA: IEEE, 0404n. 1., **13**(4), 600-612 [cit. 2017-04-22]. DOI: 10.1109/TIP.2003.819861. ISSN 10577149.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

| | |
|-----------------|--|
| AC | střídavé koeficienty – alternate coefficient |
| B | blue – modrá barevná složka |
| bpp | bitů na pixel – bit per pixel |
| BZSG | Základy počítačové sazby a grafiky |
| C_b | modrá chrominanční složka |
| C_r | červená chrominanční složka |
| DC | stejnoseměrné koeficienty – direct coefficient |
| DCT | diskrétní cosinová transformace – discrete cosine transformation |
| DP | diplomová práce – diploma thesis |
| DR | Douglas-Rachford |
| EOB | konec bloku – end of block |
| EPP | error per pixel – průměrná chyba na pixel |
| G | green – zelená barevná složka |
| I | jednotková matice |
| IDCT | inverzní DCT – inverse DCT |
| JPEG | Joint Photographic Experts Group |
| JFIF | JPEG File Interchange Format |
| ker | jádro lineárního zobrazení |
| MGMP | Moderní počítačová grafika |
| MSE | mean squared error – střední kvadratická chyba |
| NP | nedeterministická polynomiální složitost |
| NSP | Null Space Property – vlastnost nulového prostoru |
| prox_f | proximální operátor funkce f |
| PSNR | peak signal to noise ratio – špičkový poměr signálu k šumu |
| px | pixel – obrazový bod |
| Q | hodnota kvalitativního faktoru JPEG |
| R | red – červená barevná složka |
| REPP | relative error per pixel – relativní průměrná chyba na pixel |
| RGB | barevný model RGB |
| rank | hodnost matice |
| RIP | Restricted Isometry Property – vlastnost zeslabené isometrie |
| soft | soft thresholding – měkké prahování |
| spark | vlastnost matice týkající se lineární závislosti sloupců |
| SSIM | structural similarity index – index strukturální podobnosti |
| SUBJ | vyhodnocení na základě subjektivního dojmu |
| supp | nosič vektoru |
| TV | totální variace |

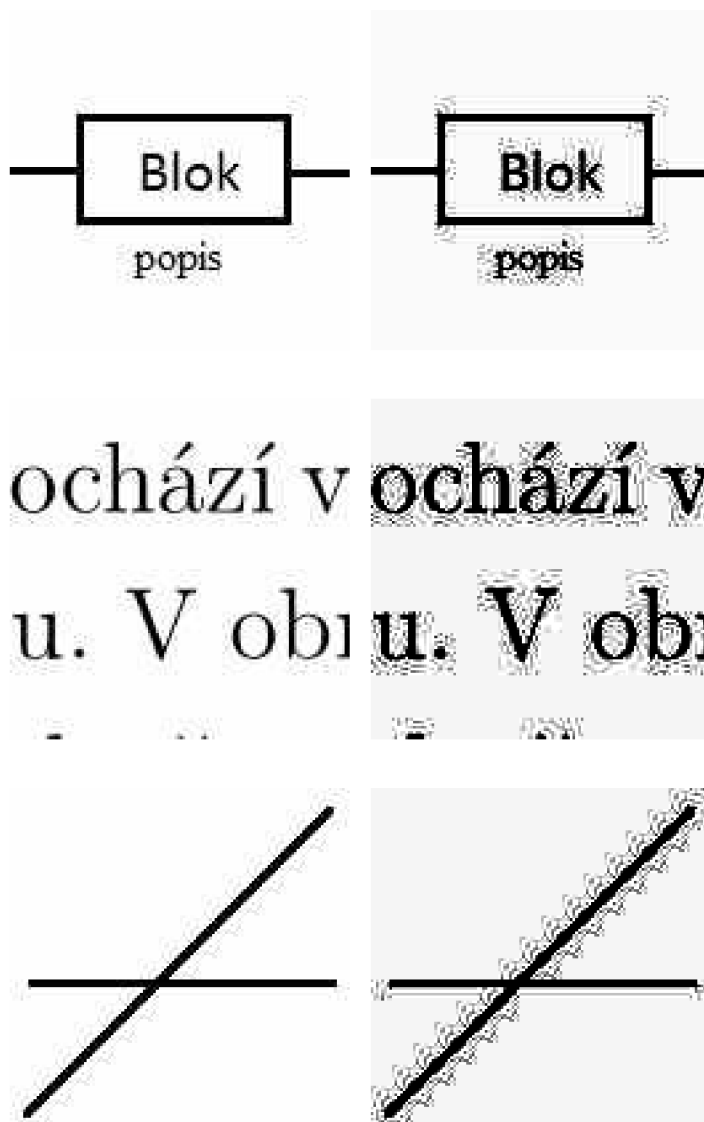
| | |
|---------|--|
| vec | operátor vektorizace matice |
| vlc | variable length coding – kódování s proměnnou délkou |
| Y | jasová složka |
| $YCbCr$ | barevný model $YCbCr$ |
| ZRL | zero run length |

SEZNAM PŘÍLOH

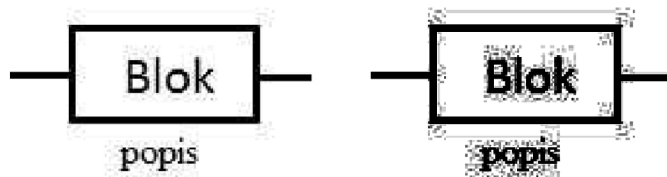
| | |
|--------------------------------------|----|
| A Rozdíl mezi obrázky | 75 |
| B Obsah přiloženého disku | 79 |
| C Hodnoty EPP, REPP, MSE, PSNR, SSIM | 83 |

A ROZDÍL MEZI OBRÁZKY

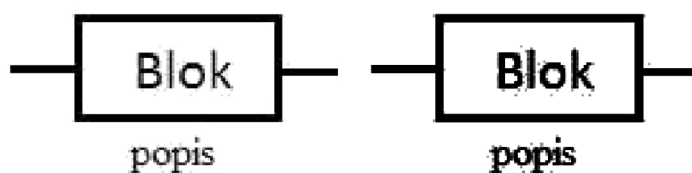
Ukázka původních obrázků v porovnání s upravenými pro tistěnou verzi této práce.



Obr. A.1: Zobrazení rozdílu mezi originálem a upraveným obrázkem pro obr. 2.12. Obrázky v levé části jsou původní a v pravé části jde o obrázky upravené pro tisk.

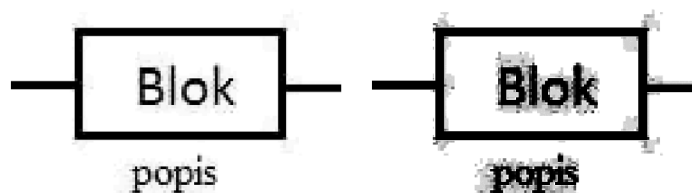


(a) poškozený ($Q = 5$) (b) poškozený ($Q = 5$)



(c) restaurovaný (d) restaurovaný

Obr. A.2: Zobrazení rozdílu mezi originálem a upraveným obrázkem pro obr. 5.2. Obrázky v levé části jsou původní a v pravé části jde o obrázky upravené pro tisk.

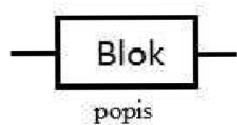


(a) originál (b) upravený

Obr. A.3: Zobrazení rozdílu mezi originálem a upraveným obrázkem pro obr. 5.2. Obrázek A.3a v levé části je původní a v pravé části jde o obrázek upravený pro tisk.

JPEG kodek, Q=5

PSNR: TV restaurace



SSIM: TV restaurace

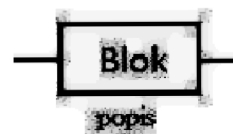
SUBJ: TV restaurace



(a) porovnání kvality zobrazení

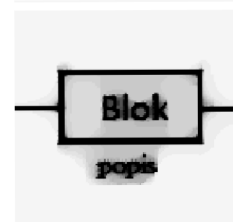
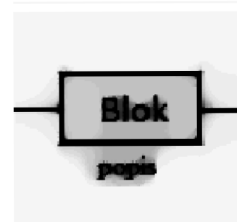
JPEG kodek, Q=5

PSNR: TV restaurace



SSIM: TV restaurace

SUBJ: TV restaurace



(b) porovnání kvality zobrazení

Obr. A.4: Zobrazení rozdílu mezi originálem a upraveným obrázkem pro obr. 6.3. Obrázky v horní části jsou původní a v dolní části upravené pro tisk.



(a) originál



(b) upravený

Obr. A.5: Zobrazení rozdílu mezi originálem a upraveným obrázkem pro obr. 6.6. Obrázek v horní části je původní a v dolní části jde o obrázek upravený pro tisk.

B OBSAH PŘILOŽENÉHO DISKU

Příložené CD obsahuje zdrojové kódy vytvořené a otestované v programovém prostředí MatLab (R2016a):

| | |
|--------------------------------------|---|
| / | kořenový adresář příloženého CD |
| └─ matlab | adresář se zdrojovými kódy |
| ├─ columnlegend | knihovna z www.mathworks.com |
| │ └─ columnlegend.m | Poupravená knihovna. |
| │ └─ license.txt | |
| │ └─ simonhenin-columnlegend.zip | Staženo z www.mathworks.com |
| │ └─ zdroj.txt | Soubor obsahující odkaz na stažení knihovny. |
| ├─ externi | Knihovna z www.mathworks.com. |
| │ └─ image2animation.m | Knihovna pro vytváření gif animací. |
| │ └─ zdroj.txt | Soubor obsahující odkaz na stažení knihovny. |
| ├─ ginputc | Knihovna z www.mathworks.com. |
| │ └─ ginputc.m | upravená knihovna ginput. |
| │ └─ ginputc.zip | Staženo z www.mathworks.com. |
| │ └─ zdroj.txt | Soubor obsahující odkaz na stažení knihovny. |
| │ └─ license.txt | |
| ├─ jpeg | Část knihovny pro podporu výuky předmětů MGMP a BZSG. |
| │ └─ demo_jpeg.m | Ukázka funkce JPEG kodéru/dekodéru. |
| │ └─ demo_jpeg_alternativni_funkce.m | Ukázka funkce alternativního JPEG kodéru/dekodéru. |
| │ └─ demo_jpeg_uprav.m | Upravené demo JPEG kodéru/dekodéru. |
| │ └─ demo_jpeg_uprav_2.m | Upravené demo JPEG kodéru/dekodéru. |
| │ └─ extrect_huffman_table.m | |
| │ └─ huffman_table_ac_lum.m | |
| │ └─ huffman_table_dc_lum.m | |
| │ └─ imColorSep.m | |
| │ └─ inzigzag.m | |
| │ └─ jedn_doplnek.m | |
| │ └─ jpeg_decode.m | Upravený dekodér pro potřeby DP. |
| │ └─ jpeg_encode.m | Upravený dekodér pro potřeby DP. |
| │ └─ quant_matrix.m | |
| │ └─ zdroj.txt | Soubor obsahující odkaz na stažení knihovny. |
| │ └─ zigzag.m | |
| ├─ mereni_kvality | Složka obsahuje funkce pro měření kvality. |
| │ └─ aepp.m | Průměrná chyba na pixel. |
| │ └─ delta_vypocet.m | Výpočet delty pro SSIM. |
| │ └─ delta_xy_vypocet.m | Výpočet delty _{xy} pro SSIM. |
| │ └─ mse_custom.m | Střední kvadratická chyba implementace pro PSNR. |
| │ └─ psnr_custom.m | Peak Signal to Noise Ratio [dB]. |
| │ └─ raepm.m | Relativní chyba na pixel. |
| │ └─ ssim_custom.m | Implementace structural similarity index. |
| │ └─ suma_pixelu.m | Suma hodnot pixelů v obrázku. |

```

test.....Adresář připravený pro vyzkoušení funkce
vytvor_testovaci_obrazky.m.
├── blok_diagram_
│   └── blok_diagram_orig.png..... Originální soubor.
├── cary_
│   └── cary_orig.png..... Originální soubor.
├── lena_
│   └── lena_orig.png..... Originální soubor.
├── letter_
│   └── letter_orig.png..... Originální soubor.
└── text_
    └── text_orig.png..... Originální soubor.
test_data.....Adresář obsahující podsložky s testovacími daty.
├── blok_diagram_
│   ├── blok_diagram_.mat..... Tensor s daty pro blokový diagram.
│   ├── blok_diagram_*.jpg. Náhledové obrázky pro jednotlivé nastavení
│   │   kvality.
│   ├── blok_diagram_orig.png..... Originální soubor.
│   └── zvolene_body.mat..... Tensor zvolených bodů.
├── cary_
│   ├── cary_.mat..... Tensor s daty pro blokový diagram.
│   ├── cary_*.jpg...Náhledové obrázky pro jednotlivé nastavení kvality.
│   ├── cary_orig.png..... Originální soubor.
│   └── zvolene_body.mat..... Tensor zvolených bodů.
├── lena_
│   ├── lena_.mat..... Tensor s daty pro blokový diagram.
│   ├── lena_*.jpg...Náhledové obrázky pro jednotlivé nastavení kvality.
│   ├── lena_orig.png..... Originální soubor.
│   └── zvolene_body.mat..... Tensor zvolených bodů.
├── letter_
│   ├── letter_.mat..... Tensor s daty pro blokový diagram.
│   ├── letter_*.jpg.Náhledové obrázky pro jednotlivé nastavení kvality.
│   ├── letter_orig.png..... Originální soubor.
│   └── zvolene_body.mat..... Tensor zvolených bodů.
├── text_
│   ├── text_.mat..... Tensor s daty pro blokový diagram.
│   ├── text_*.jpg...Náhledové obrázky pro jednotlivé nastavení kvality.
│   ├── text_orig.png..... Originální soubor.
│   └── zvolene_body.mat..... Tensor zvolených bodů.
├── blok_animace.gif
├── cary_animace.gif
├── lena_animace.gif
├── letter_animace.gif
└── text_animace.gif
unlocbox ..... Adresář obsahuje rozbalený UNLocBoX verze 1.7.3, zip
s UNLocBoXem v této verzi a odkaz na stažení.

```

vytvoreno Adresář s vytvořenými zdrojovými kódy.

- demo_DP.m . Souhrnná ukázka vytvořených funkcí, včetně přehledných výsledků.
- demo_DR_chart_data.m Funkce slouží ke sledování rychlosti konvergence Douglas-Rachford algoritmu při určitém nastavení parametru γ .
- demo_DR_nalezenej_nejlepsiho.m .. Zde je možné procházet výsledky pomocí šipek.
- demo_DR_test_manual.m Ukázka implementace Douglas-Rachford algoritmu.
- demo_epp_graf.m
- demo_jpeg_alternativni.m Ukázka funkce alternativního JPEG kodeku.
- demo_jpeg_DP.m
- demo_jpeg_DP_2.m
- demo_mekke_prahovani.m
- demo_mse_graf.m
- demo_norm_fro_graf.m Sledování závislosti Frobeniovi normy v závislosti na nastaveném kvalitativním faktoru JPEG.
- demo_PSNR_graf.m
- demo_PSNR_nejlepsi_nastaveni.m
- demo_SSIM_graf.m
- demo_SSIM_nejlepsi_nastaveni.m
- demo_SUBJ_nejlepsi_nastaveni.m
- demo_vliv_lambda.m ... Sledování vlivu regularizačního parametru λ .
- demo_zobrazeni_nejlepsich_vysledku.m
- demo_zobrazeni_nejlepsich_vysledku_lena.m
- demo_zobrazeni_rozdilu_mezi_originalem_a_degradovany.m
- doplň_body.m Funkce sloužící k doplnění předem zvolených bodů do dekódovaného jpeg souboru.
- DR_chart_help.m Pomocná funkce pro DR_chart_data.m.
- DR_jpeg.m Podpůrná funkce pro demo_jpeg_DP.m.
- DR_reconstruct.m .. Rekonstrukce obrázku pomocí Douglas-Rachford algoritmu, využito ve funkci vyhodnoceni_vysledku_komplet.m.
- DR_threshold.m Prahování pro výsledky Douglas-Rachford algoritmu.
- hledani_nejlepsi_konfigurace.m Pro prox_{TV} .
- PSNR_look_up_pracovni_prox_tv.m
- SSIM_look_up_pracovni_prox_tv.m
- SUBJ_gamma_look_up_pracovni_prox_tv
- SUBJ_gamma_look_up_prox_tv
- vyber_bodu.m Tato funkce slouží k výběru bodů v obrázku.
- vyhodnoceni_chybovosti.m . Tato funkce vrací hodnotu EPP, REPP, MSE, PSNR a SSIM.
- vyhodnoceni_vysledku.m Vyhodnocení výsledků pro konkrétní rekonstrukci.

- └─ vyhodnoceni_vysledku_komplet.m Porovnání výsledků pro jednotlivé metody pomocí tabulky.
- └─ vysledek_demo_jpeg_DP_2.m Funkce pro předání výsledku demo_jpeg_DP_2.
- └─ vytvor_testovaci_obrazky.m
- └─ zobraz_vysledky.m
- └─ zobrazeni_vysledku_komplet.m.. Zobrazení porovnání výsledků pro jednotlivé metody.
- └─ readme.txt
- └─ startup_grafika.m Upravený spouštěcí soubor z knihovny jpeg.
- └─ videa Adresář s vytvořenými videi (odzkoušeno v přehrávači VLC).
 - └─ demo_DP.mp4
 - └─ demo_DR_chart_data.mp4
 - └─ demo_DR_nalezenej_nejlepsiho.mp4
 - └─ demo_DR_test_manual.mp4
 - └─ demo_epp_graf.mp4
 - └─ demo_jpeg_alternativni.mp4
 - └─ demo_jpeg_DP.mp4
 - └─ demo_jpeg_DP_2.mp4
 - └─ demo_mekke_prahovani.mp4
 - └─ demo_mse_graf.mp4
 - └─ demo_norm_fro_graf.mp4
 - └─ demo_PSNR_graf.mp4
 - └─ demo_PSNR_nejlepsi_nastaveni.mp4
 - └─ demo_SSIM_graf.mp4
 - └─ demo_SSIM_nejlepsi_nastaveni.mp4
 - └─ demo_SUBJ_nejlepsi_nastaveni.mp4
 - └─ demo_vliv_lambda.mp4
 - └─ demo_zobrazeni_nejlepsich_vysledku.mp4
 - └─ demo_zobrazeni_nejlepsich_vysledku_lena.mp4
 - └─ demo_zobrazeni_rozdilu_mezi_originalem_a_degradovanim.mp4
 - └─ vytvor_testovaci_obrazky.mp4
- └─ JirakJ_release.pdf

C HODNOTY EPP, REPP, MSE, PSNR, SSIM

Tab. C.1: Tabulka hodnot pro blokový diagram.

| | | | | | | | | | | |
|-------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|
| Q | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| EPP | 31,7238 | 9,8139 | 7,4749 | 7,8116 | 11,67 | 5,6763 | 5,2682 | 6,1178 | 7,419 | 3,9848 |
| MSE | 1858,7 | 713,7 | 517,2 | 379,1 | 406,7 | 303,1 | 251,9 | 225,5 | 193,9 | 168,7 |
| PSNR | 15,4214 | 19,6404 | 20,9403 | 22,3669 | 22,0383 | 23,3144 | 24,1184 | 24,5999 | 25,2543 | 25,8587 |
| SSIM | 0,7869 | 0,8413 | 0,8881 | 0,895 | 0,9094 | 0,932 | 0,9206 | 0,9362 | 0,9389 | 0,9316 |
| Q | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| EPP | 4,2322 | 4,5 | 5,5765 | 3,4848 | 3,9408 | 4,5697 | 3,1304 | 3,8372 | 3,7011 | 4,071 |
| MSE | 165,6 | 144,7 | 132,2 | 122,9 | 115,7 | 112,8 | 107,7 | 103,7 | 95,8 | 90,3 |
| PSNR | 25,9408 | 26,5265 | 26,9191 | 27,2361 | 27,4961 | 27,6076 | 27,8091 | 27,9714 | 28,3165 | 28,5744 |
| SSIM | 0,9261 | 0,9389 | 0,949 | 0,9431 | 0,9542 | 0,9518 | 0,9544 | 0,9606 | 0,9634 | 0,9689 |
| Q | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| EPP | 2,6004 | 3,3262 | 2,3982 | 2,4286 | 2,2816 | 2,441 | 2,3151 | 2,2989 | 2,9244 | 2,2254 |
| MSE | 83 | 80,4 | 72,3 | 70,9 | 67,1 | 69,5 | 66,6 | 64,6 | 63,5 | 62,7 |
| PSNR | 28,9403 | 29,0799 | 29,5375 | 29,6232 | 29,8649 | 29,7122 | 29,894 | 30,0265 | 30,1026 | 30,1602 |
| SSIM | 0,9681 | 0,9696 | 0,971 | 0,9711 | 0,9729 | 0,9649 | 0,9709 | 0,9737 | 0,9757 | 0,9732 |
| Q | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| EPP | 2,2532 | 2,2064 | 2,7675 | 2,65 | 2,6077 | 2,5247 | 2,4821 | 2,5286 | 2,5769 | 1,8408 |
| MSE | 60,1 | 60 | 54,4 | 51,4 | 50,1 | 47 | 45,1 | 45,4 | 45 | 43,3 |
| PSNR | 30,3428 | 30,3482 | 30,7716 | 31,0217 | 31,1348 | 31,4071 | 31,5906 | 31,5562 | 31,5985 | 31,7616 |
| SSIM | 0,974 | 0,9724 | 0,9767 | 0,9781 | 0,9784 | 0,9784 | 0,9796 | 0,9779 | 0,979 | 0,9772 |
| Q | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| EPP | 1,8647 | 2,4731 | 2,467 | 2,5691 | 2,4457 | 1,6404 | 1,6815 | 1,6545 | 2,2751 | 2,2827 |
| MSE | 43,4 | 43,4 | 41,8 | 41,5 | 40,1 | 38,6 | 38,3 | 38,2 | 37,4 | 36,8 |
| PSNR | 31,7576 | 31,7601 | 31,9152 | 31,9498 | 32,1003 | 32,2613 | 32,2999 | 32,3131 | 32,4077 | 32,4742 |
| SSIM | 0,9774 | 0,9783 | 0,9785 | 0,9775 | 0,9774 | 0,9773 | 0,9768 | 0,9777 | 0,9804 | 0,9808 |
| Q | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| EPP | 2,2491 | 1,5162 | 1,5027 | 1,5649 | 1,564 | 1,5528 | 1,4898 | 1,4807 | 1,4265 | 1,435 |
| MSE | 34,9 | 33,1 | 32,5 | 31,6 | 30,2 | 29 | 28,4 | 27,6 | 26 | 24,7 |
| PSNR | 32,6991 | 32,927 | 33,0119 | 33,1273 | 33,3358 | 33,5051 | 33,5932 | 33,7295 | 33,977 | 34,2066 |
| SSIM | 0,9807 | 0,9795 | 0,9801 | 0,982 | 0,9824 | 0,9837 | 0,9848 | 0,9845 | 0,9853 | 0,9846 |
| Q | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| EPP | 1,4094 | 1,4063 | 1,4264 | 1,3839 | 1,9668 | 1,959 | 1,9724 | 1,2672 | 1,2292 | 1,1861 |
| MSE | 24 | 23,5 | 22,9 | 22,1 | 21,4 | 20,8 | 19,8 | 18,9 | 18,2 | 18 |
| PSNR | 34,3273 | 34,4291 | 34,5417 | 34,688 | 34,8189 | 34,9518 | 35,1588 | 35,3705 | 35,5211 | 35,585 |
| SSIM | 0,9847 | 0,9834 | 0,984 | 0,9845 | 0,9852 | 0,9848 | 0,9858 | 0,9855 | 0,9867 | 0,9869 |
| Q | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| EPP | 1,15 | 1,1876 | 1,1826 | 1,0774 | 1,0452 | 1,0206 | 1,0737 | 1,0293 | 1,1129 | 1,1091 |
| MSE | 17 | 16,5 | 15,6 | 14,3 | 13,7 | 13,8 | 14,9 | 14,6 | 15,3 | 15,2 |
| PSNR | 35,8238 | 35,9509 | 36,2095 | 36,585 | 36,7585 | 36,7196 | 36,3874 | 36,5015 | 36,2838 | 36,3003 |
| SSIM | 0,987 | 0,9873 | 0,9882 | 0,9881 | 0,9886 | 0,9877 | 0,9859 | 0,9861 | 0,9859 | 0,9859 |
| Q | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| EPP | 1,0802 | 1,0474 | 1,1038 | 1,0325 | 1,0126 | 1,0341 | 0,9429 | 0,9382 | 0,9241 | 0,9155 |
| MSE | 14,8 | 15,3 | 15,4 | 14,6 | 13,6 | 13,6 | 12,5 | 12,3 | 11,9 | 11,3 |
| PSNR | 36,4155 | 36,2717 | 36,2588 | 36,485 | 36,8013 | 36,8109 | 37,1542 | 37,2189 | 37,3891 | 37,6025 |
| SSIM | 0,9861 | 0,9867 | 0,9852 | 0,9869 | 0,9879 | 0,9889 | 0,9887 | 0,9901 | 0,991 | 0,9913 |
| Q | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| EPP | 0,9209 | 0,8622 | 0,8625 | 0,8548 | 0,8388 | 0,8223 | 0,8137 | 0,8072 | 0,7878 | 0,7911 |
| MSE | 11,4 | 10,7 | 10,8 | 10,5 | 10,1 | 9,9 | 9,6 | 9,7 | 9,5 | 9,4 |
| PSNR | 37,5667 | 37,839 | 37,7913 | 37,9012 | 38,0969 | 38,1908 | 38,3005 | 38,2552 | 38,371 | 38,3929 |
| SSIM | 0,9912 | 0,99 | 0,9899 | 0,992 | 0,9918 | 0,9926 | 0,9923 | 0,9928 | 0,9928 | 0,9931 |