

OBSAH:

1	ÚVOD	5
2	ZÁKLADNÍ POJMY	6
2.1	Signál	6
2.2	2D Diracův impuls	6
2.3	2D Fourierova transformace	6
2.4	Konvoluce	7
2.5	Model zkreslení	7
2.6	Šum	8
3	DEKONVOLUCE	9
3.1	Bayesova věta	9
3.1.1	Obecný Bayesův vzorec	9
3.2	Bayesovská dekonvoluce	9
3.3	Dekonvoluční metody	10
3.3.1	Wienerův filtr (WF)	10
3.3.2	Maximum A Posteriori (MAP)	10
3.3.3	Maximum Likelihood Estimation (MLE)	11
4	STRÍDAVÝ MINIMALIZAČNÍ ALGORITMUS PRO MULTIKANÁLOVOU SLEPOU DEKONVOLUCI	12
4.1	Alternating Minimization Algorithm (AM)	12
4.1.1	Regularizace zkreslení $R(h)$	12
4.1.2	Regularizace obrazu $Q(f)$	13
4.1.3	Iterative Minimization Algorithm (IMA)	13
4.2	Optimalizace	13
4.2.1	Simplexová metoda	14
4.2.2	Metoda největšího spádu	15
4.2.3	Kvazi-Newtonova metoda	15
4.3	Realizace MBD-AM v Matlabu	16
4.3.1	Popis programu „MBD_AM_IMA_fh“	16
4.4	Realizace MBD-AM pomocí knihovny NAG C	21
4.4.1	Knihovna NAG C	21
4.4.2	Popis programu „optimalizace_h_ccc“ a „optimalizace_f_ccc“	22
4.4.3	MEX soubory	23
4.4.4	Kompilace MEX souborů	24
4.5	Realizace MBD-AM v grafickém rozhraní	25
4.6	Hodnocení kvality restaurační metody MBD-AM	26
4.6.1	Hodnocení MBD-AM z pohledu počtu vnějších iterací	26
4.6.2	Hodnocení MBD-AM z pohledu velikosti poměru signál/šum	29
4.6.3	Hodnocení MBD-AM z pohledu počtu kanálů	32
4.6.4	Hodnocení MBD-AM z pohledu typu optimalizace	34
4.6.5	Hodnocení MBD-AM z pohledu korekčních parametrů λ a γ	36
4.7	Aplikace restaurační metody MBD-AM na reálná data	37

5	ZÁVĚR	39
6	LITERATURA	41

Seznam obrázků:

Obrázek 2.1:	Konvoluční model zkreslení s aditivním šumem	8
Obrázek 4.1:	Originální obraz před a po zkreslení pěti různými PSF	17
Obrázek 4.2:	Grafické rozhraní programu na výpočet MBD-AM	25
Obrázek 4.3:	Porovnání kvality MBD-AM pro různý počet vnějších iterací a „Testovací“ obraz	27
Obrázek 4.4:	Graf závislosti střední kvadratické chyby na počtu vnějších iterací	28
Obrázek 4.5:	Porovnání kvality MBD-AM pro různý počet vnějších iterací a obraz „Kameraman“	28
Obrázek 4.6:	Ukázka vstupních „Testovacích“ obrazů zkreslených PSF a poškozených šumem	29
Obrázek 4.7:	Porovnání kvality MBD-AM z pohledu velikosti SNR pro „Testovací“ obraz	30
Obrázek 4.8:	Graf závislosti střední kvadratické chyby na velikosti SNR	30
Obrázek 4.9:	Ukázka vstupních obrazů „Kameraman“ zkreslených PSF a poškozených šumem	31
Obrázek 4.10:	Porovnání kvality MBD-AM pro různé velikosti SNR a obraz „Kameraman“	31
Obrázek 4.11:	Porovnání kvality MBD-AM pro různý počet kanálů a „Testovací“ obraz	32
Obrázek 4.12:	Graf závislosti středních kvadratických chyb na počtu kanálů	33
Obrázek 4.13:	Porovnání kvality MBD-AM pro různý počet kanálů a obraz „Kameraman“	33
Obrázek 4.14:	Výsledný „Testovací“ obraz simplexové metody	34
Obrázek 4.15:	Porovnání kvality MBD-AM pro různý typ optimalizace a „Testovací“ obraz	35
Obrázek 4.16:	Porovnání kvality MBD-AM pro různý typ optimalizace a obraz „Kameraman“	36
Obrázek 4.17:	Porovnání kvality MBD-AM s ohledem na změnu parametru γ	36
Obrázek 4.18:	Reálná data	37
Obrázek 4.19:	Zkreslené obrazy	37
Obrázek 4.20:	Restaurované obrazy	38

Seznam tabulek:

Tabulka 4.1:	Hodnoty PMSE a doby výpočtu „Testovacího“ obrazu pro i iterací	27
Tabulka 4.2:	Hodnoty PMSE a doby výpočtu obrazu „Kameraman“ pro i iterací	27
Tabulka 4.3:	Hodnoty PMSE pro „Testovací“ obraz s ohledem na šum	29
Tabulka 4.4:	Hodnoty PMSE pro obraz „Kameraman“ s ohledem na šum	30
Tabulka 4.5:	Hodnoty PMSE s ohledem na počet kanálů pro „Testovací“ obraz	32
Tabulka 4.6:	Hodnoty PMSE s ohledem na počet kanálů pro obraz „Kameraman“	33
Tabulka 4.7:	Hodnoty PMSE optimalizační simplexové metody pro „Testovací“ obraz	34
Tabulka 4.8:	Hodnoty PMSE pro různý typ optimalizace a „Testovací“ obraz	35
Tabulka 4.9:	Hodnoty PMSE pro různý typ optimalizace a obraz „Kameraman“	35

Seznam zkratek:

FT: Fourierova transformace (Fourier Transform)

1D: One Dimensional

2D: Two Dimensional

PSF: Point Spread Function

WF: Wienerův filtr

WKF: Wienerův korekční faktor

MAP: Maximum A Posteriori

MLE: Maximum Likelihood Estimation

IMA: Iterative Minimization Algorithm

MBD-AM: Alternating Minimization Algorithm for Multichannel Blind Deconvolution

TV: Total variation

NAG: Numerical Algorithm Group

MEX: Matlab Executable

BFGS: Broyden, Fletcher, Goldfarb, Shanno (jedna z kvazi-newtonovských metod, pojmenovaná dle autorského kolektivu)

1 ÚVOD

Častou úlohou zpracovávání signálů je obnovení neznámého signálu (obrazu) z jeho zkreslené a šumem rušené podoby tak, aby se co nejvíce podobal původnímu originálnímu signálu (obrazu). V případě, že u zkresleného obrazu převažuje vliv lineárního zkreslení systému, jímž signál prošel, nad šumem, jedná se o dekonvoluci. Dekonvoluce jsou restaurační metody, obecně se dají rozdělit do dvou skupin, a to na neslepé dekonvoluce, které při výpočtu používají fixní znalost impulsní charakteristiky, ať už získanou měřením, nebo použitím teoretického modelu. Naproti tomu jsou metody slepé dekonvoluce vycházející z úplné neznalosti impulsní charakteristiky, nebo používají pouze částečné informace o impulsní charakteristice, které se navíc v průběhu výpočtu mění.

Jedny z metod dekonvoluce obrazů, jsou založeny na Bayesovské logice, jež popisuje dekonvoluci jako pravděpodobnostní problém. Bayesovská logika představuje formu statistické analýzy, která kvantifikuje neurčitý vstup stanovením pravděpodobnosti jeho výskytu, a to s použitím předem známých souvisejících informací. Pokud nejsou k dispozici žádné informace, přiřadí se každému takovému vstupu stejná pravděpodobnost, protože jsou stejně možné. U slepé multikanálové dekonvoluce obrazů, kdy tedy nejsou známy impulsní charakteristiky, jsou apriorní informací zkreslené obrazy, jejichž počet odpovídá počtu kanálů. Dále se předpokládá, že impulsní charakteristiky zachovávají intenzitu obrazu a šum má nulovou střední hodnotu i nulový rozptyl. Tzv. střídavý minimalizační algoritmus pro multikanálovou slepou dekonvoluci obrazu (Alternating Minimization Algorithm for Multichannel Blind Deconvolution (MBD-AM)), na základě těchto poznatků a předpokladů, a s pomocí regularizace a teorie optimalizace hledá řešení, tedy původní originální obraz. Vhodným nástrojem pro realizaci této dekonvoluční metody je programové prostředí Matlab MathWorks se svými vlastními funkcemi, popřípadě i funkcemi z knihovny NAG C, jež lze v Matlabu spustit pomocí tzv. MEX souborů.

2 ZÁKLADNÍ POJMY

V této kapitole jsou uvedeny pojmy, které blíže uvádí do problematiky dekonvoluce.

2.1 SIGNÁL

Signál je fyzikální veličina nesoucí informaci o stavu systému, který jej generuje. Tato informace může být užitečná (užitečný signál) nebo může být její obsah nežádoucí (rušení, šum). Identifikace signálu spočívá v nalezení jeho matematického modelu, kterým je funkce nebo posloupnost $f(x)$ proměnné x . Signály mohou být spojité nebo diskrétní, jednorozměrné nebo vícerozměrné, reálné nebo komplexní, stochastické nebo deterministické, sudé nebo liché, periodické nebo neperiodické, kauzální nebo nekauzální, viz [1]. V této diplomové práci se uvažuje 2D (two-dimensional) signál, neboli statický obraz jako jasová funkce dvou prostorových souřadnic $f(m,n)$, kde m, n určují rozměry obrazu [1] [2]:

$$m \in \langle -m_{\max}, m_{\max} \rangle \quad n \in \langle -n_{\max}, n_{\max} \rangle$$

2.2 2D DIRACŮV IMPULS

Mezi důležité dvourozměrné signály patří *2D Diracův impuls* $\delta(m,n)$, který lze získat vynásobením dvou 1D Diracových funkcí, neboli dvou nekonečně vysokých a nekonečně úzkých obdélníkových impulsů, jejichž plochy jsou rovny jedné. Pro 2D Diracův impuls platí:

$$\delta(m,n) = \begin{cases} \infty & \text{pro } m = 0 \wedge n = 0 \\ 0 & \text{pro } m \neq 0 \vee n \neq 0 \end{cases} \quad (2.1)$$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(m,n) \partial m \partial n = 1 \quad (2.2)$$

Důležité vlastnosti 2D Diracova impulsu jsou:

- filtrační vlastnosti delta distribuce

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(m,n) \delta(m-\xi, n-\eta) \partial m \partial n = f(\xi, \eta) \quad (2.3)$$

- posun do libovolné pozice

$$\delta(m-\xi, n-\eta) \quad (2.4)$$

- vyjádření pomocí 2D FT

$$\delta(m,n) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-j(um+vn)} \partial u \partial v \quad (2.5)$$

- odezvou systému (procesu zobrazení) na jednotkový (Diracův) impuls je impulzní charakteristika PSF (Point Spread Function – „Funkce rozprostření bodu“) procesu zobrazení [2] [6]

2.3 2D FOURIEROVA TRANSFORMACE

Často používanou a obecně nejznámější transformací ke zpracování obrazů je *2D Fourierova transformace*, která je zobecněním 1D FT.

$$F(u, v) = \text{FT}_{2\text{D}}\{f(m, n)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(m, n) e^{-j(um+vn)} \partial m \partial n \quad (2.6)$$

Jedná se tedy o zobrazení funkce (2D obrazu) $f(m, n)$ z originální roviny do spektrální roviny prostorových kmitočtů (u, v) . Výsledkem transformace je 2D spektrum. Vlastnosti Fourierovy transformace jsou obdobné jako u jednorozměrné transformace:

- FT je invertibilní

$$f(m, n) = \text{FT}_{2\text{D}}^{-1}\{F(u, v)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j(um+vn)} \partial u \partial v \quad (2.7)$$

- FT je lineární, platí princip superpozice
- Konvoluční vlastnost

$$\{f(m, n) * h(m, n)\} \xrightarrow{\text{FT}} \{F(u, v) H(u, v)\} \quad (2.8)$$

Konvoluce originálních funkcí v originální oblasti odpovídá součinu jejich spekter v kmitočtové oblasti a obráceně, součinu v originální oblasti odpovídá konvoluci spekter. Tato vlastnost je poměrně důležitá, neboť *konvoluce* se často používá u algoritmů pro zpracování dvourozměrného obrazu [2] [4].

2.4 KONVOLUCE

Konvoluce je vlastně matematickou operací zpracovávající dvě funkce, při níž vzniká funkce třetí. Pro spojité obrazy je definována konvolučním integrálem:

$$g(m, n) = h(m, n) * f(m, n) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(m - \xi, n - \eta) f(\xi, \eta) \partial \xi \partial \eta, \quad (2.9)$$

pro diskrétní pak:

$$g(m, n) = h(m, n) * f(m, n) = \sum_{\xi=-k}^k \sum_{\eta=-k}^k h(m - \xi, n - \eta) f(\xi, \eta), \quad (2.10)$$

kde $g(m, n)$ reprezentuje výsledný obraz konvoluce, $f(m, n)$ původní signál a $h(m, n)$ impulsní charakteristiku, respektive point spread function, a k pak rozměry diskrétního obrazu [5] [9].

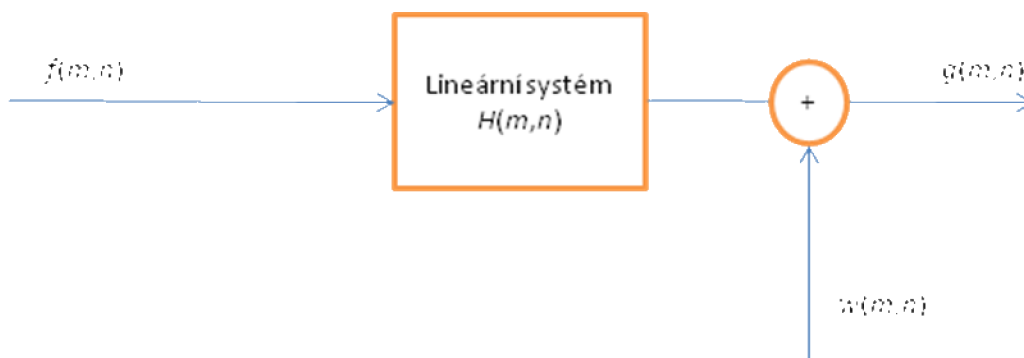
2.5 MODEL ZKRESLENÍ

Častou úlohou zpracovávání signálů je obnovení neznámého signálu z jeho zkreslené a šumem rušené podoby. **Obrázek 2.1** názorně ukazuje proces zkreslení původního obrazu $f(m, n)$ konvolucí s impulsní charakteristikou (PSF) $h(m, n)$ lineárního systému a následné zašumění obrazu aditivním šumem $w(m, n)$. Procesy zkreslení jsou popsány pro diskrétní obrazy rovnicí (2.11), pro spojité pak (2.12) [3].

$$g(m, n) = f(m, n) * h(m, n) + w(m, n) \quad (2.11)$$

$$g(m, n) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(m, n) h(m, n, \xi, \eta) \partial \xi \partial \eta + w(m, n) \quad (2.12)$$

Na výstupu tedy získáváme poškozený signál $g(m, n)$, který je značně odlišný od původního signálu $f(m, n)$.



Obrázek 2.1: Konvoluční model zkreslení s aditivním šumem

V případě, že u zkresleného obrazu $g(m,n)$ převažuje vliv šumu nad lineárním zkreslením systému, jde o restaurační úlohu „zlepšení poměru signál – šum“, v opačném případě se jedná o tzv. dekonvoluci, i zde se však bere v úvahu vliv šumu [3].

2.6 ŠUM

Nejčastěji se v modelech zkreslení setkáváme s Gaussovým nebo Poissonovým šumem.

Gaussov model šumu je aproximován pomocí Gaussova rozložení. Dvourozměrná náhodná veličina s Gaussovým normálovým rozložením má hustotu pravděpodobnosti danou vztahem:

$$p(f) = \prod_{m,n} \frac{1}{\sigma_f \sqrt{2\pi}} e^{-\frac{f^2(m,n)}{2\sigma_f^2}}, \quad (2.13)$$

kde σ_f je směrodatná odchylka.

Gaussov šum lze považovat za aditivní.

Poissonův model šumu odpovídá kvantové povaze světla, jeho rozložení je dáno:

$$p(N) = \prod_{i=1}^M \frac{\mu^N \exp(-\mu)}{N!}, \quad (2.14)$$

kde N je počet detekovaných fotonů a μ odchylka.

Poissonův šum již nelze považovat za aditivní [7].

3 DEKONVOLUCE

Dekonvoluce má za cíl obnovit zkreslený a zašuměný obraz $g(m,n)$ tak, aby se co nejvíce podobal původnímu originálnímu obrazu $f(m,n)$. Existuje velké množství jednotlivých dekonvolučních metod, obecně by se však dali rozdělit do dvou skupin, a to na *neslepé dekonvoluce*, které při výpočtu používají fixní znalost PSF, ať už získanou měřením, nebo použitím teoretického modelu. Naproti tomu jsou metody *slepé dekonvoluce* vycházející z úplné neznalosti PSF, nebo používají pouze částečné informace o PSF, které se navíc v průběhu výpočtu mění. Cílem této práce je blíže popsat jednotlivé neslepé i slepé metody dekonvoluce pomocí Bayesovy pravděpodobnosti [7].

3.1 BAYESOVA VĚTA

Thomas Bayes (1702-1761, anglický matematik) odvodil způsob výpočtu aposteriorních pravděpodobností pomocí apriorních pravděpodobností (Bayesův vzorec (3.1)). V bayesovském pojetí je pravděpodobnost náhodného jevu interpretována jako stupeň důvěry jedince v tento jev. V tomto pojetí je tedy pravděpodobnost náhodného jevu ovlivněna nejen podmínkami náhodného pokusu, ale také informacemi, které má konkrétní osoba k dispozici a jejím postojem k těmto informacím.

3.1.1 Obecný Bayesův vzorec

Obecný Bayesův vzorec vychází z pravidel pro násobení podmíněných a nepodmíněných pravděpodobností výskytu jevů A (hypotéza) a B (předpoklad):

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}, \quad (3.1)$$

kde $p(A|B)$ reprezentuje podmíněnou pravděpodobnost, že nastane jev A za předpokladu, že nastal jev B . Naopak $p(B|A)$ představuje pravděpodobnost jevu B v případě, že nastal jev A . $p(A)$ a $p(B)$ jsou nepodmíněné pravděpodobnosti výskytu jevu A a B . Pravděpodobnost hypotézy před provedením náhodného pokusu $p(B)$ se nazývá pravděpodobnost *a priori* a pravděpodobnost hypotézy po provedení náhodného pokusu $p(B|A)$ se nazývá pravděpodobnost *a posteriori* [8].

3.2 BAYESOVSKÁ DEKONVOLUCE

Bayesovská logika tedy představuje formu statistické analýzy, která kvantifikuje neurčitý vstup stanovením pravděpodobnosti jeho výskytu, a to s použitím předem známých souvisejících informací. Na základě této logiky lze popsat problém dekonvoluce jako pravděpodobnostní problém. Dosazením hledaného obrazu $f(m,n)$ i získaného zkresleného a zašuměného snímku $g(m,n)$ do vzorce (3.1) se získá:

$$p(f|g) = \frac{p(g|f)p(f)}{p(g)}, \quad (3.2)$$

což je řešení problému dekonvoluce pomocí Bayesova přístupu. (Ve vzorci (3.2) je pro přehlednost psáno jen f a g , myslí se tím ale samozřejmě 2D obrazy $f(m,n)$ a $g(m,n)$.) [7]

Pravděpodobnost $p(f)$ se nazývá **apriorní pravděpodobnost hypotéz** (potenciálních odhadovaných obrazů $f(m,n)$). V aplikacích Bayesova vzorce se hypotézy vzájemně vylučují a právě jedna je správná. Jsou to pravděpodobnosti hypotéz před pozorováním signálu. Pokud o nich nejsou žádné informace, přiřadí se každé hypotéze stejná pravděpodobnost, protože jsou *a priori* stejně možné. Podmíněná hustota pravděpodobnosti $p(g|f)$ je nazývána **pravděpodobnostní mírou** výsledku testu na datech $g(m,n)$ za předpokladu platnosti hypotézy

$f(m,n)$. Pokud se uvažují různé hypotézy pro stejný soubor dat, pak tento výraz vyjadřuje věrohodnostní funkci. Hustota pravděpodobnosti $p(g)$ je **úplnou pravděpodobností** (evidencí), neboť zkeslený obraz $g(m,n)$ bývá k dispozici. A podmíněná hustota pravděpodobnosti $p(f|g)$ je **aposteriorní pravděpodobnost** určená zkeslenými obrazy $g(m,n)$ a apriorní informací. Vzorec (3.2) tedy ukazuje, jak pravděpodobná je hypotéza $f(m,n)$ po pozorování signálu $g(m,n)$ [8]. Pro slepou dekonvoluci, kdy není znám ani obraz $f(m,n)$, ani PSF, se vzorec (3.2) vyjádří ve tvaru:

$$p(f, h|g) = \frac{p(g|f, h)p(f)p(h)}{p(g)}, \quad (3.3)$$

kde pravděpodobnost $p(h)$ je **apriorní pravděpodobnost hypotéz** PSF.

Dekonvoluční metoda vycházející z tohoto přístupu se nazývá *Maximum A Posteriori*. Řešení se hledá pomocí maximalizace vztahu (3.2), popřípadě (3.3) pro slepou dekonvoluci. (Podrobnější popis viz kapitola 3.3.2.)

3.3 DEKONVOLUČNÍ METODY

3.3.1 Wienerův filtr (WF)

Wienerův filtr je prostý inverzní filtr doplněný Wienerovým korekčním faktorem (WKF). V případě, že se uvažuje model zkeslení dle obrázku (viz **Obrázek 1.1**), systém $H(m,n)$ je časově invariantní a $f(m,n)$, $w(m,n)$ jsou stacionární procesy, pak WF ve frekvenční oblasti je popsán vztahem:

$$M(\omega) = \frac{H^*(\omega)S_{ff}(\omega)}{|H(\omega)|^2 S_{ff}(\omega) + S_{ww}(\omega)} = \frac{1}{H(\omega)} \frac{|H(\omega)|^2}{|H(\omega)|^2 + \frac{S_{ww}(\omega)}{S_{ff}(\omega)}}, \quad (3.4)$$

kde první zlomek reprezentuje prostý inverzní filtr a druhý zlomek Wienerův korekční faktor, jež je vždy reálný (nabývá hodnot $\langle 0,1 \rangle$), takže neovlivňuje fázovou charakteristiku filtru. Bude-li šum zanedbatelný, $WKF \rightarrow 1$ a WF bude roven prostému inverznímu filtru. Bude-li naopak šum velký nebo systém $H(m,n)$ nulový, $WKF \rightarrow 0$, a pak i $WF \rightarrow 0$. $H(\omega)$ je frekvenční charakteristika zkeslujícího systému. S_{ff} , S_{gg} a S_{ww} jsou pak výkonová spektra obrazu originálního, zkesleného a šumu. WF je navržen tak, aby minimalizoval střední kvadratickou chybu mezi zkesleným obrazem $g(m,n)$ a původním $f(m,n)$ [3].

$$\min E = \|g - f\|^2 \quad (3.5)$$

Praktičtější vyjádření WF bez výkonového spektra obvykle nedostupného originálu $f(m,n)$ [2]:

$$M(\omega) = \frac{1}{H(\omega)} \frac{S_{gg}(\omega) - S_{ww}(\omega)}{S_{gg}(\omega)} \quad (3.6)$$

3.3.2 Maximum A Posteriori (MAP)

Maximum A Posteriori je dekonvoluční metodou využívající Bayesova přístupu, jejíž řešení se hledá pro neslepou dekonvoluci pomocí maximalizace vztahu (3.2).

$$f_{MAP} = \arg\{\max_f p(f|g)\} \quad (3.7)$$

Vzorec (3.7) se následně rozepíše, oba členy se zlogaritmují přirozeným logaritmem a vyjádřením pomocí záporného znaménka se získá k MAP ekvivalentní minimalizační funkce:

$$\begin{aligned} f_{MAP} &= \arg\{\max_f p(g|f)p(f)\} = \\ &= \arg\{\max_f \ln p(g|f) + \ln p(f)\} = \\ &= \arg\{\min_f -\ln p(g|f) - \ln p(f)\}. \end{aligned} \quad (3.8)$$

Za $p(f)$ se dosadí vztah (2.13) a za $p(g|f)$:

$$p(g|f) = \prod_{m,n} \frac{1}{\sigma_w \sqrt{2\pi}} e^{-\frac{|g-f*h|^2}{2\sigma_w^2}}, \quad (3.9)$$

přičemž $w(m,n)$ se vyjádří ze vztahu (2.11).

A výsledkem je

$$\begin{aligned} f_{MAP} &= \arg\{\min_f -\ln \frac{1}{\sigma_w \sqrt{2\pi}} + \sum_{m,n} \frac{|g-f*h|^2}{2\sigma_w^2} - \ln \frac{1}{\sigma_f \sqrt{2\pi}} + \sum_{m,n} \frac{f^2}{2\sigma_f^2}\} \\ f_{MAP} &= \arg\{\min_f \sigma_f^2 \sum_{m,n} |g-f*h|^2 + \sum_{m,n} \sigma_w^2 |f|^2\}. \end{aligned} \quad (3.10)$$

kde člen ε_{ML} je označován jako maximální míra pravděpodobnosti (maximum likelihood term)

$$\varepsilon_{ML} = \sigma_f^2 \sum_{m,n} |g-f*h|^2 \quad (3.11)$$

a ε_{PR} jako regularizační člen (viz [11]).

$$\varepsilon_{PR} = \sum_{m,n} \sigma_w^2 |f|^2 \quad (3.12)$$

Řešením metody MAP je nalezení globálního minima pomocí vzorce (3.10).

Předpokládá se, že obraz $f(m,n)$ a šum $w(m,n)$ mají Gaussovské rozložení, tj. střední hodnoty μ_f a μ_w jsou rovny nule, σ_f a σ_w jsou směrodatné odchylky. Dále se předpokládá, že $w(m,n)$ je bílý šum a obraz $f(m,n)$ má vzájemně nezávislé pixely, čili vlastnosti bílého šumu. Odvození metody MAP pro slepou dekonvoluci lze nalézt např. v [10].

3.3.3 Maximum Likelihood Estimation (MLE)

Další dekonvoluční metodou je MLE (Maximum Likelihood Estimation – maximální pravděpodobnost odhadu), nejčastěji se používá pro odhad parametrů. Tato metoda vychází ze znalosti statistického modelu šumu a hustoty pravděpodobnosti $p(g|f)$ a na těchto základech restauruje původní obraz. Zjišťuje, která z hypotéz obrazu $f(m,n)$ je nejpravděpodobnější, jestliže na výstupu se objeví zkraslený obraz $g(m,n)$. Stejně jako u metody MAP se maximalizuje, tentokrát však hustota pravděpodobnosti $p(g|f)$.

$$f_{ML} = \arg\{\max_f p(g|f)\} \quad (3.13)$$

Po obdobné úpravě jako je v kapitole 3.3.2 a dosazení do vzorce pro případ bílého šumu vyjde:

$$f_{ML} = \arg\{\min_f \|g - f * h\|_2^2\}, \quad (3.14)$$

kde $\|f\|_2$ je L2 (Eukleidova) norma: $\|f\|_2 = \sqrt{\sum f_i^2}$

Pro případ Poissonova šumu lze odvození nalézt v [7], [12].

4 STRÍDAVÝ MINIMALIZAČNÍ ALGORITMUS PRO MULTIKANÁLOVOU SLEPOU DEKONVOLUCI

Tato kapitola pojednává o střídavém minimalizačním algoritmu pro multikanálovou slepou dekonvoluci obrazu (Alternating Minimization Algorithm for Multichannel Blind Deconvolution (MBD-AM)), jež byl zkreslen rovnoměrně zkreslujícími kanály. Druhá část kapitoly obsahuje realizaci MBD-AM v Matlabu a s využitím knihovny NAG C.

4.1 ALTERNATING MINIMIZATION ALGORITHM (AM)

Cílem této metody je najít řešení rovnice:

$$g_i(m, n) = (f * h_i)(m, n) + w_i(m, n). \quad (4.1)$$

Rozdíl ve srovnání s modelem zkreslení (2.11) je v indexu i , jež vyjadřuje počet zkreslujících kanálů, a tím i počet zkreslených obrazů $g_i(m, n)$, které jsou k dispozici. Protože se jedná o slepou dekonvoluci, je to zároveň i jediná apriorní informace, neboť není znám originální obraz $f(m, n)$, šum $w_i(m, n)$, ani jednotlivé PSF $h_i(m, n)$. Dále se předpokládá, že šum má nulovou střední hodnotu i nulový rozptyl a PSF zachovává intenzitu obrazu. Lze tedy psát:

$$\frac{1}{|\Omega|} \int_{\Omega} (f * h_i - g_i) dx = \sigma^2 \quad (4.2)$$

$$\int_{\Omega} h(x) dx = 1 \quad (4.3)$$

(pro zjednodušení $x \approx (m, n)$)
a rozměry obrazu (m, n) náleží do oblasti Ω :

$$g_i(m, n) = (f * h_i)(m, n) \Leftrightarrow (m, n) \in \Omega. \quad (4.4)$$

Nechť $Q(f)$ a $R(h)$ jsou regularizační funkce odhadovaného originálního obrazu $f(m, n)$ a PSF $h = \{h_1, \dots, h_N\}$. Pak minimalizační problém je formulován jako $\min_{f, h} Q(f) + \gamma R(h)$ vycházející z (4.2) a (4.3). Pomocí Langrangova multiplikátoru se převede na optimalizační úlohu, jejíž řešení je nalezení takových f a h , které minimalizují funkci:

$$E(f, h) = \frac{1}{2} \sum_{i=1}^N \|h_i * f - g_i\|^2 + \gamma R(h) + \lambda Q(f), \quad (4.5)$$

kde λ a γ jsou kladné parametry ovlivňující správnost řešení f a h .

4.1.1 Regularizace zkreslení $R(h)$

Regularizace jednotlivých PSF h_i je založena na předpokladu, že zkreslující kanály jsou vzájemně nezávislé, pak pro jakékoli h_i a h_j platí:

$$\|g_i * h_j - g_j * h_i\|^2 = 0, \quad (4.6)$$

jestliže šum ze vzorce modelu zkreslení (4.1) byl zanedbán. Regularizace zkreslení pak odpovídá vzorci:

$$R(h) = \frac{1}{2} \sum_{1 \leq i < j \leq N} \|g_i * h_j - g_j * h_i\|^2. \quad (4.7)$$

4.1.2 Regularizace obrazu $Q(f)$

Regularizace vzorce (4.1) s ohledem na funkci obrazu může být přiblížena buď klasickým přístupem, tzv. Tichonovou regularizací (4.8), nebo Total variation (TV), (4.9):

$$Q(f) = \int_{\Omega} (|\nabla f(x)|^2) dx, \quad (4.8)$$

kde ∇f značí gradient obrazu f . Tichonova regularizace však není vhodná pro jednoduché realizace, protože L2 norma obrazového gradientu koresponduje s hranami obrazu a dochází k vyhlazovacímu efektu. Naproti tomu dobré anizotropní vlastnosti má TV:

$$Q_{\phi}(f) = \int_{\Omega} \phi(|\nabla f(x)|) dx, \quad (4.9)$$

kde Φ je ryze konvexní neklesající lineárně rostoucí funkce [10].

Příklady $\Phi(s)$:

$$\begin{aligned} \Phi(s) &= \sqrt{1+s^2} - 1 \\ \Phi(s) &= \log(\cosh(s)) \\ \Phi(s) &= \log(1+s^2) \\ \Phi(s) &= \frac{s^2}{(1+s^2)} \\ \Phi(s) &= \arctan(s^2) \end{aligned} \quad (4.10)$$

4.1.3 Iterative Minimization Algorithm (IMA)

Dosazením (4.7) a (4.9) do (4.5) se získá minimalizační úloha:

$$E(f, h) = \frac{1}{2} \sum_{i=1}^N \|h_i * f - g_i\|^2 + \frac{\gamma}{2} \sum_{1 \leq i < j \leq N} \|g_i * h_j - g_j * h_i\|^2 + \lambda \int_{\Omega} \phi(|\nabla f(x)|) dx. \quad (4.11)$$

$E(f, h)$, jako funkce proměnných f a h v této podobě však není konvexní. Konvexnosti se dosáhne pro f tím, že h je fixně dáno, a obráceně (pro h , je-li dáno f). Minimalizační posloupnosti (f_n, h_n) mohou mít řešení postaveno na střídání dvou minimalizačních podúloh:

$$f^n = \arg \min_f E(f, h^{n-1}), \quad h^n = \arg \min_h E(f^n, h). \quad (4.12)$$

Nicméně ani tak nelze garantovat, že globální minimum bude nalezeno touto cestou. Testování však ukazuje dobré konvergenční vlastnosti algoritmu pro mnoho reálných úloh [10].

4.2 OPTIMALIZACE

Optimalizační úloha má obecně za cíl nalezení řešení (minima nebo maxima funkce), které je omezené vstupními podmínkami a zároveň nejlépe vyhovuje daným kritériím.

Vstupními prvky optimalizační úlohy jsou:

- vektor proměnných:

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^T,$$

kde $\mathbf{x} \in R^n$ (R^n značí n -rozměrný Euklidovský – kartézský prostor).

- omezující podmínky:

$$P \subseteq U,$$

kde P reprezentuje množinu přípustných řešení a U množinu všech možných řešení.

- kritériální (objektová) funkce:

$$E : U \rightarrow R \text{ (} R \text{ - množina reálných čísel),}$$

která přiřazuje každému řešení jeho hodnotu.

Optimální hodnoty kritériální funkce se pak hledají dle:

$$\min/\max_{x \in P} E(x). \quad (4.13)$$

U lineární optimalizační úlohy je kritériální funkcí lineární rovnice, jež se řeší simplexovým algoritmem. Komplexní kvadratická optimalizační úloha má kvadratickou konvexní kritériální funkci, řeší se Wolfeho algoritmem a vícekritériální optimalizační model pak využívá více kritériálních funkcí. Podrobněji o optimalizačních metodách viz [16], [17] a [14].

4.2.1 Simplexová metoda

Metoda simplexů (také známá jako Nelderova-Meadova metoda) je iterační výpočetní postup, jež hledá minimum funkce $E(x)$, kde $x = (x_1, x_2, \dots, x_n)^T$ reprezentuje vektor o n proměnných bez použití jejich derivací.

Metoda spočívá v posouvání vrcholů n -úhelníka v $n-1$ dimenzionálním prostoru (např. trojúhelníka - ve dvourozměrném prostoru) podle jistých pravidel směrem k minimu. Nejprve dochází k přenesení nejvyššího a tedy nejhůře položeného bodu souměrně k těžišti tělesa (reflexe):

$$E(x_r) < E(x_g) \text{ pro vrchol } x_g \neq x_w \Rightarrow x_w := x_r, \quad (4.14)$$

x_r je bod reflexe, x_w nejhůře položený bod a x_g jiný než nejhorší vrchol.

Pokud je tato poloha nejnižší ze všech ostatních, dochází k definování nového bodu x_e na místě dvojnásobné vzdálenosti od těžiště tělesa (expanze):

$$E(x_r) < E(x_b) \text{ a } E(x_e) < E(x_b) \Rightarrow x_w := x_e, \quad (4.15)$$

x_e je bod expanze, x_b nejlépe položený bod.

Jestliže je však tato poloha nejvyšší, dochází k umístění vrcholu doprostřed mezi původní vrchol a těžiště (kontrakce):

$$E(x_r) < E(x_w) \text{ a } E(x_{ce}) < E(x_r) \Rightarrow x_w := x_{ce}, \quad (4.16)$$

$$E(x_r) < E(x_w) \text{ a } E(x_{ci}) < E(x_w) \Rightarrow x_w := x_{ci},$$

x_{ce} a x_{ci} jsou body vnější a vnitřní kontrakce.

V případě, že je i nadále uvažovaný vrchol nejvyšší, dochází k posunu všech vrcholů x_i o polovinu směrem k nejnižšímu (redukce), [20]:

$$x_i := \frac{1}{2}(x_b + x_i) \text{ pro všechna } x_i \neq x_b, \quad (4.17)$$

Simplexová metoda se nazývá heuristickou, což znamená, že je založena kromě logického uvažování a zkušeností také na pozorování a experimentování. Metoda bývá velmi pomalá, zvláště pak v blízkosti minima. Také představuje velký počet operací, proto je vhodná spíše pro minimalizaci funkce o menším počtu proměnných $n \leq 10$. Mezi výhody této metody patří stabilita, a je tedy užitečná pro funkce podléhající nepřesnostem. Simplexová metoda je

reprezentovaná v knihovně NAG C funkcí e04ccc nebo nag_opt_simplex, v Matlabu pak funkcí fminsearch [16], [17], [20].

4.2.2 Metoda největšího spádu

Metoda největšího spádu se řadí mezi iterační gradientní metody, což znamená, že používá derivaci kritériální funkce:

$$G(\mathbf{x}) \equiv \nabla E(\mathbf{x}) = \left(\frac{\partial E(\mathbf{x})}{\partial x_1}, \frac{\partial E(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial E(\mathbf{x})}{\partial x_n} \right)^T. \quad (4.18)$$

Metoda je založena na předpokladu, že funkce $E(\mathbf{x})$ klesá nejrychleji ve směru záporného gradientu, a tedy spádový vektor \mathbf{d}_k bude ve tvaru:

$$\mathbf{d}_k = -G(\mathbf{x}_k), \quad (4.19)$$

kde k značí k -tou iteraci.

Na polopřímce $\mathbf{x}_k + s_k \mathbf{d}_k$, $s_k \geq 0$ se vybere bod:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + s_k \mathbf{d}_k, \quad (4.20)$$

pro který platí $E(\mathbf{x}_{k+1}) < E(\mathbf{x}_k)$. s_k udává délku kroku, tedy vzdálenost bodů \mathbf{x}_k a \mathbf{x}_{k+1} .

Metoda vychází z počátečního bodu \mathbf{x}_0 a postupuje po směrovém vektoru \mathbf{d}_0 , dokud nedosáhne nejnižšího bodu v tomto směru – \mathbf{x}_1 . Odtud pokračuje ve směru \mathbf{d}_1 , který je kolmý na směr \mathbf{d}_0 , dokud nedosáhne nejnižšího bodu – \mathbf{x}_2 . Výpočet se ukončí ve chvíli, kdy je dosaženo požadované tolerance přesnosti:

$$\|\nabla E(\mathbf{x}_{k+1})\| < \varepsilon. \quad (4.21)$$

Metoda největšího spádu je poměrně jednoduchá na realizaci s nízkou prostorovou složitostí. Mezi její hlavní nevýhody patří velmi pomalá konvergence v oblastech malého spádu a chyby způsobené zaokrouhlováním. Někdy se dokonce od globálního minima vzdaluje a konverguje k nějakému lokálnímu minimu. [20]

4.2.3 Kvazi-Newtonova metoda

Kvazi-Newtonova metoda se řadí k iteračním gradientním metodám. Vychází z klasické Newtonovy metody:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - H_k^{-1} G_k, \quad (4.22)$$

kde směrový vektor $\mathbf{d}_k = -H_k^{-1} G_k$ a H_k^{-1} je Hessova matice (druhých derivací) funkce $E(\mathbf{x})$:

$$G'(\mathbf{x}) \equiv H(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 E(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 E(\mathbf{x})}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 E(\mathbf{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 E(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 E(\mathbf{x})}{\partial x_2^2} & \dots & \frac{\partial^2 E(\mathbf{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 E(\mathbf{x})}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 E(\mathbf{x})}{\partial x_n^2} \end{pmatrix}. \quad (4.23)$$

Pokud je počáteční aproximace x_0 dostatečně blízko řešení, Newtonova metoda velmi rychle – kvadraticky konverguje. V opačném případě nemusí konvergovat vůbec. Další nevýhodou je její výpočetní náročnost, neboť v každé iteraci se musí počítat inverzní matice druhých derivací.

Tyto nedostatky se snaží řešit kvazi-newtonovské metody. Jednou z nich je tzv. BFGS (Broyden, Fletcher, Goldfarb, Shanno), pojmenovaná dle autorského kolektivu, který ji navrhl. K určení směru přímkové optimalizace slouží inverzní matice k hessiánu, k determinantu Hessovy matice. U metody BFGS odpadají problémy s derivací, neboť si sama numericky stanoví analytické derivace. Aproximace Hessovy matice má tvar:

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k^T s_k s_k^T H_k}{s_k^T H_k s_k}, \quad (4.24)$$

kde $s_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, $q_k = \nabla E(\mathbf{x}_{k+1}) - \nabla E(\mathbf{x}_k)$.

Matice H_k se volí tak, aby směrový vektor \mathbf{d}_k byl spádový. Např. pro $k = 0$ se $H_0 = I$, kde I reprezentuje jednotkovou matici, potom $\mathbf{d}_0 = -G(\mathbf{x}_0)$ odpovídá směru metody největšího spádu. Čím více se \mathbf{x}_k blíží optimu (minimu), tím lépe H_k aproximuje k inverzní matici H^{-1} , tudíž \mathbf{d}_k se přibližuje směrovému vektoru $-H_k^{-1}G_k$ Newtonovy metody [20], [14].

4.3 REALIZACE MBD-AM V MATLABU

Pro realizaci programu na výpočet MBD-AM (Alternating Minimization Algorithm for Multichannel Blind Deconvolution) bylo zvoleno programové prostředí Matlab, které díky své grafice a existenci knihoven Signal procesing toolbox a Optimization toolbox s řadou specializovaných funkcí je velmi vhodným prostředkem.

4.3.1 Popis programu „MBD_AM_IMA_fh“

Program je tvořen se známými daty, aby bylo možné určit kvalitu restaurace obrazu. Program začíná načtením uměle vytvořeného obrazu $f(m,n)$, viz **Obrázek 4.1 a** „Originální obraz“ (převzatý z [13]), nebo obrazu reálného objektu, viz **Obrázek 4.1 b** „Originální obraz“ (převzatý z [14]). Ty jsou dále zkrusleny různými PSF, viz **Obrázek 4.1 c**, reprezentující různé kanály, viz **Obrázek 4.1 a, b** „Zkruslené obrazy“. Zde je potřeba dodržet podmínku, že PSF zachovává intenzitu obrazu. Takto zkruslený obraz může být ještě zašuměn, dle další podmínky, Gaussovským bílým šumem. Odpovídající část zdrojového kódu:

```
% Načtení originalního obrazu
f=imread('Originalni_obraz.jpg');
f=double(rgb2gray(f));

% Jednotlivé PSF
h111=[0 1 0; 1 1 1; 0 1 0];
h111=h111/sum(sum(abs(h111)));
h222=[0 0 0; 1 1 1; 0 0 0];
h222=h222/sum(sum(abs(h222)));
h333=[0 1 0; 0 1 0; 0 1 0];
h333=h333/sum(sum(abs(h333)));
h444=[1 0 0; 0 1 0; 0 0 1];
h444=h444/sum(sum(abs(h444)));
h555=[0 0 1; 0 1 0; 1 0 0];
h555=h555/sum(sum(abs(h555)));
```

pozn.: Vydělením jednotlivých PSF absolutní hodnotou součtu masky je splněna podmínka zachování intenzity obrazu.



Obrázek 4.1: Originální obraz před a po zkreslení pěti různými PSF (uměle vytvořený obraz (a), obraz reálného objektu (b), zkreslující PSF (c))

```
% Zkreslení obrazu konvolucí původního obrazu s PSF
g1=conv2(f,h111,'same');
g2=conv2(f,h222,'same');
g3=conv2(f,h333,'same');
g4=conv2(f,h444,'same');
g5=conv2(f,h555,'same');

% Zašumění obrazu Gaussovským bílým šumem
g1=g1/256;
g1=imnoise(g1,'gaussian',0,0.00055); %0.0055
g1=g1*256;
g1=round(g1);

g2=g2/256;
g2=imnoise(g2,'gaussian',0,0.00055); %0.0055
g2=g2*256;
g2=round(g2);

g3=g3/256;
g3=imnoise(g3,'gaussian',0,0.00055); %0.0055
g3=g3*256;
g3=round(g3);

g4=g4/256;
g4=imnoise(g4,'gaussian',0,0.00055); %0.0055
g4=g4*256;
g4=round(g4);

g5=g5/256;
g5=imnoise(g5,'gaussian',0,0.00055); %0.0055
g5=g5*256;
g5=round(g5);
```

pozn.:

Ač metoda MBD-AM předpokládá, že šum má nulovou střední hodnotu i rozptyl (viz kapitola 4.1), v běžné praxi restauračních úloh nebývá tato podmínka splněna. Proto i v této práci se uvažuje nenulový rozptyl šumu.

$J = \text{imnoise}(I, 'gaussian', m, v)$, kde k obrazu I je přidáván Gaussův bílý šum se střední hodnotou $m = 0$ a rozptylem v .

Šum bývá ale většinou udáván v rámci poměru signál/šum (SNR), dle vzorce:

$$SNR_{dB} = 10 \log \left(\frac{S_{gg}}{S_{ww}} \right) = \left(\frac{\frac{1}{mn} \sum_{m,n} g^2(m,n)}{\frac{1}{mn} \sum_{m,n} w^2(m,n)} \right), \quad (4.25)$$

kde S_{ww} a S_{gg} jsou výkonová spektra šumu a signálu. Za předpokladu bílého šumu se S_{ww} rovná rozptylu šumu σ_n^2 . Při zvoleném SNR se pak odtud odvodí příslušný rozptyl. Například při $SNR = 30$ dB pro obraz kameramana (viz **Obrázek 4.1 b**) se $\sigma_n^2 = 0.00069$, a pro uměle vytvořený obraz (viz **Obrázek 4.1 b**) se $\sigma_n^2 = 0.00055$. [2], [9]

V téhle fázi programu se originální obraz „zapomene“ a k dispozici zůstávají jen zkeslené obrazy, jako jediná apriorní informace. Znáмым vstupem pro samotný algoritmus MBD-AM jsou tedy pouze zkeslené obrazy $g1-g5$. Obraz f a PSF $h1-h5$ se musí odhadnout. Protože o nich nejsou žádné apriorní informace, přiřadí se každému pixelu obrazu f a PSF $h1-h5$ stejná pravděpodobnost. Odhady se označí jako $f0$ a $h0$.

```
% pocatecni odhady f0, h0
f0=ones(size(g1,1),size(g1,2));
h0=ones(3,3);
h0=h0/sum(sum(abs(h0)));
h1=h0; h2=h0; h3=h0; h4=h0; h5=h0;
```

Nyní přichází na řadu samotné jádro programu, a to optimalizace funkce $E(f, h)$, viz (4.11). Nejprve je potřeba nadefinovat parametry optimalizace, čili maximální počet vnitřních iterací 'MaxIter' a volání funkce 'MaxFunEvals', konečnou toleranci funkční hodnoty 'TolFun' a proměnných 'TolX' a hlavně samotnou metodu optimalizace. Pro kvazi-Newtonovu metodu s aproximací hessiánu pomocí BFGS se nastavuje 'HessUpdate', 'bfgs', pro metodu největšího spádu pak 'HessUpdate', 'steepdesc', (více viz [13] nebo [14]). Jednotlivé parametry jsou nastaveny experimentálně.

```
options = optimset('LargeScale','off','MaxFunEvals',...
300000,'MaxIter',10,'Display','iter',...
'TolFun',0.00000001,'HessUpdate','bfgs',...
'LineSearchType','quadcubic','TolX',0.00000001);
```

Pro simplexovou metodu:

```
options = optimset('LargeScale','off',...
'MaxIter',10000,'TolFun',0.00001,...
'Simplex','on','TolX',0.00001);
```

Počet vnitřních iterací 'MaxIter' je zde řádově mnohem větší než u BFGS z důvodu povahy metody (simplexová metoda je velmi pomalá, viz kapitola 4.2.1).

Po nadefinování optimalizace následuje realizace dvou minimalizačních podúloh dle vzorce (4.12).

```
tic; % spuštění časovače
for i=1:2
    % minimalizace h
    x_h0 = [reshape(h1,1,[]), reshape(h2,1,[]), ...
            reshape(h3,1,[]), reshape(h4,1,[]), reshape(h5,1,[])];

    [xh,fval_h,exitflag_h,output_h] = fminunc('ima_h',x_h0,options);

    EH=xh;
    EH(EH<0)=0;
    EH(EH>255)=255;

    h1 = reshape(EH(1:9),3,[]);
    h2 = reshape(EH(10:18),3,[]);
    h3 = reshape(EH(19:27),3,[]);
    h4 = reshape(EH(28:36),3,[]);
    h5 = reshape(EH(37:45),3,[]);

    % minimalizace f
    x_f0 = [reshape(f0,1,[])];

    [xf,fval_f,exitflag_f,output_f] = fminunc('ima_f',x_f0,options);

    EF=xf;
    EF(EF<0)=0;
    EF(EF>255)=255;

    f0 = reshape(EF(1:2500),50,[]);

end
toc % zastavení časovače
```

pozn.:

Simplexová metoda je v Matlabu definovaná funkcí `fminsearch` a dává se v kódu na stejné místo jako funkce `fminunc`.

`[x,...] = fminunc (fun, x0, options)`, kde `fminunc` hledá globální minimum funkce `fun`, dle optimalizace definované členem `options` a začíná od vektoru `x0`, výsledek zapisuje do proměnné `x`. Funkce `fun`, resp. vzorec (4.11) je v tomto případě přepsán do dvou m-filů zvlášť pro f a h ('`ima_f`' a '`ima_h`'). Tyto minimalizační algoritmy se střídají ve for-cyklu („počet for-cyklů“ se bude dále označovat jako „počet vnějších iterací“ MBD-AM) Metoda konverguje ke správnému výsledku, neboť pracuje s neustále se aktualizujícími vzorky f a h .

V Matlabu je to realizováno pomocí funkcí `global`:

Pro '`ima_f`':

```
function E=ima_f(zyx)

global g1, global h1
global g2, global h2
global g3, global h3
global g4, global h4
global g5, global h5

f_obr = reshape(zyx(1:2500),50,[]);
```

Pro 'ima_h':

```
function E=ima_h(xyz)

global g1
global g2
global g3
global g4
global g5

global f0

% PSF
h1 = reshape(xyz(1:9),3,[]);
h2 = reshape(xyz(10:18),3,[]);
h3 = reshape(xyz(19:27),3,[]);
h4 = reshape(xyz(28:36),3,[]);
h5 = reshape(xyz(37:45),3,[]);

f_obr = f0;
```

Odtud je pro obě funkce program již stejný a prezentuje vzorec (4.11).

```
% ITERACNI MINIMALIZACNI ALGORITMUS %
% Konvoluce
Wo=(sum(sum((conv2(f_obr,h1,'same')-g1).^2))+...
+sum(sum((conv2(f_obr,h2,'same')-g2).^2))+...
+sum(sum((conv2(f_obr,h3,'same')-g3).^2))+...
+sum(sum((conv2(f_obr,h4,'same')-g4).^2))+...
+sum(sum((conv2(f_obr,h5,'same')-g5).^2)));
W=Wo.*1/2;

% Regularizace zkreslení R(h)
Ro=(sum(sum(((conv2(g1,h2,'same'))-(conv2(g2,h1,'same'))).^2))...
+sum(sum(((conv2(g2,h3,'same'))-(conv2(g3,h2,'same'))).^2))...
+sum(sum(((conv2(g3,h4,'same'))-(conv2(g4,h3,'same'))).^2))...
+sum(sum(((conv2(g4,h5,'same'))-(conv2(g5,h4,'same'))).^2))...
+sum(sum(((conv2(g5,h1,'same'))-(conv2(g1,h5,'same'))).^2)));
gama=1.3193;
R=Ro.*(gama/2);

% Regularizace TV obrazu Q(f)
L2=eye(size(f_obr,1),size(f_obr,2));
L1=[(L2(:,end)),(L2(:,(1:(end-1))))];
L1=L1.*(-1);
L=L1+L2;
L=[(L(1:end-1,:));0*(L(end,:))];
fx=sum(L'.*f_obr);
fy=sum(L.*f_obr);
beta=0.0001;
f_xyb=fx.^2+fy.^2+beta;
fi=sqrt(f_xyb);
Qo=sum(sum(fi));

lambda=0.00016;

Q=Qo.*lambda;

E = W+R+Q;
```

Hodnoty korekčních parametrů $\lambda = 0.00016$ a $\gamma = 1.3193$ jsou odhadnuty experimentálně, [10].

Regularizační člen obrazu $Q(f)$ používající gradientu Total Variation je ve tvaru: [11]

$$Q_{TV}(f) = \sum_{m,n} \Phi(\nabla f(m,n)) \approx \sum_{m,n} \sqrt{\sum (Lf)^2 + \sum (L^T f)^2} + \beta, \quad (4.26)$$

člen Tichonovské regularizace pak:

$$Q(f) \approx \sum_{m,n} \left(\sum (Lf)^2 + \sum (L^T f)^2 \right), \quad (4.27)$$

kde β je parametr blížký nule (0.0001) a L je matice 1. derivace ve tvaru:

$$L_1 = \begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & \cdot & \cdot & \cdot & \vdots \\ 0 & \dots & \dots & -1 & 1 \end{pmatrix} \quad (4.28)$$

o rozměrech $((n-d) \times n)$, kde n je rozměr obrazu a d značí d -tou derivaci.

Poslední částí programu je výpočet kvality metody MBD-AM pro restauraci obrazu danou procentuálním vyjádřením střední kvadratické chyby:

$$PMSE(f) = 100 \frac{\|fo(m,n) - f(m,n)\|}{\|f(m,n)\|}. \quad (4.29)$$

Tato střední kvadratická chyba však nutně nemusí korespondovat s vizuálním dojmem kvality obrazu.

```
% kvalita MBD-AM algoritmu
PMSE_f=100*sqrt(sum(sum(f0-f).^2))/sqrt(sum(sum(f.^2)))
PMSE_h1=100*sqrt(sum(sum(h1-h111).^2))/sqrt(sum(sum(h111.^2)))
PMSE_h2=100*sqrt(sum(sum(h2-h222).^2))/sqrt(sum(sum(h222.^2)))
PMSE_h3=100*sqrt(sum(sum(h3-h333).^2))/sqrt(sum(sum(h333.^2)))
PMSE_h4=100*sqrt(sum(sum(h4-h444).^2))/sqrt(sum(sum(h444.^2)))
PMSE_h5=100*sqrt(sum(sum(h5-h555).^2))/sqrt(sum(sum(h555.^2)))
```

4.4 REALIZACE MBD-AM POMOCÍ KNIHOVNY NAG C

Z hlediska časové a paměťové náročnosti programů, je jazyk C obecně vhodnější než Matlab. U algoritmu MBD-AM se to týká především simplexové metody, jež vyžaduje velký počet vnitřních iterací. Proto se v této kapitole přechází k programování hlavního jádra (optimalizační úlohy (4.11), resp. (4.12)) pomocí knihovny NAG v jazyce C.

4.4.1 Knihovna NAG C

Obsahem knihovny NAG je komplexní sbírka funkcí určených k řešení numerických a statistických problémů. Knihovna se dále dělí do jednotlivých kapitol, z nichž se každá věnuje konkrétnímu odvětví numerické analýzy nebo statistiky. Kapitoly se značí třemi znaky (např. c06 – Fourier Transforms), jejich dokumentované funkce pak šesti znaky (např. c06eac - nag_fft_real, nebo c06ecc - nag_fft_complex). Další kapitoly a funkce je možné nalézt na stránkách NAG [17], kde se kromě samotných funkcí nachází také velmi kvalitní *Help* včetně podrobného teoretického rozboru dané problematiky.

Za zmínku rovněž stojí, že NAG je uzpůsoben pro široký okruh programovacích prostředí a jazyků, jako například C#.NET, Excel, Java, Labview, Maple, Matlab, Visual Basic, a také proto má uplatnění v mnoha oborech, např. průmyslovém odvětví, ekonomii, ve vývoji, výzkumu na akademické půdě apod.

Tato diplomová práce využívá NAG v rámci programovacího jazyka C, zvláště pak kapitulu „e04 – Minimizing or Maximizing a Function“ a její funkci „e04ccc - nag_opt_simplex“. Tato funkce se zakládá na algoritmu simplexové metody. [17]

4.4.2 Popis programu „optimalizace_h_ccc“ a „optimalizace_f_ccc“

Programy „optimalizace_h_ccc“ a „optimalizace_f_ccc“ slouží k optimalizaci funkce $E(f, h)$, viz (4.11), resp. odpovídají dvěma minimalizačním podúlohám, viz (4.12). Na začátku každého programu, který volá nějakou funkci z knihovny NAG C, musí být hlavičkový soubor <nag.h> (definuje základní prostředí), případně také <nag_stdlib.h> (definuje alokaci paměti NAG_ALLOC a NAG_FREE.). Každá kapitola má rovněž svůj hlavičkový soubor. Proto program, který volá funkci „e04ccc - nag_opt_simplex“ z kapitoly e04, musí obsahovat hlavičkový soubor <nage04.h>. Tento program rovněž využívá knihovny <math.h>, jež obsahuje funkce vykonávající matematické operace a knihovnu pro standardní vstupy a výstupy <stdio.h> s funkcemi jako scanf(), printf() apod. Všechny hlavičkové soubory jsou volány příkazem #include [18].

Specifikace funkce **funct** $E(x)$ definované uživatelem:

```
void funct (Integer n, double *xc, double *objf, Nag_Comm *comm)
```

kde **n** je na vstupu celé číslo a reprezentuje počet proměnných vektoru **x**,

xc [n] – reálné číslo na vstupu se zvýšenou přesností, udává hodnotu prvkům vektoru **x**,

objf - reálné číslo na výstupu se zvýšenou přesností, udává hodnotu $E(x)$ v daném bodě x ,

Nag_Comm *comm – struktura obsahující ukazatel na komunikaci s uživatelem, pomocí níž definuje doplňující parametry. Pokud uživatel tuto komunikaci nepotřebuje, nastaví nulový ukazatel NAGCOMM_NULL.

NagError *fail – v případě, že se vyskytne chybný parametr, indikátor chyb ihned ukončí program.

Pro zlepšení přesnosti výpočtu lze použít volání ukazatele na strukturu **Nag_E04_Opt.**, jež obsahuje hodnoty optimalizačních parametrů. U programů „optimalizace_h_ccc“ a „optimalizace_f_ccc“ bylo využito příkazu **max_iter**, neboli maximálního počtu iterací, **optim_tool** – požadované přesnosti výpočtu ε (Pokud je ε dosaženo před proběhnutím maximálního počtu iterací, běh programu se ukončí a vypíše se výsledek.). Odpovídající část programu:

```
/* Nacteni struktury options */
nag_opt_init(&options);
options.max_iter = 10000;
options.optim_tool = 0.01;
```

Jádro programu optimalizace pomocí simplexového algoritmu:

```
nag_opt_simplex(n, funct, x, &objf, &options, &comm, &fail);
```

Jak již bylo výše řečeno, v této práci se jazyk C využije pouze k programování optimalizační úlohy. Vektor proměnných \mathbf{x} a hodnota \mathbf{n} se získají z Matlabu, kde se uloží do textového souboru jako „obraz_f.txt“, kritériální funkce **objf** pak pomocí tzv. MEX-souboru (viz kapitola 4.4.3).

```
% Nacteni f0 do souboru txt
obraz_f = [lf, f0];
save -ASCII obraz_f.txt obraz_f,
```

kde f_0 odpovídá vektoru počátečního odhadu obrazu $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ (viz kapitola 4.3.1) a lf značí délku vektoru, tedy počet proměnných n .

V programu C se pak soubor „obraz_f.txt“ otevře, načtou se z něj hodnoty a uloží do patřičných proměnných.

```
/* Nacteni pole pocatecniho odhadu */

FILE *param;
param = fopen("obraz_f.txt", "r");
fscanf(param, "%f", &ff);
n = ff;

for(index=0; index<n; index++)
{
    fscanf(param, "%f", &ff);
    x[index] = ff;
}
```

Po ukončení optimalizačního algoritmu se výsledné hodnoty zapíše a uloží do textového souboru „vysledek_f.txt“ v programu C.

```
/*Ulozeni vysledku*/

param = fopen("vysledek_f.txt", "w");
for(index=0; index<n; index++)
{
    ff = x[index];
    fprintf(param, "%f\n", ff);
}
fclose(param);
```

a načtou se v Matlabu.

```
% Nacteni f0 do souboru txt
vysledek_f = load('-ASCII', 'vysledek_f.txt');
```

4.4.3 MEX soubory

Pomocí MEX-souborů (MEX-filů) je možné volat proměnné nebo celé funkce z Matlabu do kódu tvořeného v jazyce C. Jestliže se tento kód zkompile, může být použit jako funkce v Matlabu a volá se stejným způsobem jako klasické m-fily. Jelikož se jedná o zkompileované soubory, jsou i rychlejší, než m-fily.

Každý C MEX-file musí začínat hlavičkovým souborem:

```
#include "mex.h",
```

dále se na místo funkce `main()` píše:

```
void mexFunction(int nlhs, mxArray *plhs[],...
                 int nrhs, const mxArray *prhs[]).
```

`mexFunction` má čtyři parametry (např. `[y1, y2]=myFunction(x1, x2, x3)`).

`nlhs` – celočíselný parametr – reprezentuje počet výstupních proměnných `y1, y2` (`nlhs = 2`)

`plhs` – pole ukazatele na výstupní matice – nemá nastaveny počáteční hodnoty.

Jestliže `plhs > 0`, pak MEX-soubor bude plnit `plhs` pole ukazatele na matici platnými hodnotami, v opačném případě podá chybové hlášení.

`nrhs` – celočíselný parametr – reprezentuje počet vstupních proměnných `x1, x2, x3` (`nrhs = 3`)

`prhs` – pole ukazatele na vstupní matice – např. `prhs[0]` je ukazatel na `x1` a `prhs[1]` je ukazatel na `x2`.

Hlavní strukturou pro zápis dat v Matlabu je matice (příkaz v MEX-souboru `mxArray`), neboť pomocí ní lze zapsat reálné hodnoty, komplexní čísla, pole proměnných, vektory, celé číslo a další datové struktury.

Funkce `mxGetPr` se používá jako ukazatel na reálná data, `mxGetScalar` jako ukazatel na celé číslo. `mexCallMATLAB` volá funkci z Matlabu a udává počet vstupních a výstupních proměnných [14], [18], [19].

Volání kriteriální (objektové) funkce **objf**, resp. `'ima_f'` vytvořené v Matlabu v kapitole 4.3.1 do jazyka C:

```
/* Volani funkce ima_f z Matlabu do *objf v C */
{
    mxArray *result;
    mxArray *arguments[1];
    int i;

    arguments[0] = mxCreateDoubleMatrix(1, n, mxREAL);
    double *outArray = mxGetPr(arguments[0]);

    for (i = 0; i < n; i++)
        outArray[i] = xc[i];

    mexCallMATLAB(1, &result, 1, arguments, "ima_f");

    *objf = (int)(mxGetScalar(result));
}
```

4.4.4 Kompilace MEX souborů

Kompilace obecně slouží k překladu kódu ze vstupního jazyka do jazyka výstupního. Kompilátor „`kompiluj.m`“ pro C MEX soubory je program, který provádí překlad kódu `'optimalizace_f_ccc.c'` z jazyka C, na funkci „`optimalizace_f_ccc.mexa64`“ pro Matlab.

Podoba funkce, která kompiluje MEX-soubor:

```
function kompiluj(zdrojak)

eval(['mex -I/opt/NAG/c116a08dgl/include
      /opt/NAG/c116a08dgl/lib/libnagc_nag.a
      -lpthread -lm ' zdrojak]);
```

kde `mex` je příkaz pro kompilaci souboru „zdrojak“, `-I/opt/NAG/c116a08dgl/include/` odkazuje na soubory, v nichž se mají hledat hlavičkové soubory, `opt/NAG/c116a08dgl/lib/libnagc_nag.a` ukazuje cestu ke zkompilevaným knihovnám NAG, `-lpthread` volá zkompilevanou matematickou knihovnu umožňující paralelní výpočty a `-lm` knihovnu matematických funkcí.

Do Command Window v Matlabu pak už jen stačí zadat:

```
kompiluj('optimalizace_f_ccc.c')
```

a výsledkem je soubor „optimalizace_f_ccc.mexa64“, jež nyní odpovídá optimalizačnímu algoritmu simplexové metody pro obraz f a lze jej použít jako m-file.

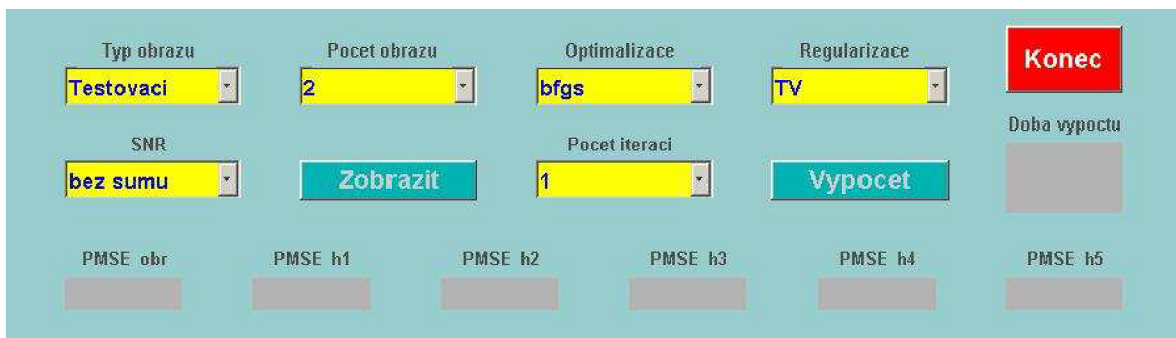
```
% optimalizacni funkce pro f (mex-soubor)
optimalizace_f_ccc;
```

Pro optimalizaci PSF se postupuje obdobně jako v předchozích kapitolách u optimalizace obrazu f , s výstupem „optimalizace_h_ccc.mexa64“. Další běh programu je stejný jako v kapitole (4.3.1). [17]

4.5 REALIZACE MBD-AM V GRAFICKÉM ROZHŘANÍ

Programové prostředí Matlab má široké možnosti použití. Může se pomocí něj počítat přímo v Command Window, nebo programovat v m-souborech, popřípadě tvořit uživatelské rozhraní. Grafické (uživatelské) rozhraní je velmi vhodné zvláště v případech programů, u nichž si uživatel přeje snadno nastavovat a měnit různé parametry nebo proměnné, tzn., aniž by pokaždé musel zasahovat do jednotlivých m-souborů. Z kapitol 4.3 a 4.4 vyplývá pro metodu MBD_AM možnost volby typu obrazů, počtu kanálů, velikosti SNR, typu optimalizace a regularizace, nebo počtu vnějších iterací střídavého minimalizačního algoritmu IMA, což vedlo k vytvoření programu na realizaci MBD_AM v grafickém rozhraní.

Spuštěním m-souboru MBD_AM_IMA_fh.m se otevře grafické okno, viz **Obrázek 4.2**.



Obrázek 4.2: Grafické rozhraní programu na výpočet MBD_AM

V něm si uživatel nejprve vybere typ obrazu, s nímž chce pracovat:

Typ obrazu - ('Testovací| Kameraman| Lena| Text'),

dále zvolí počet zkraslených obrazů, resp. počet kanálů:

Počet obrazů - ('2| 3| 4| 5'),

velikost poměru signál/šum:

SNR - ('bez šumu| 40 dB| 30 dB| 20 dB| 10 dB '),

Po zmáčknutí tlačítka „Zobrazit“, se otevřou dvě okna. V případě volby např. Typ obrazu: Kameraman, Počet obrazů: 5 a SNR: bez šumu, se v prvním okně, s názvem „Originální a zkraslené obrazy“, objeví obrazy f a g_1-g_5 , viz **Obrázek 4.1 b**, a ve druhém okně s názvem „Originální PSF“ zkraslující impulzní charakteristiky h_1-h_5 , viz **Obrázek 4.1 c**.

Nyní má uživatel k dispozici vstupní data (daný počet zkraslených obrazů) a může volit parametry pro následnou slepou dekonvoluci:

typ optimalizace:

Optimalizace - ('bfgs| steepdesc| simplex_m| simplex_nag'),

typ regularizace:

Regularizace - ('TV| Tichonov '),

a počet vnějších iterací IMA:

Počet iterací - ('1| 2| 3| 5| 10').

Po spuštění střídavého minimalizačního algoritmu tlačítkem „Výpočet“ se, po dokončení výpočtu MBD-AM, který trvá s ohledem na volbu parametrů několik minut, otevřou další dvě okna. Pro volbu např. Optimalizace: bfgs, Regularizace TV a Počet iterací 3, se v okně, s názvem „Originální a restaurovaný obraz“ objeví obrazy f a xf , viz **Obrázek 4.5 g** a ve druhém okně „Restaurované PSF“, příslušné PSF₁-PSF₅ viz **Obrázek 4.5 i**. Následně se do šedých polí, vypíší střední kvadratické odchylky PMSE obrazu a impulzních charakteristik v procentech a doba výpočtu v sekundách. [21]

pozn.: Simplexová optimalizační metoda vytvořená pomocí MEX souboru lze spustit v operačním systému Linux, resp. Unix.

4.6 HODNOCENÍ KVALITY RESTAURAČNÍ METODY MBD-AM

Kvalita metody MBD-AM je testována na uměle vytvořeném obrazu „Testovací“, (viz **Obrázek 4.1 a** „Originální obraz“) a na obrazu skutečného objektu „Kameraman“ (viz **Obrázek 4.1 b** „Originální obraz“). Oba mají rozměry 50 x 50 pixelů. Vychází se ze znalosti pěti různě zkraslených obrazů g_1-g_5 , (viz **Obrázek 4.1 a, b** „Zkraslené obrazy“).

Testování probíhá na počítači s parametry 16GB RAM, 2x Intel Xeon 3000. Tato informace se udává hlavně kvůli době výpočtu.

4.6.1 Hodnocení MBD-AM z pohledu počtu vnějších iterací

Za fixní parametry se zvolil typ obrazu „Testovací“, 5 vstupních zkraslených obrazů, bez šumu, regularizace total variation a kvazi-Newtonova metoda BFGS s počtem vnitřních iterací 10. Jediný proměnný parametr zůstává počet vnějších iterací IMA.

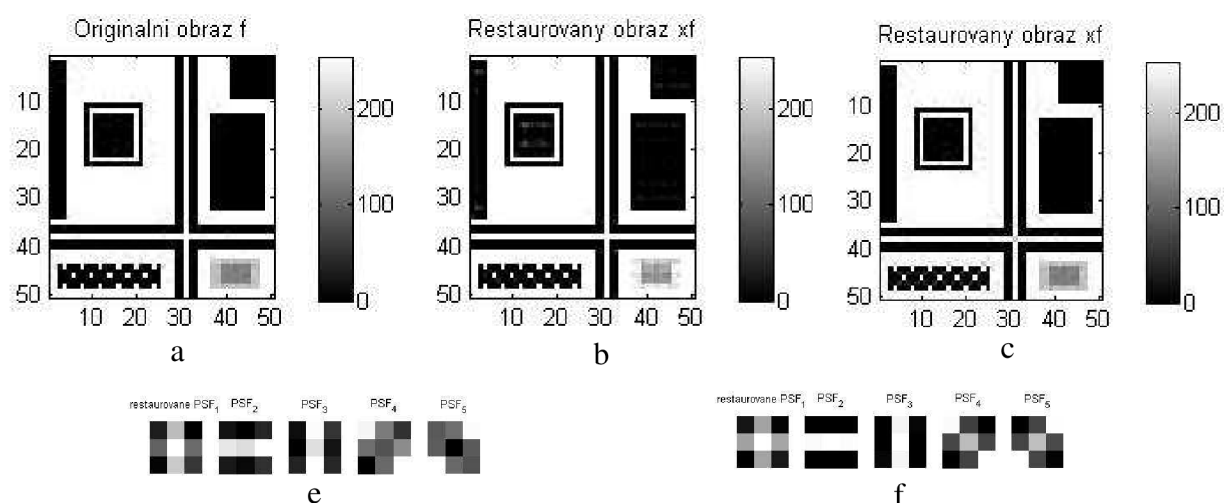
Při těchto hodnotách se dosahují dobré výsledky restaurace za relativně krátkou dobu výpočtu. V tabulce (viz **Tabulka 4.1**) jsou uvedeny hodnoty středních kvadratických chyb f a h_1-h_5 a také doba výpočtu pro různý počet střídání minimalizačních podúloh - vzorců (4.12). Jedna vnější iterace trvá přibližně 100 sekund.

Tabulka 4.1: Hodnoty PMSE a doby výpočtu „Testovacího“ obrazu pro i iterací

Počet iterací	PMSE_f [%]	PMSE_h1 [%]	PMSE_h2 [%]	PMSE_h3 [%]	PMSE_h4 [%]	PMSE_h5 [%]	Čas [s]
1	12.07	41.31	21.38	64.48	19.07	20.94	106
2	3.45	1.51	1.93	2.24	1.93	1.72	210
3	4.17	1.52	3.06	2.35	2.06	1.27	316
5	5.43	1.62	2.48	2.56	1.07	1.06	492

Z tabulky (viz **Tabulka 4.1**) a jí příslušného grafu (viz **Obrázek 4.4 a**) je patrné, že výraznějšího zlepšení se dosáhlo po skončení druhé iteraci, a dále již kvalita restaurace nijak významně nestoupá - pouze doba výpočtu, což je ale nežádoucí. Tohoto poznatku se tedy využije při dalším počítání.

Vizuálně lze kvalitu restaurační metody MBD-AM z pohledu počtu vnějších iterací určit z obrázku (viz **Obrázek 4.3**).



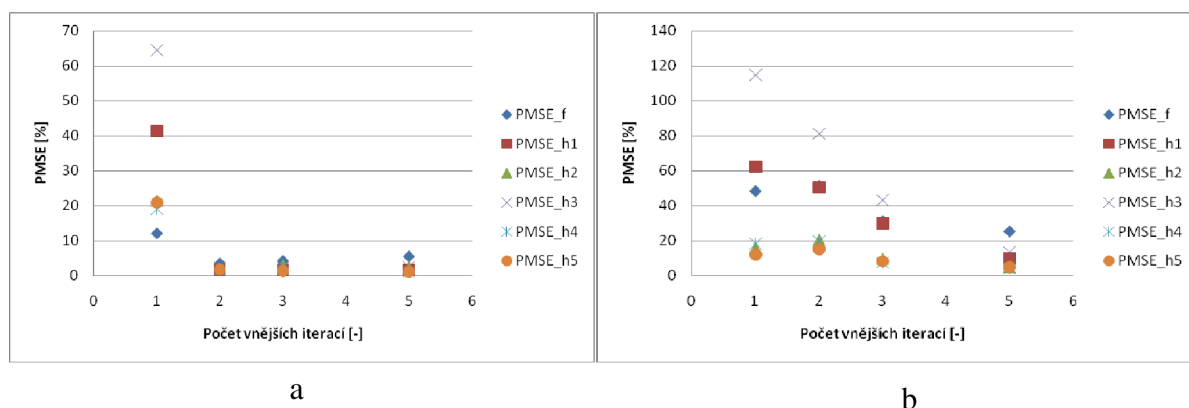
Obrázek 4.3: Porovnání kvality MBD-AM pro různý počet vnějších iterací a „Testovací“ obraz, (původní obraz (a), restaurovaný obraz (b) a PSF (e) po jedné iteraci, restaurovaný obraz (c) a PSF (f) po pěti iteracích)

Pro obraz typu „Kameraman“ a při zachování ostatních parametrů se dosáhne následujících výsledků, (viz **Tabulka 4.2**).

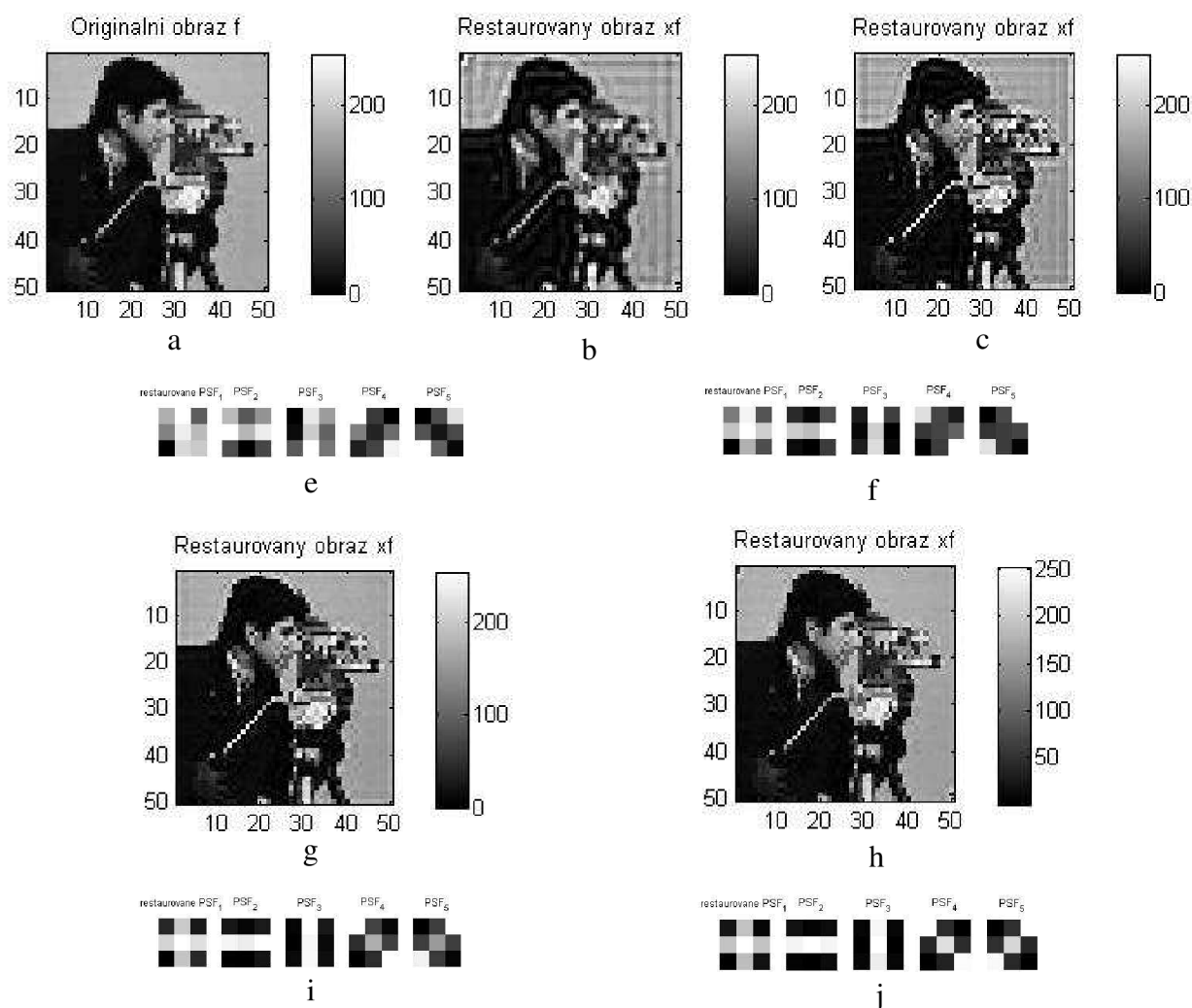
Tabulka 4.2: Hodnoty PMSE a doby výpočtu obrazu „Kameraman“ pro i iterací

Počet iterací	PMSE_f [%]	PMSE_h1 [%]	PMSE_h2 [%]	PMSE_h3 [%]	PMSE_h4 [%]	PMSE_h5 [%]	Čas [s]
1	48.26	62.39	16.13	114.97	18.47	12.22	116
2	51.15	50.63	20.82	81.28	19.69	15.33	209
3	31.15	29.90	9.83	43.12	7.58	8.35	318
5	25.31	9.84	5.03	13.11	4.62	4.98	500

Protože se opět jedná o obraz 50x50 pixelů, doba výpočtu je přibližně stejná jako u „Testovacího“ obrazu. Střední kvadratické chyby však dosahují mnohem větších hodnot, (viz **Tabulka 4.2** a **Obrázek 4.4 b**), což je dáno typem obrazu. Ten se již mnohem více blíží reálným datům, a také se u něj hůře regulují změny intenzity obrazu při dekonvolučním algoritmu. Pro další výpočty se budou používat tři iterace, kde došlo ke zlepšení o 20%. Vizuálně lze kvalitu restaurace posoudit z obrázku (viz **Obrázek 4.5**).



Obrázek 4.4: Graf závislosti střední kvadratické chyby na počtu vnějších iterací (pro „Testovací“ obraz (a) a obraz „Kameraman“ (b))



Obrázek 4.5: Porovnání kvality MBD-AM pro různý počet vnějších iterací a obraz „Kameraman“, (původní obraz (a), restaurovaný obraz (b) a PSF (e) po jedné iteraci, restaurovaný obraz (c) a PSF (f) po dvou iteracích, restaurovaný obraz (g) a PSF (i) po třech iteracích, restaurovaný obraz (h) a PSF (j) po pěti iteracích)

4.6.2 Hodnocení MBD-AM z pohledu velikosti poměru signál/šum

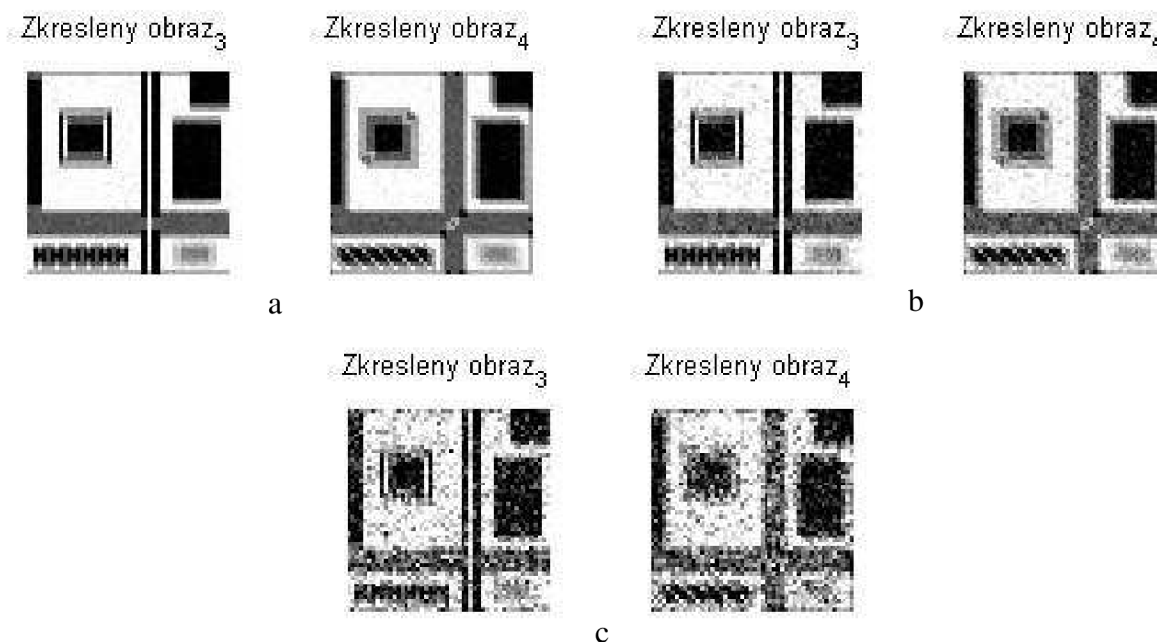
Vliv šumu na kvalitu restaurace, pro „Testovací“ obraz, 2 vnější iterace (zvoleno na základě výsledků z předchozí kapitoly), 5 zkreslených obrazů, BFGS optimalizaci a regularizaci TV, je uveden v tabulce (viz **Tabulka 4.3**) a jí odpovídajícím grafu (viz **Obrázek 4.8 a**).

Nutno podotknout, že v kapitole 4.6.1 byla metoda testována pro obraz zkreslený pouze impulzními charakteristikami, nikoli však šumem. Šum se zde uvažuje Gaussův bílý se střední hodnotou 0 a různými hodnotami rozptylu, s ohledem na požadovaný poměr signál/šum.

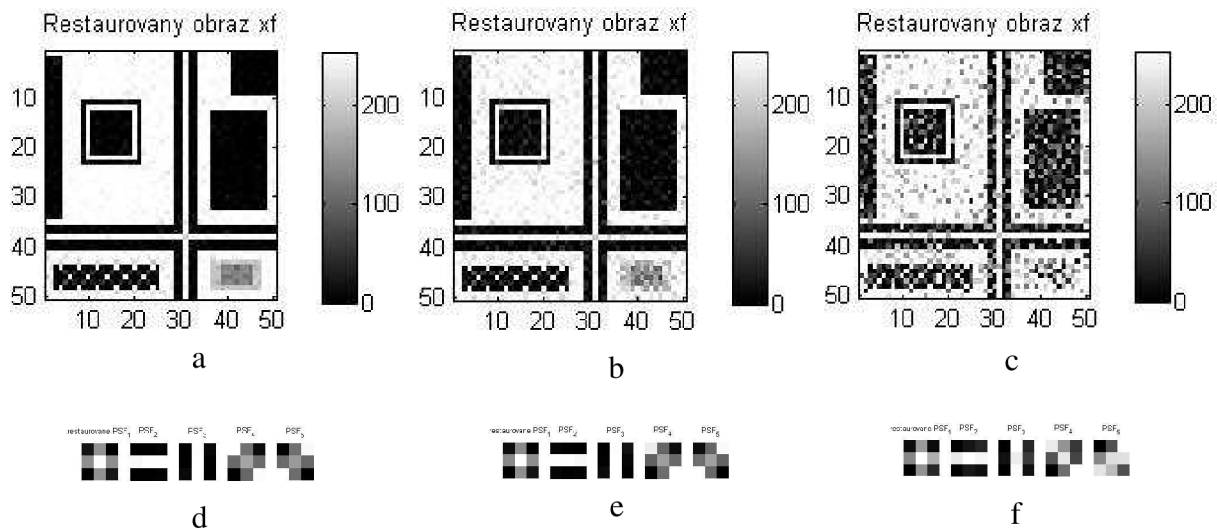
Tabulka 4.3: Hodnoty PMSE pro „Testovací“ obraz s ohledem na šum

SNR [dB]	PMSE_f [%]	PMSE_h1 [%]	PMSE_h2 [%]	PMSE_h3 [%]	PMSE_h4 [%]	PMSE_h5 [%]
10	85.11	36.20	21.08	67.75	20.89	21.15
20	19.81	4.63	3.58	13.48	4.04	3.48
30	6.62	1.77	2.34	3.97	2.43	2.31
40	4.12	1.63	2.04	2.23	1.86	1.93

Pro hodnoty SNR 30 dB a více, (viz **Obrázek 4.6 a**), je restaurace obrazu velmi kvalitní (viz **Obrázek 4.7 a**), při vizuální kontrole kvality restaurace ani není znát, že byl obraz kromě zkreslení PSF i zašuměn. Pro hodnoty SNR 20 dB a méně, je již střední kvadratická chyba 20%, pro 10 dB až 80% což je samozřejmě patrné i při vizuální kontrole kvality (viz zašuměný **Obrázek 4.6 b, c** a příslušný restaurovaný **Obrázek 4.7 a, b**). Nicméně i v tomto případě má restaurace obrazu smysl, neboť stále dochází ke zkvalitnění poškozených obrazů. Některé restaurační metody totiž při silném poškození obrazu šumem o velkém rozptylu takový obraz ještě více znehodnotí (viz [13]). V dalším testování programu na výpočet MBD-AM se bude používat poměr signál/šum 30 dB.



Obrázek 4.6: Ukázka vstupních „Testovacích“ obrazů zkreslených PSF a poškozených šumem, (SNR 30 dB (a), SNR 20dB (b), SNR 10 dB (c))

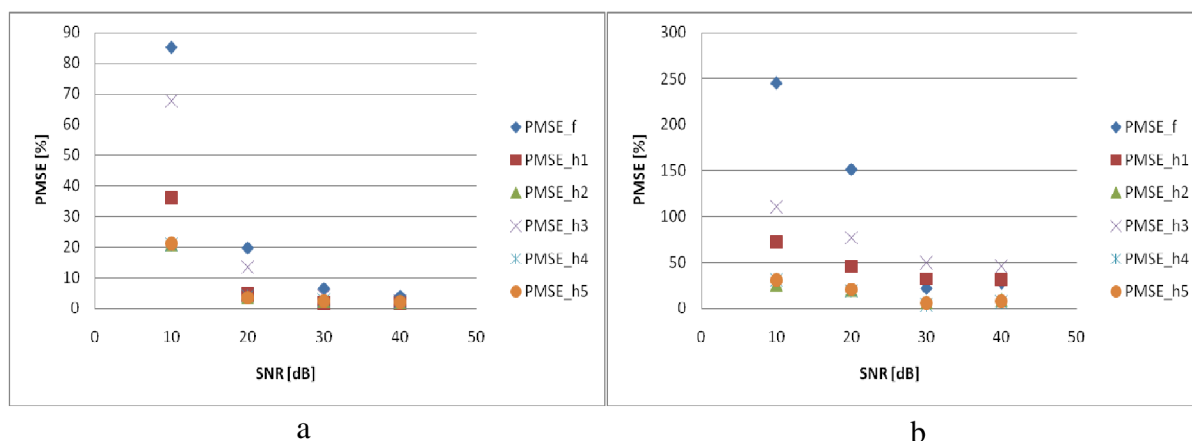


Obrázek 4.7: Porovnání kvality MBD-AM z pohledu velikosti SNR pro „Testovací“ obraz, (restaurovaný obraz (a) a PSF (d) pro SNR 30 dB, restaurovaný obraz (b) a PSF (e) pro SNR 20 dB, restaurovaný obraz (c) a PSF (f) pro SNR 10 dB)

Výsledky testování vlivu šumu pro restauraci obrazu „Kameraman“ a třemi vnějšími iteracemi IMA (ostatní parametry zůstaly nezměněny), jsou zaznamenány v tabulce (viz **Tabulka 4.4**) a grafu (viz **Obrázek 4.8 b**).

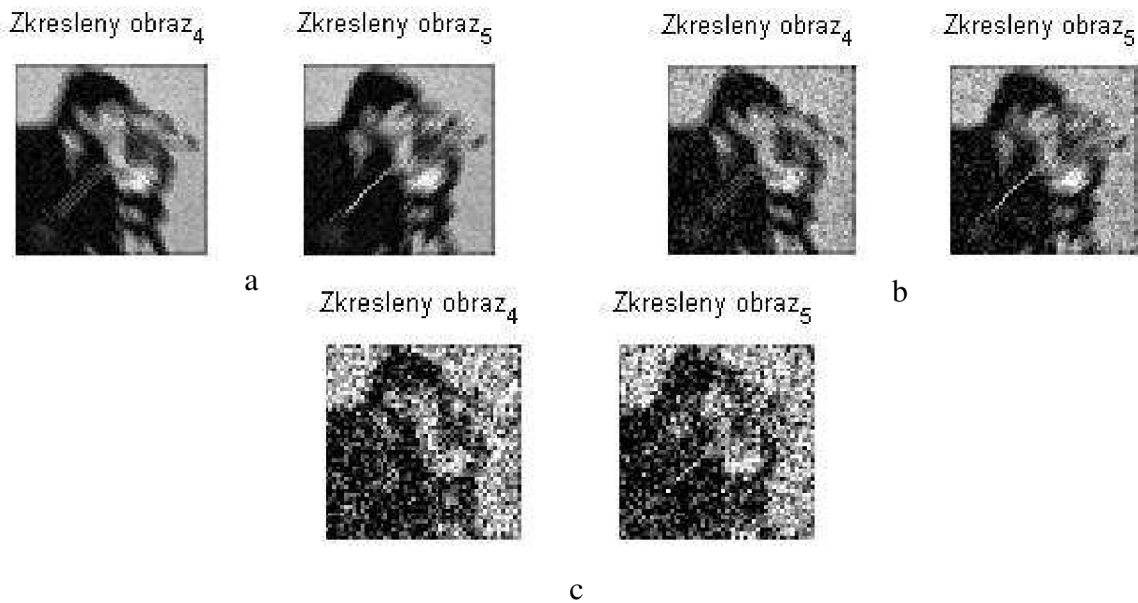
Tabulka 4.4: Hodnoty PMSE pro obraz „Kameraman“ s ohledem na šum

SNR [dB]	PMSE_f [%]	PMSE_h1 [%]	PMSE_h2 [%]	PMSE_h3 [%]	PMSE_h4 [%]	PMSE_h5 [%]
10	244.69	72.38	26.98	111.25	31.09	31.65
20	151.55	45.96	20.37	77.67	19.85	21.32
30	22.39	32.39	6.85	50.60	3.88	6.35
40	28.08	31.48	9.00	46.59	7.18	8.28

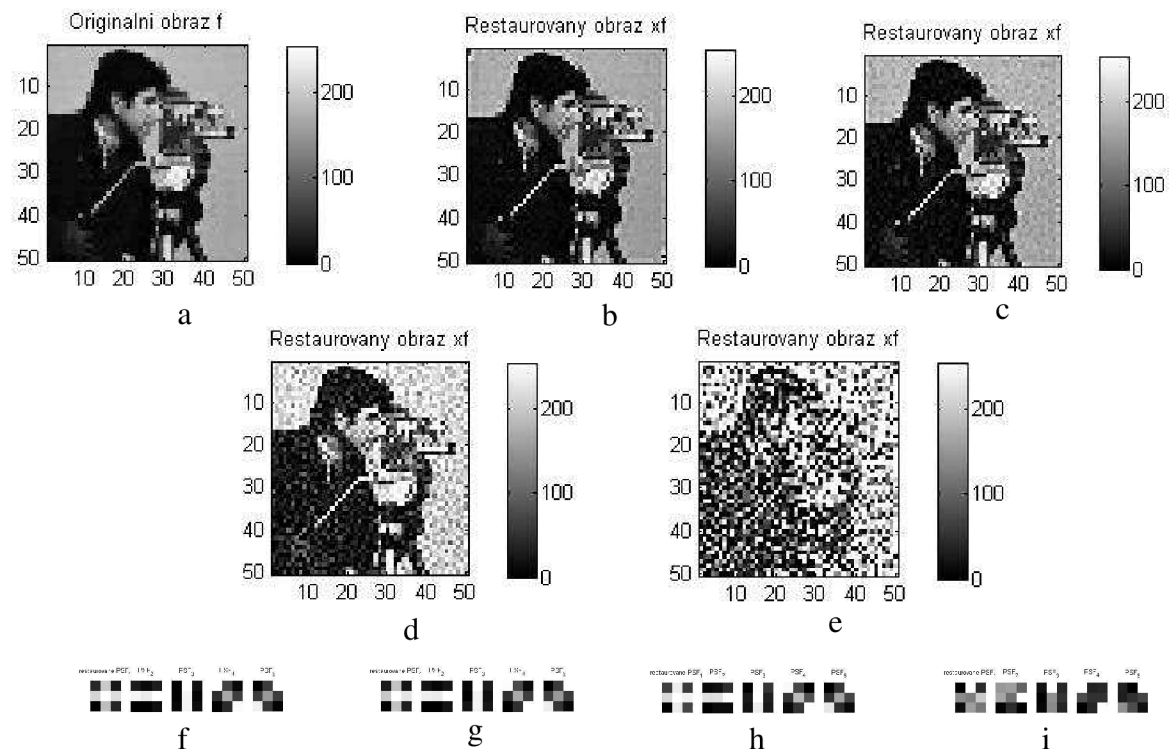


Obrázek 4.8: Graf závislosti střední kvadratické chyby na velikosti SNR (pro „Testovací“ obraz (a) a obraz „Kameraman“ (b))

I zde je na první pohled zřejmé, že při SNR nad 30 dB (viz **Obrázek 4.9 a**) se dosahuje kvalitní restaurace (viz **Obrázek 4.10 b, c**), při hodnotách pod 20 dB (viz **Obrázek 4.9 a, b**) je šum na restaurovaných obrazech velmi znatelný (viz **Obrázek 4.10 d, e**). Proto se pro další výpočty bude brát SNR 30 dB.



Obrázek 4.9: Ukázka vstupních obrazů „Kameraman“ zkreslených PSF a poškozených šumem, (SNR 30 dB (a), SNR 20dB (b), SNR 10 dB (c))



Obrázek 4.10: Porovnání kvality MBD-AM pro různé velikosti SNR a obraz „Kameraman“, (originální obraz (a), restaurovaný obraz (b) a PSF (f) pro SNR 40 dB, restaurovaný obraz (c) a PSF (g) pro SNR 30 dB, restaurovaný obraz (d) a PSF (h) pro SNR 20 dB, restaurovaný obraz (e) a PSF (i) pro SNR 10 dB)

4.6.3 Hodnocení MBD-AM z pohledu počtu kanálů

Dalším hodnocením kvality restaurační metody MBD-AM je různý počet vstupních zkraslených obrazů, resp. počet kanálů. Protože se jedná o metodu multikanálovou, nefunguje tato pro jeden kanál, což bylo experimentálně zjištěno na „Testovacím“ obrazu. Výsledný obraz nezachoval původní intenzitu obrazu a střední kvadratické chyby pro f a PSF se vyšplhaly k hodnotám 300%.

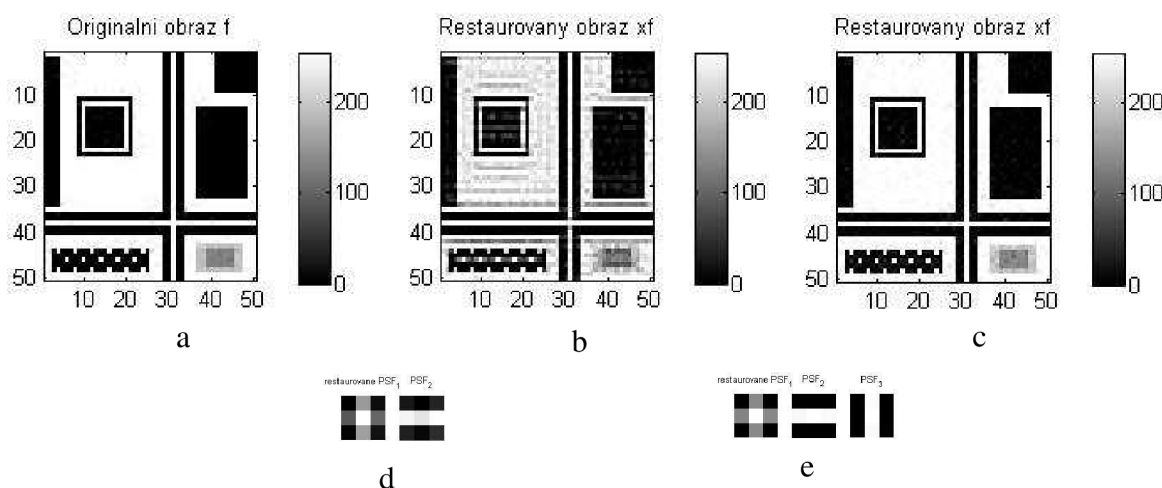
„Testovací“ obraz byl zašuměn bílým šumem, SNR 30 dB a ostatní parametry MBD-AM zůstaly stejné, jako v předchozí kapitole, samozřejmě se mění počet vstupních obrazů. Hodnoty středních kvadratických chyb obrazu f a PSF pro různý počet kanálů jsou uvedeny v tabulce (viz **Tabulka 3.3**) a grafu (viz **Obrázek 4.12 a**).

Tabulka 4.5: Hodnoty PMSE s ohledem na počet kanálů pro „Testovací“ obraz

Počet kanálů	PMSE_f [%]	PMSE_h1 [%]	PMSE_h2 [%]	PMSE_h3 [%]	PMSE_h4 [%]	PMSE_h5 [%]	Čas [s]
2	48.32	17.13	12.42	-	-	-	152
3	7.01	2.70	2.15	5.42	-	-	191
4	6.17	2.86	1.68	5.94	1.73	-	249
5	6.62	1.77	2.34	3.97	2.43	2.31	316

Z tabulky (viz **Tabulka 4.5**) a grafu (viz **Obrázek**) je patrné, že nejvýraznějšího zlepšení kvality restaurace „Testovaného“ obrazu došlo při použití tří kanálů a odtud se hodnoty PMSE ustálily.

Tuto skutečnost lze pozorovat a kvalitu restaurační metody i vizuálně posoudit z obrázku (viz **Obrázek 4.11 b** pro dva kanály, **Obrázek 4.11 c** pro tři kanály a odpovídající PSF **Obrázek 4.11 d** pro dva kanály, **Obrázek 4.11 e** pro tři kanály dekonvoluce).



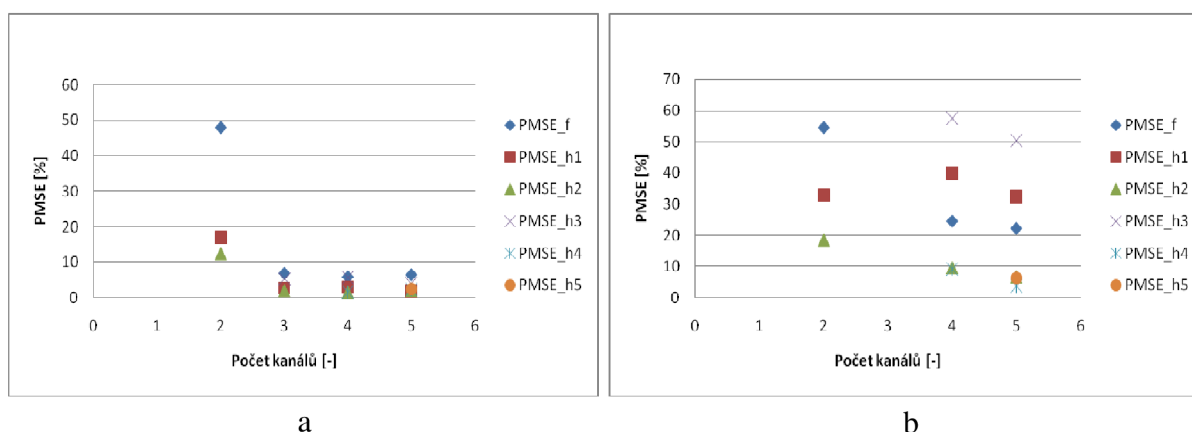
Obrázek 4.11: Porovnání kvality MBD-AM pro různý počet kanálů a „Testovací“ obraz, (originální obraz (a), restaurovaný obraz (b) a PSF (d) pro dva kanály, restaurovaný obraz (c) a PSF (e) pro tři kanály)

Pro hodnocení kvality restaurace obrazu „Kameraman“ z pohledu počtu vstupních zkraslených obrazů byl rovněž použit poměr signál/šum 30 dB, 3 vnější iterace a ostatní parametry se nemění. Výsledky jsou zaznamenány v tabulce, (viz **Tabulka 4.6**), a jí příslušném grafu, (viz **Obrázek 4.12 b**). Pro lepší názornost grafu jsou v něm záměrně vynechány hodnoty PMSE pro tři kanály, neboť jsou příliš velké. To je dáno nedodržením intenzity obrazu.

Vizuální kvalita tak velké chybě totiž rozhodně neodpovídá, (viz **Obrázek 4.13 c**). Také proto se u dalších výpočtů pro obraz „Kameraman“ spíše použijí čtyři kanály (viz **Obrázek 4.13 d**).

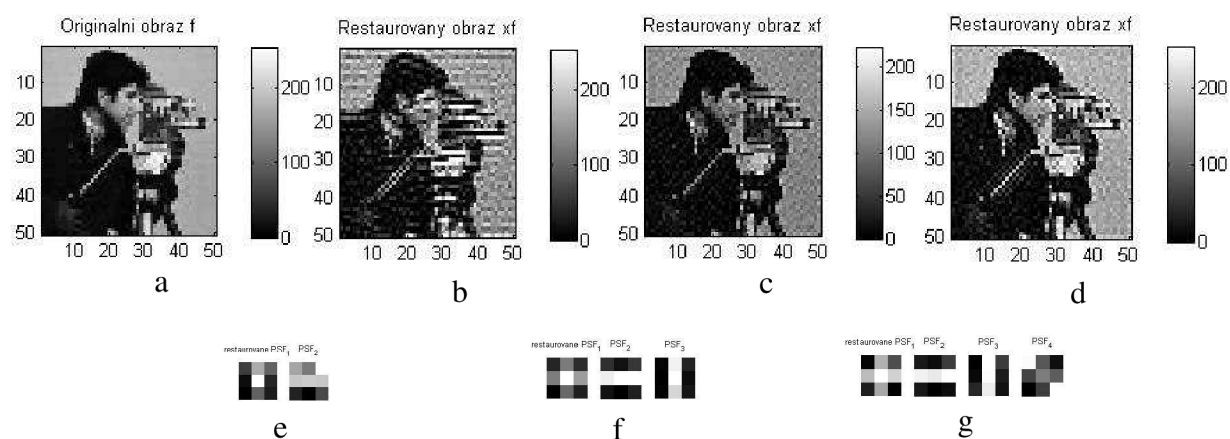
Tabulka 4.6: Hodnoty PMSE s ohledem na počet kanálů pro obraz „Kameraman“

Počet kanálů	PMSE_f [%]	PMSE_h1 [%]	PMSE_h2 [%]	PMSE_h3 [%]	PMSE_h4 [%]	PMSE_h5 [%]	Čas [s]
2	54.46	32.81	18.56	-	-	-	179
3	160.09	55.70	40.76	45.86	-	-	217
4	24.68	40.00	9.78	57.54	9.21	-	263
5	22.39	32.39	6.85	50.60	3.88	6.35	318



Obrázek 4.12: Graf závislosti středních kvadratických chyb na počtu kanálů (pro „Testovací“ obraz (a) a obraz „Kameraman“ (b))

Vizuální kvalitu pro různý počet vstupních obrazů lze hodnotit z obrázku, viz **Obrázek 4.13**.



Obrázek 4.13: Porovnání kvality MBD-AM pro různý počet kanálů a obraz „Kameraman“, (originální obraz (a), restaurovaný obraz (b) a PSF (e) pro dva kanály, restaurovaný obraz (c) a PSF (f) pro tři kanály, restaurovaný obraz (d) a PSF (g) pro čtyři kanály)

Z tabulek, (viz **Tabulka 4.5, 4.6**) lze také odpozorovat, že s klesajícím počtem kanálů, se zkracuje doba výpočtu.

4.6.4 Hodnocení MBD-AM z pohledu typu optimalizace

Simplexová metoda

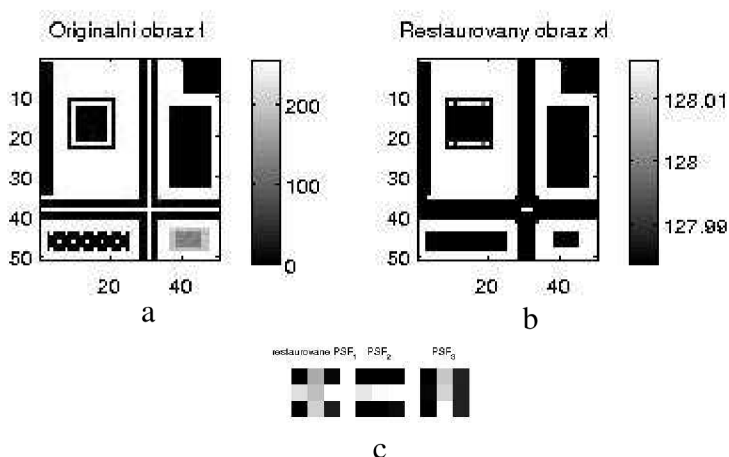
Pro simplexovou metodu se zvolí „Testovací“ obraz, počet kanálů 3, bez šumu, TV regularizace, 2 vnější iterace IMA a 10000 vnitřních iterací simplexové metody. Poté se použijí postupně obě metody, naprogramované jak v Matlabu, tak i pomocí knihovny NAG C.

Metody dosahují přibližně stejných hodnot PMSE, (viz **Tabulka 4.7**), avšak doba výpočtu simplexové metody v Matlabu je 10 krát delší, než pomocí knihovny NAG C.

Tabulka 4.7: Hodnoty PMSE optimalizační simplexové metody pro „Testovací“ obraz

Typ optimalizace	PMSE_f [%]	PMSE_h1 [%]	PMSE_h2 [%]	PMSE_h3 [%]	Čas [s]
Simplex_NAG	272.69	32.86	33.51	29.23	334
Simplex_Matlab	272.61	32.71	23.22	29.54	3202

Vizuálně lze kvalitu optimalizace pomocí NAG C posoudit z obrázku, viz **Obrázek 4.14**. Odtud je možné vypořadovat, že restaurované PSF se blíží originálnímu, zatímco u obrazu jsou vynechány detaily a není zachována intenzita obrazu.



Obrázek 4.14: Výsledný „Testovací“ obraz simplexové metody, (originální obraz (a), restaurovaný obraz z NAG C(b) a PSF (c) pro tři kanály

Protože simplexová metoda konverguje velmi pomalu a výsledek často závisí na počátečním odhadu, byla tentokrát vstupní jednotková matice $f\theta$ o rozměru (m,n) vynásobená 128, což je přibližně střední hodnota intenzity pixelů v obraze. Ani tímto však nedošlo ke zlepšení, pouze se hodnoty intenzity u restaurovaného obrazu posunuly do intervalu $\langle 127.99; 128.01 \rangle$, viz **Obrázek 4.14 b**, tedy o 128 bodů nahoru. Příčinou tohoto jevu může být také nevhodné zvolení hodnot korekčních parametrů λ a γ , viz kapitola 4.6.5.

pozn.: Experimentálně bylo zjištěno, že simplexová metoda je spíše použitelná pro neslepou dekonvoluci, zvláště pak při znalosti originálního obrazu f s cílem vypočítat impulsní charakteristiky h_i . To potvrzuje, že metoda je vhodná spíše k optimalizaci funkcí o menším počtu proměnných, viz kapitola 4.2.1.

Na základě předchozích poznatků se simplexová metoda k výpočtu MBD-AM nebude dále používat.

Metoda největšího spádu a BFGS

V následujících tabulkách, viz **Tabulka 4.8, 4.9**, jsou zaznamenány střední kvadratické chyby pro výpočet MBD-AM s pevně danými parametry a optimalizační metodu největšího spádu nebo kvazi-Newtonovu BFGS.

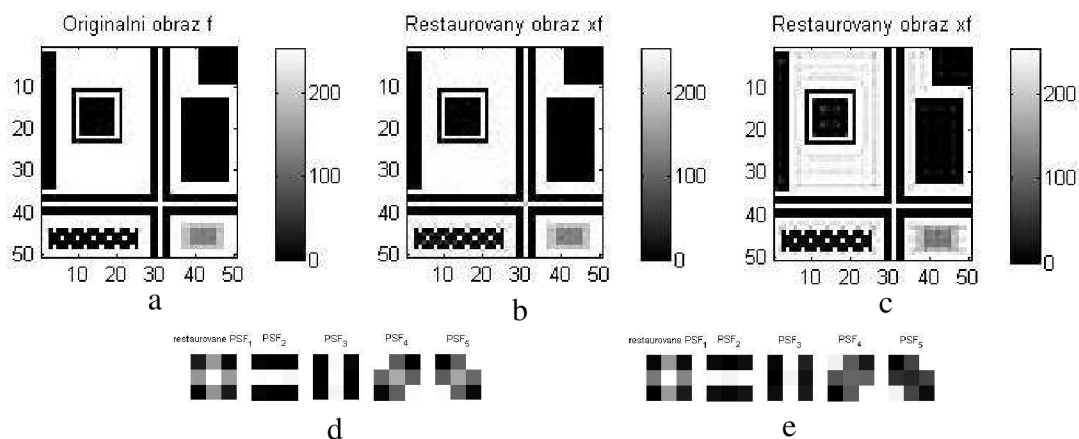
Tabulka 4.8: Hodnoty PMSE pro různý typ optimalizace a „Testovací“ obraz

SNR 30 dB, Počet obrazů 3, 2 iterace, Regularizace TV							
Typ optimalizace	PMSE_f [%]	PMSE_h1 [%]	PMSE_h2 [%]	PMSE_h3 [%]	PMSE_h4 [%]	PMSE_h5 [%]	Čas [s]
steepdesc	52.64	54.72	5.68	102.96	-	-	213
BFGS	7.01	2.70	2.15	5.42	-	-	191
SNR 40 dB, Počet obrazů 5, 3 iterace, Regularizace TV							
Typ optimalizace	PMSE_f [%]	PMSE_h1 [%]	PMSE_h2 [%]	PMSE_h3 [%]	PMSE_h4 [%]	PMSE_h5 [%]	Čas [s]
steepdesc	26.64	38.48	5.60	69.80	4.68	5.67	637
BFGS	5.11	1.24	3.12	2.42	1.72	1.99	402

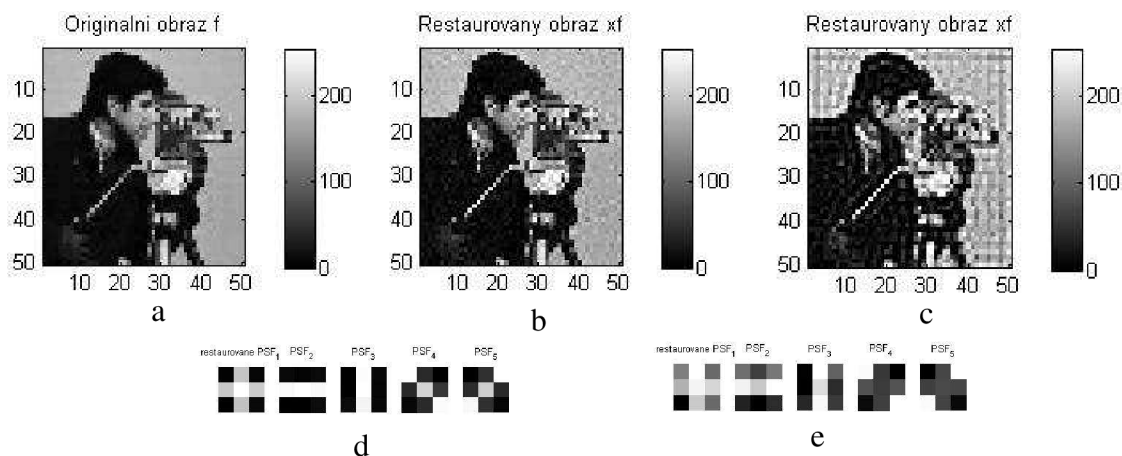
Tabulka 4.9: Hodnoty PMSE pro různý typ optimalizace a obraz „Kameraman“

SNR 30 dB, Počet obrazů 4, 3 iterace, Regularizace TV							
Typ optimalizace	PMSE_f [%]	PMSE_h1 [%]	PMSE_h2 [%]	PMSE_h3 [%]	PMSE_h4 [%]	PMSE_h5 [%]	Čas [s]
steepdesc	65.85	69.04	3.77	130.95	5.17	-	467
BFGS	24.68	40.00	9.78	57.54	9.21	-	263
SNR 30 dB, Počet obrazů 5, 5 iterací, Regularizace TV							
Typ optimalizace	PMSE_f [%]	PMSE_h1 [%]	PMSE_h2 [%]	PMSE_h3 [%]	PMSE_h4 [%]	PMSE_h5 [%]	Čas [s]
steepdesc	68.01	63.57	6.54	118.15	8.30	6.67	910
BFGS	15.01	17.49	2.20	26.22	1.58	1.97	506

Výsledky u obou typů obrazu potvrzují teoretické předpoklady (viz kapitoly 4.22 a 4.23). Kvazi-Newtonova metoda je mnohem efektivnější, neboť rychleji konverguje k minimu. Tím dosáhne lepšího výsledku za kratší dobu, což je zřejmé i z obrázků, viz **Obrázek 4.15, 4.16**.



Obrázek 4.15: Porovnání kvality MBD-AM pro různý typ optimalizace a „Testovací“ obraz, (originální obraz (a), restaurovaný obraz (b) a PSF (d) optimalizací BFGS, restaurovaný obraz (c) a PSF (e) optimalizační metodou největšího spádu)

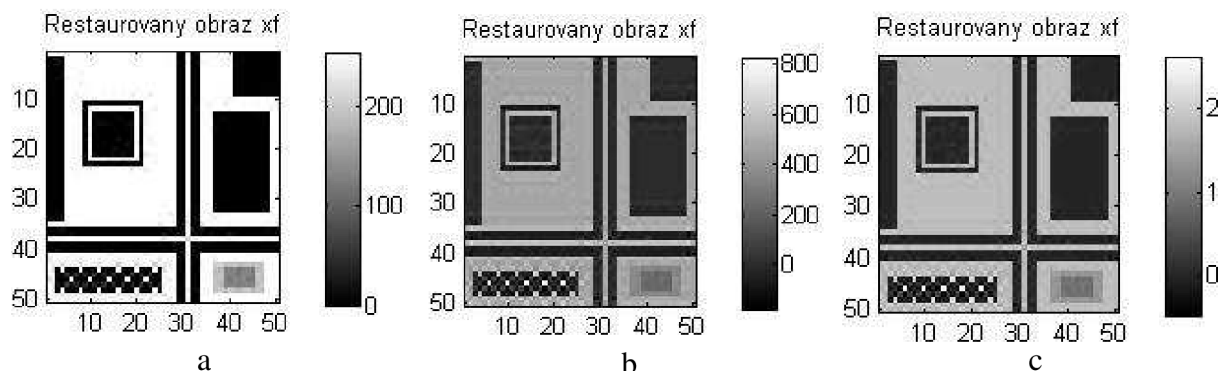


Obrázek 4.16: Porovnání kvality MBD-AM pro různý typ optimalizace a obraz „Kamera-man“, (originální obraz (a), restaurovaný obraz (b) a PSF (d) optimalizací BFGS, restaurovaný obraz (c) a PSF (e) optimalizační metodou největšího spádu)

4.6.5 Hodnocení MBD-AM z pohledu korekčních parametrů λ a γ

Hodnoty korekčních parametrů $\lambda = 0.00016$ a $\gamma = 1.3193$ jsou odhadnuty experimentálně, přičemž γ koriguje regularizační člen impulzních charakteristik $R(h)$ a λ regularizační člen obrazu $Q(f)$.

V rámci restaurační úlohy zodpovídá parametr γ především za intenzitu výsledného obrazu. Při zvýšení γ např. o tři řády dojde ke zvýšení intenzity restaurovaného obrazu, viz **Obrázek 4.17 b**, naopak při snížení γ např. o tři řády dojde ke snížení intenzity, viz **Obrázek 4.17 c**.

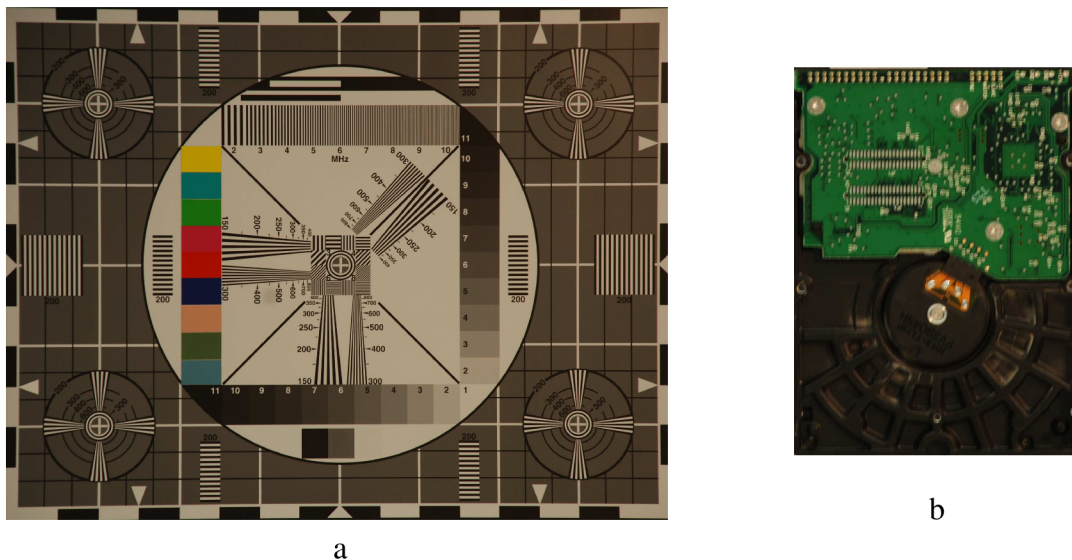


Obrázek 4.17: Porovnání kvality MBD-AM s ohledem na změnu parametru γ (restaurovaný obraz se správnou korekcí intenzity obrazu (a), restaurovaný obraz při velké hodnotě γ (b), restaurovaný obraz při malé hodnotě γ (c))

Korekční parametr λ velmi úzce souvisí s Tichonovskou nebo TV regularizací, což vyplývá ze vzorce (4.5, resp. 4.11). Pro malou hodnotu λ není patrný rozdíl mezi oběma regularizacemi, při zvýšení λ dochází ke korekci šumu v obraze, přičemž TV v případě vstupních obrazů poškozených šumem dosahuje lepších výsledků PMSE než Tichonovská regularizace. S rostoucí hladinou šumu by se měla navyšovat i hodnota λ , viz [15].

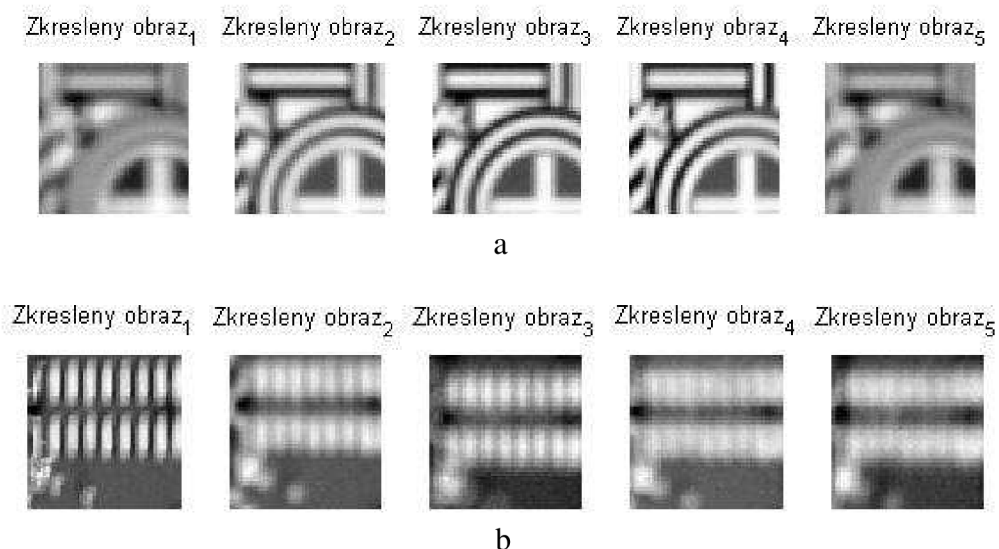
4.7 APLIKACE RESTAURAČNÍ METODY MBD-AM NA REÁLNÁ DATA

Jako reálná data byly pořízeny fotografie zkušebního obrazce pro subjektivní hodnocení prostorového rozlišení a linearity přenosu pozice (viz **Obrázek 4.18 a**) a fotografie desky plošného spoje (viz **Obrázek 4.18 b**).



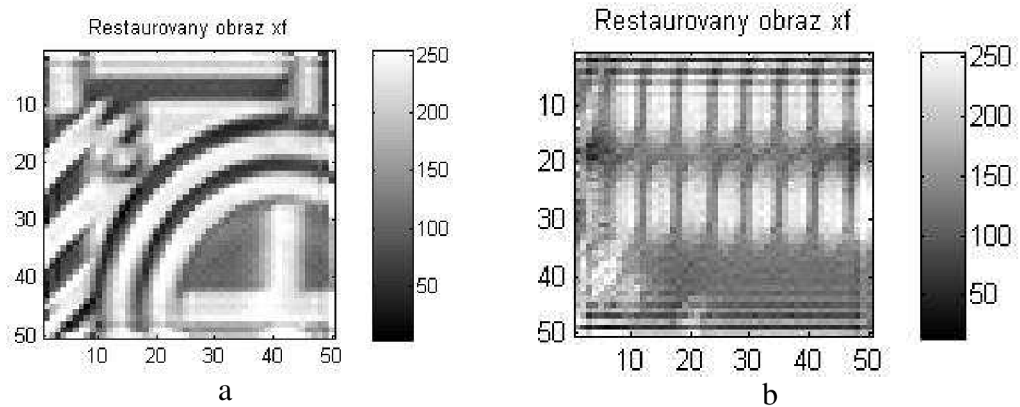
Obrázek 4.18: Reálná data (fotografie zkušebního obrazce pro subjektivní hodnocení prostorového rozlišení a linearity přenosu pozice (a), fotografie desky plošného spoje (b))

Protože se jedná o fotografie velkých rozměrů, 3872x2592 pixelů, nejprve se podvzorkují a následně se vyberou jen malé výřezy fotografií o rozměrech 50x50 pixelů, (viz **Obrázek 4.19 a, b**).



Obrázek 4.19: Zkreslené obrazy (vstupní obrazy zkušebního obrazce pro subjektivní hodnocení prostorového rozlišení a linearity přenosu pozice (a), vstupní obrazy desky plošného spoje (b))

Program na realizaci MBD-AM pro reálná data je obdobný jako u simulovaných dat, s tím rozdílem, že není potřeba uměle vytvářet zkreslené vstupní signály. Na základě předchozích poznatků se vezme 5 vstupních obrazů, použije 5 vnějších iterací IMA a zvolí kvazi-Newtonova metoda BFGS a regularizace total variation. Jelikož tentokrát není k dispozici původní, respektive nezkraslený obraz, je možné posoudit kvalitu restaurační metody pouze vizuálně, viz **Obrázek 4.20 a, b**.



Obrázek 4.20: Restaurované obrazy (obraz pro subjektivní hodnocení prostorového rozlišení a linearity přenosu pozice (a), obraz desky plošného spoje (b))

Na restaurovaných obrazech došlo ve srovnání se vstupními daty k jistému zlepšení, nicméně se nedosáhlo takového posunu, jako u simulovaných dat. Vzniklé pruhy v obraze lze přisoudit skutečnosti, že vstupní obrazy se dokonale nepřekrývají, ale jsou vzájemně posunuty.

5 ZÁVĚR

Pro realizaci v programovém prostředí Matlab byla vybrána metoda střídavého minimalizačního algoritmu pro multikanálovou slepou dekonvoluci obrazu (Alternating Minimization Algorithm for Multichannel Blind Deconvolution (MBD-AM)), využívající Bayesova přístupu a znalosti několika zkreslených obrazů daných počtem zkreslujících kanálů. Metoda je založena na střídání minimalizace funkce E (v Matlabu realizovanou for-cyklem) nejprve z pohledu PSF, a poté z pohledu obrazu. Na začátku jsou hodnoty impulzních charakteristik i obrazu odhadnuty a pak pomocí optimalizace konvergují ke správnému výsledku.

Za tímto účelem byla pomocí knihovny NAG C vytvořena optimalizační funkce simplexové metody a volána do Matlabu jako MEX soubor. Simplexová metoda je však vhodnější spíše pro úlohy s menším počtem proměnných. Příliš se neosvědčila ani metoda největšího spádu, která dlouho konverguje k minimu, a tudíž se i více projevívá zaokrouhlovací chybou. Nejlepších výsledků bylo dosaženo při aplikaci kvazi-Newtonovy optimalizační metody BFGS.

Kriteriální funkce E je dána součtem konvolučního členu, regularizačního členu impulzních charakteristik a regularizačního členu obrazu. Oba regularizační členy jsou váhovány parametry (λ a γ) ovlivňující správnost řešení. Tyto parametry se odhadují experimentálně, přičemž γ koriguje hladinu intenzity restaurovaného obrazu a λ se uplatňuje u obrazů poškozených šumem.

Při testování programu na výpočet MBD-AM na uměle vytvořených datech bylo dokázáno, že pro kvalitní restauraci obrazu stačí tři zkreslené obrazy jako vstup multikanálové dekonvoluce a dvě vnější iterace střídavého minimalizačního algoritmu, kdy dochází k výraznému zlepšení obrazu i jednotlivých PSF a dále již kvalita restaurace nijak zvlášť nestoupá.

Při testování metody na obrazech skutečných objektů, které se více blíží reálným datům, bylo zjištěno, že pro kvalitní restauraci se již potřebují čtyři zkreslené obrazy a tři iterace střídavého minimalizačního algoritmu.

Metoda je také vhodná pro zkreslené obrazy, které jsou navíc poškozené i bílým šumem, což platí pro oba typy obrazů. Pro hodnoty SNR nad 30 dB není takový šum na restaurovaném obrazu ani znát. Při SNR 20 dB a méně je šum na restaurovaném obrazu již patrný, nicméně i tak stále dochází ke zkvalitnění poškozeného obrazu.

Šum je velmi dobře korigován, kromě parametru λ , také regularizací total variation, která má na rozdíl od Tichonovské regularizace dobré anizotropní vlastnosti.

Metoda rovněž funguje i pro reálná data, i když se samozřejmě v tomto případě nedosahuje takových výsledků, jako při testování na uměle vytvořených obrazech.

Z důvodu možnosti snadné volby hodnot jednotlivých parametrů, byl program na výpočet MBD-AM vytvořen v grafickém rozhraní.

6 LITERATURA

- [1] SOVKA, P. – POLLÁK, P.: Vybrané metody číslicového zpracování signálů. Praha: Skripta FEL ČVUT, 2003. 258 s. ISBN 80-01-02821-6.
- [2] JAN, J.: Číslicová filtrace, analýza a restaurace signálů. VUT v Brně: nakladatelství VUTIUM, 2002. 427 s. ISBN 80-214-2911-9.
- [3] KOZUMPLÍK, J. – KOLÁŘ, R. – JIŘÍ, J.: Číslicové zpracování signálů v prostředí Matlab. Brno: Skripta FEKT VUT v Brně, 2001. 72 s. ISBN 80-214-1964-4.
- [4] ŠEBESTA, V. – SMÉKAL, Z.: Signály a soustavy. Brno: Vysoké učení technické v Brně, 2004. 145 s. ISBN 80-214-2434-6.
- [5] DRASTICH, A.: Netelevizní zobrazovací systémy. Brno: Vysoké učení technické v Brně, 2001. 174 s. ISBN 80-214-1974-1.
- [6] NEVŘIVA, P.: Analýza signálů a soustav. Praha: Nakladatelství BEN – technická literatura, 2000. 671 s. ISBN 80-7300-004-0.
- [7] OREL, J.: Dekonvoluce biomedicínských obrazů v digitální mikroskopii. Brno: Masarykova univerzita, Fakulta informatiky, 2006. 96 s.
- [8] ČMEJLA, R. – SOVKA, P.: Úvod do bayesovské klasifikace dat. Akustické listy, 8(2), červen 2002, str. 3-10.
- [9] <http://www.wikipedia.org>
- [10] BLUM, R. S., LIU, Z.: Multi-sensor Image Fusion And Its Applications. CRC Press, ISBN 0849334179, 2005.
- [11] STEFAN, W.: Image Restoration by blind deconvolution. Diploma Thesis. Technische Universitat Munchen, Arizona State University, 2003. 52 s.
- [12] BERTERO, M.: Deconvolution tutorial. Universita di Genova, 2003. 47s.
- [13] BARTOŠ, M.: Optimalizační metody pro restauraci obrazů. Bakalářská práce. FEKT VUT v Brně, 2005. 57 s.
- [14] <http://www.mathworks.org>
- [15] Brantál F., Šroubek F.: Vícekanálová slepá dekonvoluce vektorových obrazů. Praha: ÚTIA AV ČR, Research Report 2007/3. 91 s.
- [16] Hliněný, P.: Optimalizační úlohy. Brno: Výukový text FI MU, 2007. 96 s.

- [17] <http://www.nag.com>
- [18] Kent, J.: Visual C# 2005 bez předchozích znalostí. Brno: Computer Press a.s. 2007. 310 s. ISBN 978-80-251-1584-8.
- [19] <http://cnx.org/content/m12348/latest/>
- [20] Čermák, L.; Hlavička, R.: Numerické metody. Brno: Akademické nakladatelství Cerm, s.r.o., 2008. 110 s. ISBN 978-80-214-3752-4.
- [21] Zaplatílek K., Doňar B.: Matlab - tvorba uživatelských aplikací. Praha: Nakladatelství BEN – technická literatura, 2004. ISBN 80-7300-133-0.