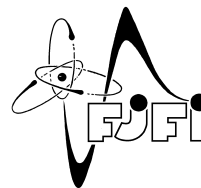




ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
Fakulta jaderná a fyzikálně inženýrská



# Slepá dekonvoluce obrazu s prostorově proměnným konvolučním jádrem

## Blind image deconvolution with space variant convolution kernel

Diplomová práce

Autor: **Bc. Antonie Brožová**  
Vedoucí práce: **Doc. Ing. Václav Šmídl, Ph.D.**  
Akademický rok: 2020/2021



## ZADÁNÍ DIPLOMOVÉ PRÁCE

Student:	Bc. Antonie Brožová
Studijní program:	Aplikace přírodních věd
Studijní obor:	Aplikované matematicko-stochastické metody
Název práce (česky):	Slepá dekonvoluce obrazu s prostorově proměnným konvolučním jádrem
Název práce (anglicky):	Blind image deconvolution with space variant convolution kernel

### Pokyny pro vypracování:

- 1) Seznamte se s problémem slepé dekonvoluce obrazu jako špatně podmíněnou inverzní úlohou. Vypracujte přehled regularizačních metod pro tuto úlohu. Zvláštní pozornost věnujte metodám založeným na pravděpodobnostních modelech.
- 2) Zaměřte se na speciální případ slepé dekonvoluce, kde konvoluční jádro je závislé na poloze pixelu v obrázku. Vypracujte přehled modelů prostorově proměnných jader a regularizačních členů používaných při jejich odhadu.
- 3) Seznamte se s daty z environmentální rastrovací elektronové mikroskopie a jejich specifickými vlastnostmi. Aplikujte existující metody na tato data a diskutujte jejich vhodnost.
- 4) Na základě předchozí analýzy navrhnete možné vylepšení stávajících modelů. Odvoďte algoritmus odhadu všech parametrů. Na syntetických datech ověřte správnou funkci navrženého algoritmu.
- 5) Aplikujte navržený algoritmus na reálná data z environmentální rastrovací elektronové mikroskopie a diskutujte výsledky.

Doporučená literatura:

- 1) D. G. Tzikas, A.C. Likas, and N. P. Galatsanos, Variational Bayesian sparse kernel-based blind image deconvolution with student's-t priors. IEEE transactions on image processing, 18(4), 2009, 753-764.
- 2) J. Kotera, V. Šmídl and F. Šroubek, Blind deconvolution with model discrepancies. IEEE Transactions on Image Processing, 26(5), 2017, 2533-2544.
- 3) H. Zhang and D. Wipf, Non-uniform camera shake removal using a spatially-adaptive sparse penalty. In Advances in Neural Information Processing Systems, 2013, 1556-1564.

Jméno a pracoviště vedoucího diplomové práce:

Doc. Ing. Václav Šmídl, Ph.D.

UTIA AV ČR, v.v.i., Pod Vodárenskou věží, 4, Praha 8

Jméno a pracoviště konzultanta:

Datum zadání diplomové práce: 28.2.2020

Datum odevzdání diplomové práce: 6.1.2021

Doba platnosti zadání je dva roky od data zadání.

V Praze dne 19. prosince 2020

.....  
garant oboru

.....  
vedoucí katedry

.....  
děkan

*Poděkování:*

Chtěla bych zde poděkovat především svému školiteli Doc. Ing. Václavu Šmídovi, Ph.D. za veškeré rady, ochotu pomoci s jakýmkoliv problémem, vstřícnost při vysvětlování problematiky a velkou trpělivost.

*Čestné prohlášení:*

Prohlašuji, že jsem tuto práci vypracovala samostatně a uvedla jsem všechnu použitou literaturu.

V Praze dne 6. 1. 2021

Antonie Brožová



*Název práce:*

## **Slepá dekonvoluce obrazu s prostorově proměnným konvolučním jádrem**

*Autor:* Antonie Brožová

*Obor:* Aplikované matematicko-stochastické metody

*Druh práce:* Diplomová práce

*Vedoucí práce:* Doc. Ing. Václav Šmídl, Ph.D., Ústav teorie informace a automatizace,  
Pod Vodárenskou věží 4, 182 08, Praha 8, Česká republika

*Abstrakt:* Tato diplomová práce se zabývá řešením slepé dekonvoluce pomocí pravděpodobnostních modelů. Srovnány budou dva přístupy. Prvním z nich je často používaná variační Bayesova metoda, pro kterou se obvykle musí volit pravděpodobnostní rozdělení z konjugovaného systému hustot, což je velmi omezující. Druhým zde navrženým postupem proto bude optimalizace ELBO (evidence lower bound) pomocí gradientního sestupu s využitím reparametrizačního triku, která dává ve volbě rozdělení větší volnost. Kromě těchto dvou metod bude na testovacích datech vyzkoušen i optimalizátor Vadam. Následně budou nejlepší algoritmy použity pro odhad variance v nekompletním obraze s cílem usnadnit skenování v environmentální rastrovací elektronové mikroskopii.

*Klíčová slova:* slepá dekonvoluce obrazu, variační Bayesova metoda, ELBO, Vadam

*Title:*

## **Blind image deconvolution with space variant convolution kernel**

*Author:* Antonie Brožová

*Abstract:* The topic of this diploma thesis is the use of probabilistic models for blind image deconvolution. Two approaches will be compared. Commonly used variational Bayes, which is the first approach, usually requires priors to be conjugate, which is very restrictive. Therefore, the second approach is optimization of ELBO (evidence lower bound) via gradient descent with the use of a reparametrization trick, which allows to choose priors more freely. Apart from these two methods, Vadam optimizer will be examined on a test dataset. Consequently, the best algorithms will be used to estimate variance of an incomplete image in order to facilitate image acquisition in an environmental scanning electron microscopy.

*Key words:* blind image deconvolution, variational Bayes method, ELBO, Vadam





# Obsah

<b>Úvod</b>	<b>13</b>
<b>1 Slepá dekonvoluce obrazu</b>	<b>15</b>
1.1 Konvoluce ve zpracování obrazu . . . . .	15
1.2 Druhy rozmazání . . . . .	16
1.3 Slepá dekonvoluce . . . . .	16
1.4 Pravděpodobnostní modely pro slepou dekonvoluci . . . . .	17
1.4.1 Totální variace . . . . .	18
1.4.2 Autoregresní modely . . . . .	19
1.4.3 Super-gaussovské rozdělení . . . . .	19
1.5 Neuronové sítě pro slepou dekonvoluci . . . . .	20
1.6 Slepá dekonvoluce s prostorově proměnným jádrem . . . . .	20
<b>2 Bayesovský přístup ke slepé dekonvoluci</b>	<b>23</b>
2.1 Variační Bayesova metoda . . . . .	23
2.2 ELBO . . . . .	27
2.3 Vadam . . . . .	28
2.4 Model rozmazaného obrazu . . . . .	29
2.5 Volba apriorních rozdělení . . . . .	29
2.6 Odvození VB . . . . .	30
2.7 Odvození ELBO . . . . .	33
2.8 Tvary momentů . . . . .	34
<b>3 Srovnání metod na testovacích datech</b>	<b>35</b>
3.1 Odhad pomocí VB . . . . .	37
3.2 Odhad pomocí ELBO . . . . .	42
3.3 Tridiagonální Sk . . . . .	47
3.4 Větší počet vzorků pro reparametrizaci . . . . .	49
3.5 VADAM . . . . .	53
3.6 Zhodnocení experimentu . . . . .	55
3.7 Odhad nosiče . . . . .	59
<b>4 Odhad z nekompletního snímku</b>	<b>63</b>
4.1 Vícerozměrná variance šumu . . . . .	63
4.2 Odvození pro ELBO . . . . .	64
4.3 Chybějící pixely . . . . .	65
4.4 Odhad přesnosti v chybějících oblastech . . . . .	65

4.5	Míra poškození . . . . .	69
4.6	Snímek z mikroskopu . . . . .	73
<b>Závěr</b>		<b>77</b>

## Použité značení

---

$\mathbb{N}$	Přirozená čísla
$\mathbb{R}$	Reálná čísla
$\mathbb{R}_+$	Nezáporná reálná čísla
$\langle 0, 1 \rangle$	Uzavřený interval od 0 do 1
$\mathbf{U} \in \mathbb{R}^{p \times r}$	Reálná matice $\mathbf{U}$ o rozměru $p \times r$
$\mathbf{u} \in \mathbb{R}^p$	Reálný $p$ -rozměrný vektor $\mathbf{u} = (u_1, u_2, \dots, u_p)^T$
$\bar{U}_k$	$k$ -tý řádek matice $\mathbf{U}$ (řádkový vektor)
$\mathbf{U}^T$	Transpozice matice $\mathbf{U}$
$\mathbf{u}^{-1}$	Vektor převrácených hodnot vektoru $\mathbf{u}$
$\mathbf{U}^{-1}$	Inverze matice $\mathbf{U}$
$\mathbb{I}_p$	Jednotková matice o rozměru $p \times p$
$\mathbf{0}_{p,r}$	Nulový vektor
$u_{i,j}$	$(i, j)$ -tý prvek matice $\mathbf{U}$
$\hat{\mathbf{U}}$	Odhad $\mathbf{U}$
$\mathbb{E}[\cdot]$	Střední hodnota
$\mathbb{E}_f[\cdot]$	Střední hodnota vzhledem k hustotě pravděpodobnosti $f$
$f^{(n)}$	$n$ -tá iterace $f$
$\text{diag}(\mathbf{u})$	Diagonální matice rozměru $p \times p$ , jejíž diagonální prvky popořadě odpovídají prvkům $\mathbf{u}$ , $\mathbf{u} \in \mathbb{R}^p$
$\text{diag}(\mathbf{U})$	$p$ -rozměrný vektor, jehož prvky po řadě odpovídají diagonálním prvkům $\mathbf{U}$ , $\mathbf{U} \in \mathbb{R}^{p \times p}$
$\text{tr}(\mathbf{U})$	Stopa matice $\mathbf{U} \in \mathbb{R}^{p \times p}$
$ \mathbf{U} $	Determinant matice $\mathbf{U}$
$\mathbf{u} \circ \mathbf{w}$	Hadamardův produkt $\mathbf{u}$ a $\mathbf{w}$ , $\mathbf{u} \circ \mathbf{w} = (u_1 v_1, u_2 v_2, \dots, u_p v_p)^T$ , $\mathbf{u}, \mathbf{w} \in \mathbb{R}^p$
$\text{vec}(\mathbf{U})$	Vektorizace matice $\mathbf{U} \in \mathbb{R}^{p \times r}$ , $\text{vec}(\mathbf{U}) =$ $= (u_{1,1}, u_{2,1}, \dots, u_{p,1}, u_{1,2}, \dots, u_{p,2}, u_{1,3}, \dots, u_{p,r})^T$
$f(u)$	Hustota pravděpodobnosti náhodné veličiny $u$
$f(u \mathbf{D})$	Podmíněná hustota pravděpodobnosti náhodné veličiny $u$
$\tilde{f}(\cdot)$	Aproximace hustoty $f(\cdot)$
$\delta(u)$	Diracova delta funkce, pro spojitou proměnnou $u \in \mathbb{U}$ : $\int_{\mathbb{U}} \delta(u)g(u) du = \int_{\mathbb{U}} \delta(u)g(u) du = g(0)$
$L$	Logaritmus sdružené posteriorní hustoty
$L_u$	Prvky $L$ obsahující $u$
$\mathbf{u} \sim F(\mathbf{s})$	Náhodná veličina s rozdělením $F$ s parametry $\mathbf{s}$
$\mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u)$	Náhodná veličina $\mathbf{u}$ s normálním rozdělením se střední hodnotou $\boldsymbol{\mu}_u$ a kovarianční maticí $\boldsymbol{\Sigma}_u$
$u \sim \mathcal{G}(\alpha_u, \beta_u)$	Náhodná veličina $u$ s gama-rozdělením s parametry $\alpha_u$ (shape), $\beta_u$ (rate)
$\mathbf{U} \circledast \mathbf{W}$	Konvoluce matic $\mathbf{U}$ a $\mathbf{W}$
$\nabla_u$	Gradientní operátor vzhledem k proměnné $\mathbf{u}$
$\frac{\partial}{\partial u}$	Parciální derivace vzhledem k $u$
$\mathcal{L}$	ELBO
$\mathcal{L}_u$	Prvky ELBO obsahující $u$



# Úvod

Každý obraz zachycený libovolným zařízením je ve srovnání s originální scénou nějakým způsobem nepřesný, může se jednat například o šum nebo rozmazání. Vzhledem k vysokému množství snímků, které vznikají každý den, je odstranění těchto poškození velké téma. Tato diplomová práce se zabývá hledáním ostrého obrazu z rozmazaného za předpokladu, že způsob rozmazání není známý, tedy slepou dekonvolucí. Lze ji řešit pomocí pravděpodobnostního modelu, jedním z častých přístupů je variační Bayesova metoda, která je ale omezující vzhledem k volbě pravděpodobnostních rozdělení. Například v případě prostorově proměnné PSF je velmi nepravděpodobné, že by bylo možné pomocí této metody převést problém na soustavu lineárních rovnic, které variační Bayesova metoda využívá k hledání aposteriorních rozdělení. Proto bude zkoumán alternativní přístup, kdy jsou odhady hledány pomocí optimalizace ELBO (evidence lower bound) gradientním sestupem s využitím reparametrizačního triku. Kromě klasického optimalizátoru Adam bude použit i optimalizátor Vadam, který slibuje snadnou implementaci a výsledky srovnatelné s variačními metodami.

Je-li slepá dekonvoluce řešena pomocí pravděpodobnostního modelu, kromě odhadu ostrého obrazu můžeme získat i další informaci, například varianci jednotlivých pixelů. To by mohlo být přínosné v environmentální rastrovací elektronové mikroskopii, která je jedním z moderních trendů v elektronové mikroskopii. Umožňuje zkoumání vzorků živé i neživé přírody v podmínkách vysokého tlaku plynů až 3000 Pa, ve kterých se nejen povrch nevodivého preparátu nenabíjí, ale dokonce lze uchránit vlhký vzorek před vyschnutím [1]. Snímání obrazu probíhá pixel po pixelu a zajímavá bývá pouze určitá oblast snímku. Pokud ji půjde od okolí odlišit už po naměření menšího počtu pixelů, nebude pak nutné skenovat celý obraz.

V první kapitole budou shrnuty metody používané pro řešení slepé dekonvoluce. Následuje kapitola s detailnějším popisem variační Bayesovy metody a ELBO a na základě této teorie bude postaven model založený pouze na variační Bayesově metodě a model využívající ELBO. Jejich přínos bude ve třetí kapitole porovnán na testovacích datech. V poslední kapitole bude zkoumána schopnost najít zajímavé oblasti v obraze s chybějícími pixely a to jak na testovacích datech, tak na snímku z environmentální rastrovací elektronové mikroskopie.



# Kapitola 1

## Slepá dekonvoluce obrazu

Rozmazaný obraz lze popsat pomocí konvoluce ostrého obrazu a operátoru rozmazání. Dekonvolucí pak rozumíme operaci, která tento postup v podstatě invertuje a z rozmazaného obrazu zrekonstruuje původní, ostrý obraz. V případě, kdy kromě ostrého obrazu není znám ani tvar rozmazání, je dekonvoluce označována jako slepá. Jedná se o špatně podmíněnou inverzní úlohu, jejíž řešení vzbuzuje velký zájem.

V této kapitole bude popsán vznik rozmazaného obrazu, druhy rozostření a způsoby, kterým lze ostrý obraz nalézt. Největší důraz bude kladen na popis obrazu pomocí pravděpodobnostních rozdělení a budou popsány nejčastěji volená rozdělení. Zmíněno bude i použití neuronových sítí pro řešení slepé dekonvoluce.

### 1.1 Konvoluce ve zpracování obrazu

Označme  $D$  naměřený obraz,  $\tilde{X}$  původní nerozmazaný obraz a  $\tilde{K}$  jako konvoluční jádro. Konvoluce se obvykle značí jako

$$\tilde{K} \circledast \tilde{X},$$

takže za předpokladu aditivního šumu  $N$  lze model zapsat jako

$$D = \tilde{K} \circledast \tilde{X} + N.$$

U obrazu se využívá její diskrétní varianta. Necht' jsou  $D$  i  $\tilde{X}$  rozměru  $n \times p$  a  $\tilde{K}$  rozměru  $s \times s$ , kde  $s = 2u + 1$ ,  $u \in \mathbb{N}$ . Pak je  $(i, j)$ -tý pixel rozmazaného obrazu dán jako

$$D(i, j) = \sum_{n=-u}^u \sum_{m=-u}^u \tilde{K}(n, m) \tilde{X}(i - n, j - m) + N(i, j). \quad (1.1)$$

Aby tento výraz platil pro každý pixel rozmazaného obrázku, je třeba  $\tilde{X}$  po obvodu rozšířit - u každého okraje o  $u$  pixelů. Nejjednodušším způsobem je rozšířit obraz pouze o nuly, pak přidaný okraj přispívá do konvoluce pouze nulovými prvky. Tento postup nemusí být vždy výhodný, neboť snižuje hodnoty na okrajích obrazu získaného konvolucí, což se může negativně projevit při následné dekonvoluci. Konvoluci lze pak zapsat v maticovém tvaru jako

$$d = Kx + n, \quad (1.2)$$

kde  $\mathbf{d}$ ,  $\mathbf{x}$  a  $\mathbf{n}$  jsou po sloupcích vektorizované matice  $\mathbf{D}$ ,  $\mathbf{X}$  a  $\mathbf{N}$  a  $\mathbf{K}$  je dvojitá bloková Toeplitzova matice vytvořená z  $\tilde{\mathbf{K}}$ . Toeplitzova matice je matice, pro kterou platí, že každá její diagonála má všechny prvky stejné. Dvojitá bloková Toeplitzova matice je bloková matice, která má v každé blokové diagonále stejné Toeplitzovy matice. Dalšími možnostmi jsou například zrcadlové prodloužení nebo periodické prodloužení, které mohou přinášet velmi podobné nepřesnosti jako nulové rozšíření.

## 1.2 Druhy rozmazání

K rozmazání obrazu dochází v důsledku různých nepřesností při snímání. Pokud by byl snímek zachycen perfektně, funkce, která zobrazuje scénu do obrazu, by byla Diracova  $\delta$ -funkce (pokud zanedbáme šum). V reálném světě však dochází například ke zkreslení v důsledku tvaru čočky fotoaparátu nebo pohybem kamery. Funkce, která zobrazuje scénu na snímek se označuje rozptylová funkce (point spread function - PSF). Mezi základní typy rozostření patří gaussovské rozostření, rozmazání pohybem nebo vzniká v důsledku špatného zaostření (out-of-focus). Většina modelů předpokládá prostorově invariantní rozostření, což znamená, že každý pixel rozmazaného obrazu je získán konvolucí s tím samým konvolučním jádrem.

Pohyb je často příčinou rozostření fotografie - může se jednat o pohyb fotografa, pohyb scény nebo pohyb některého objektu v rámci scény. Relativní pohyb kamery a scény může mít mnoho podob: translace, rotace, změna škály nebo jejich kombinace. Pokud je pohyb dostatečně rychlý vzhledem k expozici fotoaparátu, je možné ho aproximovat lineárním pohybem a tedy modelovat parametricky, například jako průměr  $L$  pixelů ve směru pohybu [10]. Ve většině reálných situací to však možné není, například drobné pohyby fotoaparátu, které jsou důsledkem špatné stability fotografa, jsou kombinací mnoha druhů pohybu. Nedají se proto tak snadno zapsat parametricky, pomocí pravděpodobnostního modelu modelu ho řešili například Fergus et al. v [14] nebo Shao et al. v [40]. Tai et al. v [47] řeší případ, kdy je rozmazání prostorově proměnné pomocí projective motion path - rozmazaný obraz berou jako průměr obrazů rozmazaných sadou pohybů, kterou kamera v průběhu snímání prošla.

Rozostření způsobené turbulentním prostředím, například atmosférou nebo vodou, je možné při dlouhé expozici modelovat jako dvourozměrnou gaussova funkci, jejíž směrodatná odchylka udává rozměr konvolučního jádra. Jako aproximace PSF se používá také ve fluorescenčních mikroskopech [57] nebo v rastrovacích elektronových mikroskopech, i když podle [29] v druhém případě příliš neodpovídá realitě.

I samotná snímací aparatura přináší do obrazu jisté zkreslení, závisí například na velikosti a počtu čoček nebo na jejich vzdálenosti od scény. Pokud je obraz výrazně špatně zaostřený, PSF se často modeluje jako kruh [10].

## 1.3 Slepá dekonvoluce

Model rozmazaného obrazu předpokládá, že vznikl konvolucí ostrého obrazu a PSF. Je to špatně podmíněná úloha, pro jejíž řešení bylo navrženo mnoho postupů.

Ke slepé dekonvoluci lze přistoupit dvěma způsoby - buď je nejdříve odhadnuto jádro a následně provedena neslepá dekonvoluce nebo se konvoluční jádro odhaduje společně s obrazem.

První zmíněná varianta se používá v případech, kdy dokážeme například změřit parametry figurující v parametrickém modelu rozostření nebo když je lze odhadnout na základě jiných snímků a předpokládaného typu PSF. Kupříkladu v případě rozostření lineárním pohybem nebo



v důsledku špatného zaostření obsahují Fourierovy obrazy PSF nuly a v jejich vzorech lze nalézt potřebné hodnoty parametrů.

Existuje celá řada algoritmů pro řešení dekonvoluce, pokud je konvoluční jádro známé (nebo už odhadnuté). Nejjednodušší variantou je použít maticový zápis (1.2) a pouze matici  $\mathbf{K}$  invertovat. Tento postup ve většině případů selhává, protože matice  $\mathbf{K}$  bývá špatně podmíněná. To lze řešit regularizací - přičtením malého čísla k diagonále matice. Inverzní filtrování lze provést i ve frekvenčním spektru. Využívá se vztahu pro Fourierův obraz konvoluce, který říká, že Fourierův obraz konvoluce odpovídá součinu Fourierových obrazů PSF a ostrého obrazu. V případě, kdy jsou hodnoty rozostření ve frekvenčním spektru příliš malé nebo dokonce nulové, tento přístup není vhodný [35]. Další možností je Wienerův filtr. Tato metoda předpokládá, že oba Fourierovy obrazy i šum jsou náhodné proměnné, a je hledán takový filtr, pro který je minimalizována střední kvadratická odchylka reálného obrazu od jeho odhadu. Pro lepší řešení rekonstrukce za přítomnosti šumu byly navrženy i jiné postupy, například parametrický Wienerův filter nebo filtrování pomocí omezených nejmenších čtverců (constrained least square filtering).

Další oblíbenou metodou je Lucy-Richardsonova dekonvoluce [30], [37]. Jedná se o algoritmus založený na Bayesově větě, který iterativně odhaduje ostrý obraz. Později byl rozšířen i pro slepou dekonvoluci, např. v [15].

## 1.4 Pravděpodobnostní modely pro slepou dekonvoluci

Postup, kdy je nejdříve odhadnuto jádro a následně je provedena slepá dekonvoluce obvykle vyžaduje nějaké známé podmínky a není příliš univerzální, proto se častěji zkoumá varianta společného odhadu obrazu a jádra. Následující metody jsou prezentovány jako metody pro společný odhad obrazu a jádra, nicméně ve většině případů je možné je použít i když je jádro známé a hledání ostrého obrazu se tím zjednoduší.

Jeden ze základních postupů zahrnuje minimalizaci kvadratické chyby rekonstrukce za určitých podmínek na  $\mathbf{k}$  a  $\mathbf{x}$  vyplývajících ze znalosti konkrétního problému, což lze zapsat jako

$$\min_{\mathbf{x}, \mathbf{k}} \left( \|\mathbf{k} \otimes \mathbf{x} - \mathbf{d}\|_2^2 - \lambda_1 J(\mathbf{k}) - \lambda_2 G(\mathbf{x}) \right), \quad (1.3)$$

kde  $\lambda_1$  a  $\lambda_2$  jsou Lagrangeovy multiplikátory a  $J(\mathbf{k})$ , resp.  $G(\mathbf{x})$  je regularizace na  $\mathbf{k}$ , resp.  $\mathbf{x}$ , např. [20], [34].

Problém slepé dekonvoluce je možné řešit i pomocí Bayesovské statistiky [5], [31], [26], [50]. Bayesovská formulace dekonvoluce se zakládá na Bayesově vzorci

$$f(\mathbf{x}, \mathbf{k} | \mathbf{d}) = \frac{f(\mathbf{d} | \mathbf{x}, \mathbf{k}) f(\mathbf{x}, \mathbf{k})}{f(\mathbf{d})}. \quad (1.4)$$

Je nutná znalost apriorního rozdělení  $f(\mathbf{x}, \mathbf{k})$ , pro které se obvykle předpokládá, že  $f(\mathbf{x}, \mathbf{k}) = f(\mathbf{x})f(\mathbf{k})$ . To se volí na základě očekávaných vlastností obrazu a PSF, nejčastěji se využívají apriorna, která vynucují hladkost nebo řídkost řešení. Dále je třeba zvolit  $f(\mathbf{d} | \mathbf{x}, \mathbf{k})$ , například jako gaussovské rozdělení předpokládáme-li gaussovský šum. Hustota  $f(\mathbf{d})$  nezávisí na  $\mathbf{x}$  ani na  $\mathbf{k}$  a proto ji často není potřeba znát pro nalezení odhadu a používá se tvar

$$p(\mathbf{x}, \mathbf{k} | \mathbf{d}) \propto p(\mathbf{d} | \mathbf{x}, \mathbf{k}) p(\mathbf{x}, \mathbf{k}).$$

Nejznámější metodou je maximum-aposteriori (MAP) odhad. Odhadem jádra a ostrého obrazu je pak argument maxima aposteriorního rozdělení, tedy

$$\hat{\mathbf{x}}, \hat{\mathbf{k}} = \operatorname{argmax}_{\mathbf{x}, \mathbf{k}} p(\mathbf{x}, \mathbf{k} | \mathbf{d}).$$

Levin et al. ukázali v [28], že za předpokladu rovnoměrného apriorního rozdělení prvků jádra, vede hledání MAP odhadu pro  $\mathbf{x}$  a  $\mathbf{k}$  k tzv. no-blur řešení, kde je odhad ostrého obrazu roven rozmazanému a konvoluční jádro Diracově  $\delta$ -funkci. Navrhují využít toho, že pro jádro je dostupný mnohem větší počet měření, neboť většinou jsou rozměry jádra mnohem menší než rozměr obrazu, a hledat MAP odhad jádra a odhad obrazu pak získat marginalizací sdružené aposteriorní pravděpodobnosti. Jejich analýzu dále rozšiřují Perrone a Favaro v [34].

Další možností je variační Bayesova metoda, která hledá řešení pomocí aproximace sdružené aposteriorní hustoty. Aproximující hustotu lze faktorizovat na aposteriorní rozdělení jednotlivých parametrů a ty se hledají takové, aby byla Kullback-Leiblerova (KL) vzdálenost od reálné hustoty minimální. Označíme-li jako  $\boldsymbol{\theta}$  neznámé náhodné proměnné modelu, pak má aproximující hustota tvar

$$\hat{f}(\boldsymbol{\theta}|\mathbf{d}) = \prod_i \hat{f}(\theta_i|\mathbf{d}),$$

a KL vzdálenost aproximace  $\hat{f}(\boldsymbol{\theta}|\mathbf{d})$  od reálné sdružené hustoty  $f(\boldsymbol{\theta}|\mathbf{d})$  je rovna

$$KL(\hat{f}(\boldsymbol{\theta}|\mathbf{D}) \| f(\boldsymbol{\theta}|\mathbf{D})) = \mathbb{E}_{\hat{f}(\boldsymbol{\theta}|\mathbf{D})} \left[ \ln \frac{\hat{f}(\boldsymbol{\theta}|\mathbf{D})}{f(\boldsymbol{\theta}|\mathbf{D})} \right],$$

kde  $\mathbb{E}_{\hat{f}(\boldsymbol{\theta}|\mathbf{D})} [\cdot]$  značí střední hodnotu vzhledem ke  $\hat{f}(\boldsymbol{\theta}|\mathbf{D})$ . Tato metoda bude detailněji popsána v kapitole 2.

Důležitým aspektem je správná volba apriorního rozdělení obrazu, neboť jeho rozměr je obvykle značně větší než rozměr konvolučního jádra a proto o něm není v rozmazané obrazu tolik informace. Avšak neznamená to, že musí nutně odpovídat přirozené představě o rozdělení obrazu, v některých případech je důležitější volit takové apriorní rozdělení, které se dokáže vyhnout lokálnímu extrému, především no-blur řešení. Diskuzi k MAP i variačnímu Bayesovi pro slepou dekonvoluci obrazu lze nalézt v [56], v následujících odstavcích budou představena nejběžněji volená apriorní rozdělení.

### 1.4.1 Totální variace

Totální variaci (TV) navrhli jako první využít k odstranění šumu Rudin, Osher a Fatemi v [38]. Tvrdí, že totální variace je jako regularizace pro zpracování obrazu vhodnější než  $L_2$  norma, která se snadněji minimalizuje. Isotropní totální variace má tvar

$$TV(\mathbf{x}) = \|\sqrt{|\nabla_h \mathbf{x}|^2 + |\nabla_v \mathbf{x}|^2}\|,$$

kde  $\nabla_h$ , resp.  $\nabla_v$  je operátor horizontální derivace  $(x_{i,j} - x_{i,j-1})$ , resp. vertikální derivace  $(x_{i,j} - x_{i-1,j})$  a anisotropní

$$TV(\mathbf{x}) = \|\|\nabla_h \mathbf{x}| + |\nabla_v \mathbf{x}|\|.$$

Většinou se volí isotropní totální variace, neboť anisotropní je pro šikmé hrany vyšší než by měla být [12]. TV je vhodná v případech, kdy má odhadovaná proměnná výraznější hrany, neboť pomáhá nalézt odhad s řídkými gradienty a proto se častěji používá jako apriorno na hledaný obraz.

Mezi prvními použili TV regularizaci pro řešení slepé dekonvoluce Chan a Wong [20]. Úlohu formulovali ve tvaru (1.3) a výraz střídavě minimalizovali vzhledem k  $\mathbf{x}$  a  $\mathbf{k}$ . Řešení není jednoznačné, proto požadovali nezápornost jádra i obrazu, středovou symetrii jádra a aby byl součet

prvků jádra roven jedné. Tato úloha odpovídá hledání MAP odhadu předpokládáme-li normální rozdělení  $\mathbf{d}$  se střední hodnotou  $\mathbf{K}\mathbf{x}$  a jednotkovou kovarianční maticí a TV apriornem na  $\mathbf{x}$  a  $\mathbf{k}$ , které v podstatě odpovídá Laplaceově rozdělení jejich gradientů.

Babacan et al. použili TV apriorno na  $\mathbf{x}$  v rámci variační Bayesovy metody v [5]. Pro apriorní rozdělení  $\mathbf{k}$  zvolili SAR model (popsaný v další sekci). Jako aposteriorní rozdělení obou proměnných zvolili v prvním případě Gaussovské a v druhém Diracovo  $\delta$  a ukázali, že druhá varianta není schopna nalézt správný rozklad při větší míře šumu.

### 1.4.2 Autoregresní modely

Autoregresní modely je možné použít pro apriorní rozdělení předpokládáme-li, že se náhodná proměnná mění hladce. Molina et al. použili v [31] modely SAR (simultaneous autoregression) a CAR (conditional autoregression) jako apriorní rozdělení obrazu. V prvním případě toto rozdělení odpovídá normálnímu rozdělení druhých derivací obrazu (Laplaceův operátor) s nulovou střední hodnotou a diagonální kovariancí se stejnými hodnotami pro všechny prvky. Druhý model také využívá normální rozdělení s nulovou střední hodnotou a přesností (inverze kovarianční matice)  $\alpha(\mathbb{I} - \phi\mathbf{N})$ , kde  $\alpha$  a  $\phi$  jsou parametry a  $\mathbf{N}$  je matice, jejíž  $(i, j)$ -tý prvek je nenulový, pokud spolu pixely  $i$  a  $j$  sousedí. V obou dvou případech se tedy jedná o apriorna, která předpokládají nějaký vztah mezi sousedními pixely obrazu. Dle [52] je možné SAR model jednoznačně přepsat na CAR model. Apriorní rozdělení konvolučního jádra zvolili Dirichletovo a ukázali, že je flexibilní a dá se využít pro různé druhy rozostření.

Pomocí variačního Bayese řešili slepou dekonvoluci se SAR apriornem na  $\mathbf{k}$  i  $\mathbf{x}$  Molina et al. v [32]. Srovnávali dvě volby aposteriorních rozdělení: normální a degenerované (Diracovo  $\delta$ ). Při větší míře šumu (SNR 20dB) se ukázalo, že druhá možnost vede na no-blur řešení, nicméně při nižším šumu (SNR 40 dB) je odhad lepší než s normálním apriornem.

Dekonvoluci lze řešit také pomocí ARMA (Autoregressive Moving Average) modelu. Předpokládá se, že ostrý obraz vznikl autoregresním procesem a konvoluční jádro je modelováno jako klouzavý průměr. Pomocí EM (Expectation-Maximization) algoritmu ho řešili například Lagen-dijk et al. v [27].

### 1.4.3 Super-gaussovské rozdělení

Super-gaussovská rozdělení mají ostřejší vrchol a těžší chvosty než gaussovské. Podle Palmera et al. [33] nazýváme rozdělení super-gaussovské, pokud má náhodná veličina kladný koeficient špičatosti (kurtosis) a sub-gaussovské, pokud ho má záporný. Tato rozdělení jsou pak řídká, neboť dávají vyšší pravděpodobnost hodnotám kolem nuly. Super-gaussovské rozdělení ostrého obrazu po aplikaci různých filtrů uvažují Babacan et al. v [6], jeho obecný tvar je

$$p(\mathbf{x}) = \prod_i p(x_i) = Z \exp \left[ - \sum_i \rho(x_i) \right],$$

kde  $x_i$  značí  $i$ -tý pixel obrazu  $\mathbf{x}$ ,  $Z$  je normalizační konstanta a pro  $\rho$  (podle [33]) platí, že  $\rho(\sqrt{y})$  je klesající a konkávní pro  $s \in (0, \infty)$ . Jako  $\rho$  volili  $\mathcal{L}_p$  normu, exponenciálu a logarimus, přičemž poslední zmíněná našla výrazně lepší odhad na testovacím obrázku.

Zvolíme-li jako apriorní rozdělení gradientů obrázku gaussovské rozdělení s přesností  $\tau_i$  pro každý pixel  $i$  a  $\tau_i$  přiřadíme apriorní gama-rozdělení, marginalizací získáme super-gaussovské rozdělení. Tento model odpovídá směsi gaussovských rozdělení s různou škálou (gaussian scale mixture). Tzikas et al. [50] ho zvolili nejen pro obraz a jádro, ale i pro šum, podobně jako

Kotera et al. [26], kteří tento model nazývají ARD (Automatic Relevance Determination [44]). Toto apriorní rozdělení dokáže dobře modelovat ostrý obraz a zároveň je jeho tvar dostatečně jednoduchý pro snadnou optimalizaci.

## 1.5 Neuronové sítě pro slepou dekonvoluci

V posledních letech se pro rekonstrukci obrazu často používají neuronové sítě, ale ve většině případů se předpokládá velká trénovací množina a jsou obvykle použitelné pouze pro konkrétní typy obrázků. Učení sítě pouze na základě jednoho obrázku se někdy označuje jako zero-shot učení. Příkladem použití zero-shot učení pro zpracování obrazu je třeba zvyšování rozlišení [41], kde se používá konvoluční neuronová síť. Pro slepou dekonvoluci používají hlubokou neuronovou síť inspirovanou Lucy-Richardsonovým algoritmem Agarwal et al. v [3].

Ulyanov et al. představili v roce 2018 *deep image prior* (DIP) [51]. Vycházejí z myšlenky, že neuronová síť potřebuje nejen dostatečně velký dataset pro dosažení dobré rekonstrukce obrazu, ale i vhodnou strukturu. Místo aby trénovali síť na velkém množství dat, používají pouze jeden degradovaný snímek a váhy takto naučené neuronové sítě vnímají jako jeho parametrizaci

$$\mathbf{x} = f_{\theta}(\mathbf{z}),$$

kde  $\theta$  jsou parametry sítě a  $\mathbf{z}$  je náhodná realizace šumu. Pro slepou dekonvoluci se DIP rozhodli použít Ren et al. v [36]. Pro zachycení struktury ostrého obrazu použili asymetrický autoencoder se zkratkovými spojeními (skip connections) a pro rozostření plně propojenou síť. Označíme-li  $f_x$  a  $f_k$  generativní síť příslušející ostrému obrazu  $\mathbf{x}$  a rozostření  $\mathbf{k}$ , hledají řešení slepé dekonvoluce minimalizací

$$\|f_x(\mathbf{z}_x) \otimes f_k(\mathbf{z}_k) - \mathbf{d}\|_2^2, \quad (1.5)$$

s podmínkami  $0 \leq f_x(\mathbf{z}_x)_i \leq 1, \forall i$  a  $f_k(\mathbf{z}_k)_j \geq 0, \sum_j f_k(\mathbf{z}_k)_j = 1, \forall j$ , kde  $\mathbf{z}_x, \mathbf{z}_k$  jsou vzorky z uniformního rozdělení. Takto navržený model není vhodný pro obrázky, ve kterých je významnější šum, proto k (1.5) přičítají ještě výraz  $\lambda TV(f_x(\mathbf{x}))$ . Později navrhují i využít ztrátovou funkci která je kombinací structural similarity index measure [53] a střední kvadratické odchylky.

## 1.6 Slepá dekonvoluce s prostorově proměnným jádrem

U slepé dekonvoluce s prostorově proměnným jádrem se předpokládá, že určité oblasti nebo každý pixel snímku mají odlišná konvoluční jádra. Nejedná se o dekonvoluci v pravém slova smyslu, neboť v tomto případě už model nelze zapsat přímo jako konvoluci, výraz (1.1) má obecnější tvar

$$\mathbf{D}(i, j) = \sum_{n=-u}^u \sum_{m=-u}^u \tilde{\mathbf{K}}(n, m, i, j) \tilde{\mathbf{X}}(i - n, j - m) + \mathbf{N}(i, j),$$

kde rozostření  $\tilde{\mathbf{K}}$  závisí i na poloze pixelu  $(i, j)$ . Úloha se ale stále dá přepsat do maticového tvaru, s tím rozdílem, že struktura konvoluční matice už není dvojité blokované Toeplitzova matice. Řešení se hledá o poznání hůře, minimálně je nutné mít nějaký předpoklad o tvaru rozmazání. Pokud se konvoluční jádro mění v rámci obrazu pomalu, je možné konvoluci uvažovat lokálně.

Jednou z možností jak získat více informací pro úspěšné doostření obrazu je použít senzory, které sledují pohyb fotoaparátu nebo měří vzdálenost fotoaparátu od scény. Například Joshi et al.

v [21] řeší prostorově proměnnou dekonvoluci obrazu Bayesovským přístupem s pomocí senzorů, které zachycují rotace a translace kamery během snímání. Takový postup je však nepraktický, vyžaduje mít senzory dopředu připravené.

Dalším často volenou cestou při odstraňování prostorově proměnného rozmazání je použití více snímků té samé scény. Například Šorel a Šroubek v [45], používají jeden rozmazaný a jeden zašuměný nebo podexponovaný snímek té samé noční scény k nalezení ostrého. Předpokládají, že konvoluční jádro se lokálně příliš nemění a rozmazání aproximují konvolucí v mřížce podobrazů. Ta pak přísluší centrálním pixelům podobrazů a pro zbytek pixelů aproximují PSF bilineární interpolací. Podobný postup zvolili Tallón et al. v [48], používají jeden podexponovaný a jeden přeexponovaný snímek. Přeexponovaný obraz degradovaný pohybem kamery rozdělují na mřížku podobrázků, u kterých jednotlivě odhadují jejich rozostření a následně na podobrázcích provedou neslepou dekonvoluci. V případě, kdy je jádro odhadnuté špatně kvůli nedostatku textury, použijí místo doostřeného podobrázku odpovídající oblast z podexponovaného snímku.

Rozdělení snímku na podobrázky, kde se řeší slepá dekonvoluce s prostorově neproměnným jádrem, je možné efektivně řešit pomocí efficient filter flow algoritmu [17]. Hirsch et al. ji pak kombinují s projective motion path modelem, který rozmazaný obraz uvažuje jako váženou sumu projekcí ostrého. Bayesovsky pak k dekonvoluci obrazu s prostorově proměnným jádrem přistupují Zhang a Wipf v [58].



## Kapitola 2

# Bayesovský přístup ke slepé dekonvoluci

V této kapitole bude nejprve představena variační Bayesova metoda a algoritmus zde používaný k nalezení aposteriorních rozdělání. Následuje popis ELBO (evidence lower bound) a alternativního způsobu odhadu parametrů modelu, reparametrizačního triku pro odhad středních hodnot a nastínění optimalizátoru Vadam pro gradientní sestup. Na základě této teorie je pak odvozen model pro řešení slepé dekonvoluce obrazu.

### 2.1 Variační Bayesova metoda

K popisu systému se skrytými proměnnými je možné použít hierarchické Bayesovské modely. Bayesovská podstata metody vyžaduje jistou znalost systému ke zkonstruování pravděpodobnostního modelu, z něhož bude možné získat dobré výsledky, například předpokládat pozitivitu proměnných nebo požadovat, aby měly řídkou strukturu.

Nechť jsou naměřena data  $\mathbf{D} \in \mathbb{R}^{p \times n}$ , jejichž strukturu chceme objasnit. Označme  $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^r$  vektor skrytých (latentních) proměnných, které data  $\mathbf{D}$  nagenovala.  $\boldsymbol{\theta}$  i  $\mathbf{D}$  budeme dále vnímat jako náhodné proměnné. Předpoklady o struktuře systému je potom možné vyjádřit volbou hustoty pravděpodobnosti  $f(\mathbf{D} | \boldsymbol{\theta})$  a apriorních hustot pravděpodobnosti  $f(\boldsymbol{\theta} | \mathbf{s})$  a  $f(\mathbf{s})$ , kde  $\mathbf{s}$  jsou hyperparametry, tzn. neznámé parametry rozdělání  $\boldsymbol{\theta}$ .  $f(\mathbf{D} | \boldsymbol{\theta})$ ,  $f(\boldsymbol{\theta} | \mathbf{s})$  a  $f(\mathbf{s})$  společně tvoří hierarchický Bayesovský model. Díky součinnému pravidlu platí

$$f(\boldsymbol{\theta}, \mathbf{s}) = f(\boldsymbol{\theta} | \mathbf{s})f(\mathbf{s}).$$

Je-li dána i hustota  $f(\mathbf{D})$ , je možné použít Bayesovy věty.

**Věta 1** (Bayesova). *Nechť jsou dána data  $\mathbf{D}$  a parametr  $\boldsymbol{\theta}$ . Pak platí*

$$f(\boldsymbol{\theta} | \mathbf{D}) = \frac{f(\mathbf{D} | \boldsymbol{\theta})f(\boldsymbol{\theta})}{f(\mathbf{D})}, \quad (2.1)$$

kde  $f(\boldsymbol{\theta} | \mathbf{D})$  nazveme aposteriorní hustotou pravděpodobnosti.

Vezmeme-li  $\boldsymbol{\theta} \equiv (\boldsymbol{\theta}, \mathbf{s})$ , získáme z věty 1 sdružené aposteriorní rozdělání latentních proměnných a jejich hyperparametrů, tj. rozdělání, které bere do úvahy jak apriorní představu o

systemu, tak ona naměřená data. V některých případech (variační Bayesova metoda) znalost  $f(\mathbf{D})$  není nutná. Díky tomu, že pro podmíněnou hustotu pravděpodobnosti platí

$$f(\mathbf{D}) = \int_{\Theta} f(\mathbf{D}, \boldsymbol{\theta}) d\boldsymbol{\theta} = \int_{\Theta} f(\mathbf{D} | \boldsymbol{\theta}) f(\boldsymbol{\theta}) d\boldsymbol{\theta}, \quad (2.2)$$

je  $f(\mathbf{D})$  pouze normalizační konstantou hustoty  $f(\boldsymbol{\theta} | \mathbf{D})$ . Proto v některých aplikacích postačí úměrnost

$$f(\boldsymbol{\theta} | \mathbf{D}) \propto f(\mathbf{D} | \boldsymbol{\theta}) f(\boldsymbol{\theta}). \quad (2.3)$$

Pomocí věty 1 je tedy možné získat sdruženou aposteriorní hustotu parametrů  $\boldsymbol{\theta}$  a  $s$ . Pro odhad latentních proměnných jejich středními hodnotami je potřeba znát jejich marginální rozdělení. Jen výjimečně je možné nalézt odhady analyticky, ve většině reálných případů je nutné sáhnout po některé z aproximačních metod, například po variační Bayesově metodě.

Nejprve zavedme systém nezávislých podmíněných hustot

$$\mathcal{F}_c = \{f(\boldsymbol{\theta} | \mathbf{D}) : f(\boldsymbol{\theta} | \mathbf{D}) = f(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_q | \mathbf{D}) = \prod_{j=1}^q f(\boldsymbol{\theta}_j | \mathbf{D})\}. \quad (2.4)$$

Uvažujeme-li pouze hustoty pravděpodobnosti z tohoto systému, je mnohem snazší najít analytické řešení, neboť k nalezení marginální hustoty  $f(\boldsymbol{\theta}_i)$  pro odhad  $\boldsymbol{\theta}_i$  pomocí střední hodnoty není nutné přintegrovávat sdruženou hustotu  $f(\boldsymbol{\theta})$ . Ve fyzice se tato faktorizace používá pod označením mean field theory, kap. 10 [9]. Tímto je zaveden předpoklad stochastické nezávislosti latentních proměnných.

Dále je výhodné uvažovat konjugované hustoty pravděpodobnosti.

**Definice 1** (Konjugovaný systém hustot). *Mějme systém hustot pravděpodobnosti  $\mathcal{F} = \{f(x | \boldsymbol{\theta})\}$ . Systém  $\mathcal{H} = \{f(\boldsymbol{\theta})\}$  se nazývá konjugovaný k systému  $\mathcal{F}$ , pokud  $\forall f(\boldsymbol{\theta}) \in \mathcal{H}$  a  $\forall f(x | \boldsymbol{\theta}) \in \mathcal{F}$  platí, že  $f(\boldsymbol{\theta} | x) \in \mathcal{H}$ . (kap. 4 v [7])*

Označme  $s$  parametry apriorního rozdělení vektoru  $\boldsymbol{\theta}$  a necht  $T = T(\mathbf{D}, s)$  je postačující statistika.

**Definice 2** (Postačující statistika). *Necht  $X$  je náhodná veličina s rozdělením závislým na parametru  $\boldsymbol{\theta}$ . Funkce  $T(X)$  se nazývá postačující statistika, pokud rozdělení  $X | T(X) = t$  nezávisí na  $\boldsymbol{\theta}$ . (kap. 1 v [7])*

Je-li  $f(\boldsymbol{\theta} | s) \in \mathcal{H}$ , tj. apriorní rozdělení  $\boldsymbol{\theta}$  s parametry  $s$  je konjugované s  $f(\mathbf{D} | \boldsymbol{\theta})$ , pak platí, že

$$f(\boldsymbol{\theta} | T) \propto f(\mathbf{D} | \boldsymbol{\theta}) f(\boldsymbol{\theta} | s),$$

viz. kap. 2 v [43], což velmi připomíná výraz (2.3). Volba konjugovaných hustot zjednodušuje hledání aposteriorního rozdělení v tom smyslu, že funkční tvar aposteriorní hustoty  $f(\boldsymbol{\theta} | T)$  je stejný jako apriorní  $f(\boldsymbol{\theta} | s)$  a mění se pouze jeho parametry. Proto je třeba volit takové funkce  $f(\boldsymbol{\theta} | s)$ , pro které  $L_D(\boldsymbol{\theta}) = f(\mathbf{D} | \boldsymbol{\theta})$ , kde  $L_D$  je věrohodnostní funkce, má stejný tvar. Speciálně k hustotám z exponenciální rodiny, tzn. hustotám tvaru

$$f(\mathbf{D} | \boldsymbol{\theta}) = h(\mathbf{D})c(\boldsymbol{\theta}) \exp \left[ \sum_{k=1}^r q_k(\boldsymbol{\theta}) T_k(\mathbf{D}) \right],$$



není problém najít konjugovanou hustotu s pomocí Neymannova faktorizačního kritéria, viz. kap. 5 v [8]. Například k normálnímu rozdělení s přesností (převrácená hodnota variance) jako hyperparametrem je konjugovaná hustota gama-rozdělení.

Máme-li možné tvary aproximací aposteriorní hustoty, je nutné zvolit takové, které se co nejméně liší od původní hustoty. Odlišnost reálné hustoty od aproximované je měřena pomocí Kullback-Leiblerovy divergence.

**Definice 3** (Kullback-Leiblerova divergence). *Kullback-Leiblerova divergence (KL divergence) funkce  $p(x)$  od funkce  $q(x)$  je definována jako*

$$KL(p(x) \parallel q(x)) = - \int p(x) \ln q(x) dx - \left( - \int p(x) \ln p(x) dx \right) = \int p(x) \ln \frac{p(x)}{q(x)}.$$

Pokud je  $p(x)$  navíc hustota pravděpodobnosti, pak

$$KL(p(x) \parallel q(x)) = \mathbb{E}_{p(x)} \left[ \ln \frac{p(x)}{q(x)} \right], \quad (2.5)$$

kde  $\mathbb{E}_{p(x)} [\cdot]$  představuje střední hodnotu vzhledem k  $p(x)$ . (kap. 3 v [43])

Označme  $\hat{f}(\boldsymbol{\theta} \mid \mathbf{D})$  aproximaci reálné hustoty  $f(\boldsymbol{\theta} \mid \mathbf{D})$ . Potom bude vzdálenost aproximované od původní hustoty měřena jako

$$KL(\hat{f}(\boldsymbol{\theta} \mid \mathbf{D}) \parallel f(\boldsymbol{\theta} \mid \mathbf{D})) = \mathbb{E}_{\hat{f}(\boldsymbol{\theta} \mid \mathbf{D})} \left[ \ln \frac{\hat{f}(\boldsymbol{\theta} \mid \mathbf{D})}{f(\boldsymbol{\theta} \mid \mathbf{D})} \right]. \quad (2.6)$$

Je třeba brát v potaz, že KL divergence není symetrická

$$KL(f(\boldsymbol{\theta} \mid \mathbf{D}) \parallel \hat{f}(\boldsymbol{\theta} \mid \mathbf{D})) \neq KL(\hat{f}(\boldsymbol{\theta} \mid \mathbf{D}) \parallel f(\boldsymbol{\theta} \mid \mathbf{D})).$$

Ve variační Bayesově metodě budeme využívat vztah (2.6), tedy vzdálenost aproximace od reálné hustoty a ne naopak.

Zavedením  $\mathcal{F}_c$  (2.4) a Kullback-Leiblerovy vzdálenosti je položen dostatečný základ pro vyslovení stěžejní věty variační Bayesovy metody.

**Věta 2** (Variační Bayesova metoda). *Nechť je  $f(\boldsymbol{\theta} \mid \mathbf{D})$  aposteriorní hustotou pravděpodobnosti vícerozměrné proměnné  $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_q)^T$ . Nechť dále  $\hat{f}(\boldsymbol{\theta} \mid \mathbf{D})$  označuje aproximaci aposteriorní hustoty nezávislých proměnných  $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_q$ , kterou je možné vyjádřit jako*

$$\hat{f}(\boldsymbol{\theta} \mid \mathbf{D}) = \hat{f}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_q \mid \mathbf{D}) = \prod_{j=1}^q \hat{f}_j(\boldsymbol{\theta}_j \mid \mathbf{D}).$$

Pak je minima Kullback-Leiblerovy divergence

$$\tilde{f}(\boldsymbol{\theta} \mid \mathbf{D}) = \underset{\hat{f}(\cdot)}{\operatorname{argmin}} KL(\hat{f}(\boldsymbol{\theta} \mid \mathbf{D}) \parallel f(\boldsymbol{\theta} \mid \mathbf{D})). \quad (2.7)$$

dosaženo pro

$$\tilde{f}(\boldsymbol{\theta}_i \mid \mathbf{D}) \propto \exp \left[ \mathbb{E}_{\tilde{f}(\boldsymbol{\theta}_{\setminus i} \mid \mathbf{D})} [\ln f(\boldsymbol{\theta}, \mathbf{D})] \right], \quad i = 1, \dots, n, \quad (2.8)$$

kde  $\boldsymbol{\theta}_{\setminus i}$  představuje vektor  $\boldsymbol{\theta}$  bez jeho  $i$ -té složky. (kap. 3 v [43])

Důkaz lze nalézt v [39], kap. 3 v [43].

Věta (2) dává přesný návod, jak postupovat při hledání aproximace.

Hledání aposteriorních rozdělání pomocí VB (variační Bayesovy metody) je v případě, kdy se zvolí konjugovaná rozdělání velmi jednoduché - jejich tvary jsou známé a je potřeba určit pouze hodnoty parametrů. Ty tvoří soustavu lineárních rovnic, která ve většině reálných případů nemá analytické řešení. Lze ho nalézt například pomocí IVB algoritmu (Iterative Variational Bayes) z kap. 3 [43]. Pro jednoduchost je zde uveden pro dva parametry  $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ . Algoritmus pro tento případ sestává ze dvou kroků:

- Určení  $n$ -té iterace aproximované hustoty pravděpodobnosti parametru  $\boldsymbol{\theta}_2$  jako

$$\tilde{f}^{(n)}(\boldsymbol{\theta}_2 | \mathbf{D}) \propto \exp \left[ \int_{\Theta_1} \tilde{f}^{(n-1)}(\boldsymbol{\theta}_1 | \mathbf{D}) \ln f(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \mathbf{D}) \, d\boldsymbol{\theta}_1 \right].$$

- Určení  $n$ -té iterace aproximované hustoty pravděpodobnosti parametru  $\boldsymbol{\theta}_1$  jako

$$\tilde{f}^{(n)}(\boldsymbol{\theta}_1 | \mathbf{D}) \propto \exp \left[ \int_{\Theta_2} \tilde{f}^{(n)}(\boldsymbol{\theta}_2 | \mathbf{D}) \ln f(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \mathbf{D}) \, d\boldsymbol{\theta}_2 \right].$$

Cyklická iterace přes  $n = 1, 2, \dots$ , monotonně snižuje KL divergenci (2.7). Inicializace  $\hat{f}^{(0)}(\boldsymbol{\theta}_1 | \mathbf{D})$  a  $\hat{f}^{(0)}(\boldsymbol{\theta}_2 | \mathbf{D})$  lze zvolit libovolně.

IVB algoritmus se někdy také nazývá EM-like nebo Variational EM algoritmus, protože silně připomíná Expectation-Maximization algoritmus (EM algoritmus) [13]. EM algoritmus sestává ze dvou kroků: v expectation kroku (E-step) je hledáno rozdělání skrytých proměnných a v maximization kroku (M-step) je hledán maximálně věrohodný odhad parametrů z rozdělání nalezených v předchozím kroku. Stejně tak je možné hledat i MAP odhad (maximum aposteriori odhad) [9] daný jako

$$\hat{\boldsymbol{\theta}}_2^{MAP} = \underset{\boldsymbol{\theta}_2}{\operatorname{argmax}} f(\boldsymbol{\theta}_2 | \mathbf{d}). \quad (2.9)$$

Zvolíme-li místo (2.4) systém

$$\mathcal{F}'_c = \{f(\boldsymbol{\theta} | \mathbf{d}) : f(\boldsymbol{\theta} | \mathbf{d}) = f(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 | \mathbf{d}) = f(\boldsymbol{\theta}_1 | \boldsymbol{\theta}_2, \mathbf{d}) \delta_{\hat{\boldsymbol{\theta}}_2}(\boldsymbol{\theta}_2)\}, \quad (2.10)$$

kde  $\delta_{\hat{\boldsymbol{\theta}}_2}(\boldsymbol{\theta}_2) = \delta(\boldsymbol{\theta}_2 - \hat{\boldsymbol{\theta}}_2)$  značí Diracovu  $\delta$ -funkci posunutou do bodu  $\hat{\boldsymbol{\theta}}_2$ , bude odhad  $\boldsymbol{\theta}_2$  pouze bodový. Diracova  $\delta$ -funkce je pro spojitou proměnnou  $x \in \mathbb{X}$  dána jako

$$\int_{\mathbb{X}} \delta_{x_0}(x) g(x) \, dx = g(x_0).$$

Opět pro jednoduchost pro dvě proměnné  $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$  sestává algoritmus ze dvou kroků

- **E-step:** Určení  $n$ -té iterace aproximované hustoty pravděpodobnosti parametru  $\boldsymbol{\theta}_1$  jako

$$\tilde{f}^{(n)}(\boldsymbol{\theta}_1 | \mathbf{D}) \sim f\left(\boldsymbol{\theta}_1 | \hat{\boldsymbol{\theta}}_2^{(n-1)}, \mathbf{D}\right).$$

- **M-step:** Určení  $n$ -té iterace odhadu  $\boldsymbol{\theta}_2$  pomocí

$$\hat{\boldsymbol{\theta}}_2^{(n)} = \underset{\boldsymbol{\theta}_2}{\operatorname{argmax}} \mathbb{E}_{\tilde{f}^{(n)}} [\ln(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \mathbf{D})].$$

## 2.2 ELBO

Výraz (2.6) minimalizovaný VB je možné přepsat následovně

$$KL(\hat{f}(\boldsymbol{\theta}|\mathbf{d})||f(\boldsymbol{\theta}|\mathbf{d})) = \int \hat{f}(\boldsymbol{\theta}|\mathbf{d}) \ln \left[ \frac{\hat{f}(\boldsymbol{\theta}|\mathbf{d})}{f(\boldsymbol{\theta}, \mathbf{d})} \right] d\boldsymbol{\theta} + \ln f(\mathbf{d}) = -\mathcal{L} + \ln f(\mathbf{d}).$$

$f(\mathbf{d})$  se nazývá *evidence* a KL divergence je vždy nezáporná,  $\mathcal{L}$  je tedy spodní hranice na  $f(\mathbf{d})$  a proto se označuje evidence lower bound (ELBO). Minimalizace KL divergence potom odpovídá minimalizaci záporně vzaté

$$\mathcal{L} = - \int \hat{f}(\boldsymbol{\theta}|\mathbf{d}) \ln \left[ \frac{\hat{f}(\boldsymbol{\theta}|\mathbf{d})}{f(\boldsymbol{\theta}, \mathbf{d})} \right] d\boldsymbol{\theta} = -\mathbb{E}_{\hat{f}(\boldsymbol{\theta}|\mathbf{d})} \left[ \ln \frac{\hat{f}(\boldsymbol{\theta}|\mathbf{d})}{f(\boldsymbol{\theta}, \mathbf{d})} \right], \quad (2.11)$$

která může být provedena například pomocí gradientního sestupu (gradient descent), tedy

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_t),$$

kde  $\eta$  je učicí parametr a  $\nabla_{\boldsymbol{\theta}}$  značí gradient vzhledem k  $\boldsymbol{\theta}$ . Tato jednoduchá optimalizace obvykle vyžaduje velký počet iterací, proto byly navrženy optimalizátory, které urychlují cestu ke stabilnímu řešení. V této práci jsou výsledky získány pomocí optimalizátoru Adam (Adaptive Momentum) [24]. Minimalizace pomocí Adama vypadá následovně

$$\begin{aligned} \mathbf{u}_{t+1} &= \beta_1 \mathbf{u}_t + (1 - \beta_1) g(\boldsymbol{\theta}_t), \\ \mathbf{s}_{t+1} &= \beta_2 \mathbf{s}_t + (1 - \beta_2) g(\boldsymbol{\theta}_t)^2, \\ \hat{\mathbf{u}}_{t+1} &= \mathbf{u}_{t+1} (1 - \beta_1^t)^{-1}, \\ \hat{\mathbf{s}}_{t+1} &= \mathbf{s}_{t+1} (1 - \beta_2^t)^{-1}, \\ \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \eta \hat{\mathbf{u}}_{t+1} \left( \sqrt{\hat{\mathbf{s}}_{t+1} + \epsilon} \right)^{-1}, \end{aligned}$$

kde  $g(\boldsymbol{\theta}_t)$  je gradient minimalizované funkce v bodě  $\boldsymbol{\theta}_t$ ,  $\eta$  je učicí parametr a  $\beta_1, \beta_2$  a  $\epsilon$  jsou parametry obvykle nastavené na 0.9, 0.999 a  $10^{-8}$ .

Aby bylo možné tento algoritmus použít, je nutné upravit výraz (2.11) do takového tvaru, který lze derivovat podle  $\boldsymbol{\theta}$ . Jinými slovy, je nutné nalézt analytické vyjádření jednotlivých středních hodnot v  $\mathcal{L}$ , což není vždy možné. Řešení tohoto problému navrhli Kingma a Welling v [25] pro variační autoencodery. Pro aproximaci střední hodnoty funkce  $z(\theta_i)$  vzhledem ke  $f(\theta_i|\mathbf{d})$  je možné použít reparametrizační trik, pokud lze  $\theta_i$  reparametrizovat pomocí náhodné proměnné, kterou jde snadno vzorkovat. Nechť je touto proměnnou  $\boldsymbol{\epsilon}$  s rozdělením  $p(\boldsymbol{\epsilon})$  a reparametrizaci označme  $\theta_i = h(\mathbf{m}, \boldsymbol{\epsilon})$ , kde  $\mathbf{m}$  jsou parametry  $f(\theta_i|\mathbf{d})$ . Pak platí

$$\nabla_{\mathbf{m}} \mathbb{E}_{f(\theta_i|\mathbf{d})} [z(\theta_i)] = \nabla_{\mathbf{m}} \mathbb{E}_{p(\boldsymbol{\epsilon})} [z(h(\mathbf{m}, \boldsymbol{\epsilon}))] \approx \frac{1}{L} \sum_{l=1}^L \nabla_{\mathbf{m}} z(h(\mathbf{m}, \boldsymbol{\epsilon}^l)), \quad (2.12)$$

kde  $\nabla_{\mathbf{m}}$  je gradientní operátor vzhledem k  $\mathbf{m}$  a  $\boldsymbol{\epsilon}^l$  je  $l$ -tý vzorek  $\boldsymbol{\epsilon}$ .

## 2.3 Vadam

Podobně jako reparametrizační trik využívá vzorky z aktuální hodnoty aposteriorního rozdělení, je možné upravit samotný optimalizátor Adam tak, aby určoval gradient v navzorkovaných hodnotách. Tento optimalizátor, nazvaný Variational ADAM (VADAM), představili Kahn et al. v [22]. Odvození se zakládá na předpokladu, že náhodná proměnná  $\boldsymbol{\theta}$  má aproximaci aposteriorního rozdělení z exponenciální rodiny s přirozeným parametrem  $\boldsymbol{\theta}$ . Označíme-li  $\mathcal{L}(\boldsymbol{\eta})$  ELBO (2.11) jako funkci přirozeného parametru  $\boldsymbol{\eta}$  a je-li zobrazení přirozeného parametru na střední hodnotu  $\boldsymbol{\theta}$  bijektivní, pak platí, že

$$\mathcal{I}(\boldsymbol{\eta})^{-1} \nabla_{\boldsymbol{\eta}} \mathcal{L} = \nabla_{\mathbf{m}} \mathcal{L}_*(\mathbf{m}),$$

kde  $\mathcal{I}(\boldsymbol{\eta})$  je informační matice a  $\mathcal{L}_*(\mathbf{m})$  ELBO vyjádřen v  $\mathbf{m}$ . Díky tomuto vztahu je pak podle [23] možné iterovat

$$\boldsymbol{\eta}_{t+1} = \boldsymbol{\eta}_t + \beta_t \nabla_{\mathbf{m}} \mathcal{L}_*(\mathbf{m}_t). \quad (2.13)$$

Pro momentový optimalizátor jako je Adam navrhuji

$$\boldsymbol{\eta}_{t+1} = \boldsymbol{\eta}_t + \frac{\beta_t}{1 - \alpha_t} \nabla_{\mathbf{m}} \mathcal{L}_*(\mathbf{m}_t) + \frac{\alpha_t}{1 - \alpha_t} (\boldsymbol{\eta}_t - \boldsymbol{\eta}_{t-1}), \quad (2.14)$$

kde  $\alpha_t$  a  $\beta_t$  jsou učící parametry.

Předpokládejme, že  $q(\boldsymbol{\theta}|\mathbf{d}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ . Parametry  $\boldsymbol{\eta}$  a  $\mathbf{m}$  jsou pak

$$\begin{aligned} \boldsymbol{\eta}^{(1)} &= \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}, & \boldsymbol{\eta}^{(2)} &= -\frac{1}{2} \boldsymbol{\Sigma}^{-1}, \\ \mathbf{m}^{(1)} &= \mathbb{E}_q[\boldsymbol{\theta}] = \boldsymbol{\mu}, & \mathbf{M}^{(2)} &= \mathbb{E}_q[\boldsymbol{\theta} \boldsymbol{\theta}^T] = \boldsymbol{\mu} \boldsymbol{\mu}^T + \boldsymbol{\Sigma}. \end{aligned}$$

Předpokládejme dále, že apriorní rozdělení  $\boldsymbol{\theta}$  je normální s nulovou střední hodnotou a přesností  $\tau$  pro všechny prvky  $\boldsymbol{\theta}$ . Vezmeme-li pak  $\boldsymbol{\Sigma}$  jako diagonální matici  $\text{diag}(\boldsymbol{\sigma})$  a označíme-li  $\mathbf{s} = \boldsymbol{\sigma}^{-1} + \tau \mathbf{1}$ , kde  $\boldsymbol{\sigma}^{-1}$  je vektor převrácených hodnot  $\boldsymbol{\sigma}$ , dostaneme

$$\begin{aligned} \mathbf{s}_{t+1} &= \left(1 - \frac{\beta_t}{1 - \alpha_t}\right) \mathbf{s}_t + \frac{\beta_t}{1 - \alpha_t} g(\boldsymbol{\theta}^{(t)})^2, \\ \boldsymbol{\mu}_{t+1} &= \boldsymbol{\mu}_t - \frac{\beta_t}{1 - \alpha_t} \left(g(\boldsymbol{\theta}^{(t)}) + \tau \boldsymbol{\mu}_t\right) (\mathbf{s}_{t+1} + \tau \mathbf{1})^{-1} + \\ &\quad + \frac{\alpha_t}{1 - \alpha_t} (\mathbf{s}_t + \tau \mathbf{1}) (\mathbf{s}_{t+1} + \tau \mathbf{1})^{-1} \circ (\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t-1}), \end{aligned}$$

kde  $\circ$  značí Hadamardův produkt,  $g(\cdot)$  je gradient funkce  $-\ln p(\mathbf{d}|\boldsymbol{\theta})$  a  $\boldsymbol{\theta}^{(t)}$  je vzorek z  $\mathcal{N}(\boldsymbol{\mu}_t, (\mathbf{s}_t + \tau \mathbf{1})^{-1})$  (v odvození byla použita aproximace  $\boldsymbol{\Sigma}_{t-1} = \boldsymbol{\Sigma}_t$ ). Odtud je pak odvozen Vadam jako

$$\begin{aligned} \mathbf{u}_{t+1} &= \gamma_1 \mathbf{u}_t + (1 - \gamma_1) \left(\hat{g}(\boldsymbol{\theta}^{(t)}) + \tau \boldsymbol{\mu}_t\right), \\ \mathbf{s}_{t+1} &= \gamma_2 \mathbf{s}_t + (1 - \gamma_2) \hat{g}(\boldsymbol{\theta}^{(t)})^2, \\ \hat{\mathbf{u}}_{t+1} &= \mathbf{u}_{t+1} \left(1 - \gamma_1^t\right)^{-1}, \\ \hat{\mathbf{s}}_{t+1} &= \mathbf{s}_{t+1} \left(1 - \gamma_2^t\right)^{-1}, \\ \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \alpha \hat{\mathbf{u}}_{t+1} \left(\sqrt{\hat{\mathbf{s}}_{t+1} + \tau}\right)^{-1}, \end{aligned}$$

kde  $\gamma_1, \gamma_2$  jsou konstanty a  $\alpha$  je učící koeficient.

## 2.4 Model rozmazaného obrazu

Jak bylo zmíněno v předchozí části, k nalezení odhadů obrazu a PSF pomocí VB se obvykle volí hustoty pravděpodobnosti z konjugovaného systému hustot. Díky této volbě potom stačí iterativně zpřesňovat hodnoty parametrů aposteriorních rozdělání například pomocí IVB algoritmu. Tato volba je ale velmi omezující, například v případě prostorově proměnného konvolučního jádra se nedá očekávat, že hledání odhadů bude odpovídat řešení soustavy lineárních rovnic. V této kapitole budou proto srovnávány dva přístupy: VB s IVB algoritmem a optimalizace ELBO pomocí gradientního sestupu. Druhý zmíněný postup není tolik restriktivní vůči volbě rozdělání a nalézt gradient ELBO vzhledem k jednotlivým proměnným je při použití programovacího jazyka s automatickou diferenciací (v této práci se používá jazyk julia) snadné.

Nejprve bude představen model rozmazaného snímku s hustotami z konjugovaného systému, aby bylo možné slepou dekonvoluci řešit oběma přístupy. Následuje odvození aproximací aposteriorních hustot pomocí VB a ELBO. Přesné tvary hustot pravděpodobnosti používané v této kapitole je možné nalézt například v dodatku A v [8].

## 2.5 Volba apriorních rozdělání

Uvažujme maticový zápis modelu (1.2) z první kapitoly

$$\mathbf{d} = \mathbf{K}\mathbf{x} + \mathbf{n} = \mathbf{X}\mathbf{k} + \mathbf{n}, \quad (2.15)$$

kde  $\mathbf{d} \in \mathbb{R}_+^{np}$  je vektorizovaný rozmazaný obraz  $\mathbf{D}$ ,  $\mathbf{K} \in \langle 0, 1 \rangle^{np \times np}$ , resp.  $\mathbf{X} \in \mathbb{R}_+^{np \times s^2}$  je dvojitá bloková Toeplitzova matice zkonstruovaná z  $\tilde{\mathbf{K}}$ , resp.  $\tilde{\mathbf{X}}$ , přičemž v případě  $\mathbf{X}$  se nejedná o kompletní dvojitou blokovou Toeplitzovu matici, ona sama ani její bloky nejsou čtvercové – poslední sloupce čtvercových matic jsou "useknuté", protože rozměr PSF je menší než rozměr obrazu.  $\mathbf{x} \in \mathbb{R}_+^{np}$ , resp.  $\mathbf{k} \in \langle 0, 1 \rangle^{s^2}$  je vektorizovaný ostrý obraz  $\tilde{\mathbf{X}}$ , resp. vektorizovaná matice rozostření  $\tilde{\mathbf{K}}$  a  $\mathbf{N}$  je vektorizovaný šum. Vektorizací je myšleno seskládání sloupců matice jeden po druhém do vektoru, tzn. je-li  $\mathbf{D}$  matice rozměru  $n \times p$  a její sloupce jsou  $(\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_p)$ , kde  $\mathbf{d}_i$  jsou sloupcové vektory, pak je  $\mathbf{d} = (\mathbf{d}_1^T, \mathbf{d}_2^T, \dots, \mathbf{d}_p^T)^T$ .

Šum byl předpokládán gaussovský s variancí  $\omega^{-1}$ , tedy rozdělání dat je

$$\mathbf{d} | \mathbf{k}, \mathbf{x}, \omega \sim \mathcal{N}(\mathbf{K}\mathbf{x}, \omega^{-1} \mathbb{I}_{np}),$$

kde  $\mathbb{I}_{np}$  je jednotková matice o rozměru  $np \times np$ . Apriorní rozdělání hyperparametru  $\omega$  je gama-rozdělání s pevně zvolenými parametry  $\gamma_0, \eta_0$ , tzn.

$$\omega \sim \mathcal{G}(\gamma_0, \eta_0).$$

Obrázek by měl být po částech hladký, proto bylo jako apriorní rozdělání obrázku zvoleno ARD apriorno (sekce 1.4.3) na jeho derivacích, jako např. v [46], [26] nebo [50]. Derivace jsou zde brány jako rozdíly sousedních pixelů - horizontálních a vertikálních sousedů  $x_{i,j} + x_{i+1,j}$  a  $x_{i,j} - x_{i,j+1}$ , které jsou napočítány pomocí součinu obrázku s maticemi  $\nabla_h$  a  $\nabla_v$ . Vezmeme-li  $\nabla_h \mathbf{x}$  a  $\nabla_v \mathbf{x}$  jako dva obrázky s normálním rozděláním a nulovou střední hodnotou a přesností  $\tau_{xh}$  a  $\tau_{xv}$ , můžeme rozdělání  $\mathbf{x}$ , podobně jako v [50] nebo v [42], napsat následovně

$$\mathbf{x} | \tau_x \sim \mathcal{N}\left(\mathbf{0}, \left(\nabla^T \text{diag}(\tau_x) \nabla\right)^{-1}\right),$$

kde  $\nabla$  je matice, která vznikla spojením  $\nabla_h$  a  $\nabla_v$  pod sebe,  $\text{diag}(\boldsymbol{\tau}_x)$  značí diagonální matici s  $\boldsymbol{\tau}_x$  na diagonále a  $\boldsymbol{\tau}_x \in \mathbb{R}_+^{n(p-1)+p(n-1)}$  je vektor přesností jednotlivých rozdílů, tedy  $\boldsymbol{\tau}_x = (\boldsymbol{\tau}_{xh}^T, \boldsymbol{\tau}_{xv}^T)^T$ . Pro samotnou přesnost, jak bylo zmíněno v sekci 1.4.3, bylo zvoleno gama-rozdělení

$$\tau_{xi} \sim \mathcal{G}(\alpha_{x0}, \beta_{x0}),$$

kde  $i = 1, 2, \dots, n(p-1) + p(n-1)$  a  $\alpha_{x0}, \beta_{x0}$  jsou parametry stejné pro všechny prvky  $\boldsymbol{\tau}_x$ , pevně zvolené.

Normální rozdělení je také apriorním rozdělením  $\mathbf{k}$

$$\mathbf{k} | \tau_k \sim \mathcal{N}(\mathbf{0}, \tau_k \mathbb{I}_m),$$

kde je přesnost  $\tau_k$  pro všechny prvky  $\mathbf{k}$  stejná a její apriorní rozdělení je také gama

$$\tau_k \sim \mathcal{G}(\alpha_{k0}, \beta_{k0}),$$

kde  $\alpha_{k0}$  a  $\beta_{k0}$  jsou pevně zvolené parametry.

## 2.6 Odvození VB

Jak bylo zmíněno v sekci 2.1, aproximace aposteriorních rozdělení odvozené pomocí VB mají tvar

$$\tilde{f}(\boldsymbol{\theta}_i | \mathbf{D}) \propto \exp \left[ \mathbb{E}_{\tilde{f}(\boldsymbol{\theta}_i | \mathbf{D})} [\ln f(\boldsymbol{\theta}, \mathbf{D})] \right], \quad i = 1, \dots, n, \quad (2.16)$$

(vzorec (2.8)). Náhodné veličiny  $\boldsymbol{\theta}$  ve vzorci (2.16) jsou v navrhovaném modelu  $\mathbf{x}, \mathbf{k}, \boldsymbol{\tau}_x, \tau_k$  a  $\omega$ . Sdružená hustota  $f(\boldsymbol{\theta}, \mathbf{D})$  je dána jako součin

$$f(\mathbf{d} | \mathbf{x}, \mathbf{k}, \omega) f(\mathbf{x} | \boldsymbol{\tau}_x) f(\mathbf{k} | \tau_k) f(\boldsymbol{\tau}_x) f(\tau_k) f(\omega),$$

kde  $f(\cdot)$  značí (apriorní) hustoty pravděpodobnosti rozdělení navržených v předchozí sekci. Označme  $L = \ln f(\boldsymbol{\theta}, \mathbf{D})$ , pak

$$\begin{aligned} L = & \frac{np}{2} \ln \omega - \frac{\omega}{2} (\mathbf{d} - \mathbf{K}\mathbf{x})^T (\mathbf{d} - \mathbf{K}\mathbf{x}) + \frac{m}{2} \ln \tau_k - \frac{\tau_k}{2} \mathbf{k}^T \mathbf{k} + \frac{1}{2} |\nabla^T \text{diag}(\boldsymbol{\tau}_x) \nabla| - \\ & - \frac{1}{2} \mathbf{x}^T \left( \nabla^T \text{diag}(\boldsymbol{\tau}_x) \nabla \right) \mathbf{x} + \sum_{j=1}^r ((\alpha_{x0} - 1) \ln \tau_{xj} - \beta_{x0} \tau_{xj}) + (\alpha_{k0} - 1) \ln \tau_k - \\ & - \beta_{k0} \tau_k + (\gamma_0 - 1) \ln \omega + \eta_0 \omega, \end{aligned} \quad (2.17)$$

kde  $m = s^2$  je počet prvků  $\mathbf{k}$  a  $r = (n-1)p + (p-1)n$  je počet prvků  $\boldsymbol{\tau}_x$ .

Pro nalezení aproximace aposteriorního rozdělení  $\mathbf{x}$  jsou nutné pouze následující členy výrazu (2.17)

$$\begin{aligned} L_x = & -\frac{\omega}{2} (\mathbf{d} - \mathbf{K}\mathbf{x})^T (\mathbf{d} - \mathbf{K}\mathbf{x}) - \frac{1}{2} \mathbf{x}^T \left( \nabla^T \text{diag}(\boldsymbol{\tau}_x) \nabla \right) \mathbf{x} = \\ = & -\frac{\omega}{2} \left( \mathbf{d}^T \mathbf{d} - \mathbf{d}^T \mathbf{K}\mathbf{x} - \mathbf{x}^T \mathbf{K}^T \mathbf{d} + \mathbf{x}^T \mathbf{K}^T \mathbf{K}\mathbf{x} \right) - \frac{1}{2} \mathbf{x}^T \left( \nabla^T \text{diag}(\boldsymbol{\tau}_x) \nabla \right) \mathbf{x}. \end{aligned}$$

Z tvaru aposteriorního rozdělení (2.16) a z členů  $L_x$  obsahujících  $\mathbf{x}$  je vidět, že aproximace jeho aposteriorního rozdělení bude normální rozdělení (z konjugovanosti)

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x).$$

Je tedy nutné nalézt takové hodnoty  $\boldsymbol{\mu}_x$  a  $\boldsymbol{\Sigma}_x$ , aby bylo možné zapsat členy  $L_x$ , které obsahují  $\mathbf{x}$  následujícím způsobem

$$-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_x)^T \boldsymbol{\Sigma}_x^{-1} (\mathbf{x} - \boldsymbol{\mu}_x) = -\frac{1}{2} \left( \mathbf{x}^T \boldsymbol{\Sigma}_x^{-1} \mathbf{x} - \mathbf{x}^T \boldsymbol{\Sigma}_x^{-1} \boldsymbol{\mu}_x - \boldsymbol{\mu}_x^T \boldsymbol{\Sigma}_x^{-1} \mathbf{x} + \boldsymbol{\mu}_x^T \boldsymbol{\Sigma}_x^{-1} \boldsymbol{\mu}_x \right),$$

až na konstantu. Aby aproximace odpovídala vztahu (2.16), je ještě třeba vzít  $L_x$  jako střední hodnotu vzhledem k aproximacím aposteriorních rozdělání všech veličin kromě  $\mathbf{x}$ . Pak dostaneme, že

$$\begin{aligned} \boldsymbol{\Sigma}_x &= \left( \hat{\omega} \widehat{\mathbf{K}^T \mathbf{K}} + \nabla^T \text{diag}(\hat{\boldsymbol{\tau}}_x) \nabla \right)^{-1}, \\ \boldsymbol{\mu}_x &= \hat{\omega} \boldsymbol{\Sigma}_x \widehat{\mathbf{K}^T} \mathbf{d}. \end{aligned}$$

kde stříška označuje střední hodnoty náhodných veličin vzhledem k aproximacím jejich aposteriorních rozdělání.

Přepíšeme-li  $\mathbf{K}\mathbf{x}$  na  $\mathbf{X}\mathbf{k}$ , je část (2.17) obsahující  $\mathbf{k}$  následující

$$\begin{aligned} L_k &= -\frac{\omega}{2} (\mathbf{d} - \mathbf{X}\mathbf{k})^T (\mathbf{d} - \mathbf{X}\mathbf{k}) - \frac{\tau_k}{2} \mathbf{k}^T \mathbf{k} = \\ &= -\frac{\omega}{2} \left( \mathbf{d}^T \mathbf{d} - \mathbf{d}^T \mathbf{X}\mathbf{k} - \mathbf{k}^T \mathbf{X}^T \mathbf{d} + \mathbf{k}^T \mathbf{X}^T \mathbf{X}\mathbf{k} \right) - \frac{\tau_k}{2} \mathbf{k}^T \mathbf{k}, \end{aligned}$$

a stejným způsobem jako v předchozím případě lze odvodit, že

$$\begin{aligned} \mathbf{k} &\sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \\ \boldsymbol{\Sigma}_k &= \left( \hat{\omega} \widehat{\mathbf{X}^T \mathbf{X}} + \hat{\tau}_k \mathbb{I} \right)^{-1}, \\ \boldsymbol{\mu}_k &= \hat{\omega} \boldsymbol{\Sigma}_k \widehat{\mathbf{X}^T} \mathbf{d}. \end{aligned}$$

Jako aproximace aposteriorních rozdělání přesností  $\boldsymbol{\tau}_x$ ,  $\tau_x$  a  $\omega$  bylo zvoleno degenerované rozdělání (Diracova  $\delta$ -funkce). Odhad náhodných proměnných  $\boldsymbol{\tau}_x$ ,  $\tau_k$  a  $\omega$  je potom MAP odhad (2.9).

MAP odhad parametru  $\boldsymbol{\tau}_x$  je argumentem maxima části (2.17) obsahující  $\boldsymbol{\tau}_x$ , tedy

$$\begin{aligned} L_{\boldsymbol{\tau}_x} &= \frac{1}{2} \ln |\nabla^T \text{diag}(\boldsymbol{\tau}_x) \nabla| - \frac{1}{2} \mathbf{x}^T \left( \nabla^T \text{diag}(\boldsymbol{\tau}_x) \nabla \right) \mathbf{x} + \\ &\quad + \sum_{j=1}^r ((\alpha_{x0} - 1) \ln \tau_{xj} - \beta_{x0} \tau_{xj}). \end{aligned}$$

Podobně jako Tzikas et al. v [50] aproximujeme determinant  $|\nabla^T \text{diag}(\boldsymbol{\tau}_x) \nabla|$  jako  $|\nabla^T \nabla| |\text{diag}(\boldsymbol{\tau}_x)|$ , takže výraz, který bude minimalizován vzhledem k  $\boldsymbol{\tau}_x$ , je

$$\begin{aligned} \tilde{L}_{\boldsymbol{\tau}_x} &= \frac{1}{2} \ln |\text{diag}(\boldsymbol{\tau}_x)| - \frac{1}{2} \mathbf{x}^T \left( \nabla^T \text{diag}(\boldsymbol{\tau}_x) \nabla \right) \mathbf{x} + \\ &\quad + \sum_{j=1}^r ((\alpha_{x0} - 1) \ln \tau_{xj} - \beta_{x0} \tau_{xj}). \end{aligned}$$

K nalezení maxima zderivujeme  $\tilde{L}_{\boldsymbol{\tau}_x}$  a položíme ho rovno nule. Nejdříve odvodíme derivaci kvadratického členu z apriorního rozdělání obrazu

$$\begin{aligned} \frac{\partial}{\partial \tau_{xj}} \left( \mathbf{x}^T \left( \nabla^T \text{diag}(\boldsymbol{\tau}_x) \nabla \right) \mathbf{x} \right) &= \left( \frac{\partial}{\partial \tau_{xj}} \left( \mathbf{x}^T \nabla^T \text{diag}(\boldsymbol{\tau}_x) \right) \right) \nabla \mathbf{x} + \\ &\quad + \mathbf{x}^T \nabla^T \text{diag}(\boldsymbol{\tau}_x) \left( \frac{\partial}{\partial \tau_{xj}} \nabla \mathbf{x} \right) = \mathbf{x}^T \nabla^T \mathcal{I}_j \nabla \mathbf{x}, \end{aligned}$$

kde

$$\mathcal{I}_j = \frac{\partial}{\partial \tau_{x_j}} \text{diag}(\boldsymbol{\tau}_x),$$

je matice nul s jedničkou na  $j$ -té pozici. Takže dohromady

$$\begin{aligned} \frac{\partial \tilde{L}_{\tau_x}}{\partial \tau_{x_j}} &= \frac{1}{2} \frac{1}{\tau_{x_j}} - \frac{1}{2} \mathbf{x}^T \nabla^T \mathcal{I}_j \nabla \mathbf{x} + (\alpha_{x_0} - 1) \frac{1}{\tau_{x_j}} - \beta_{x_0} \stackrel{!}{=} 0, \\ \tau_{x_j} \left( \frac{1}{2} \mathbf{x}^T \nabla^T \mathcal{I}_j \nabla \mathbf{x} + \beta_{x_0} \right) &= \frac{1}{2} + \alpha_{x_0} - 1, \end{aligned}$$

a MAP odhad  $\boldsymbol{\tau}_x$  je

$$\hat{\tau}_{x_j} = \left( \frac{1}{2} \mathbb{E}_{\tilde{f}(x|d)} \left[ \mathbf{x}^T \nabla^T \mathcal{I}_j \nabla \mathbf{x} \right] + \beta_{x_0} \right)^{-1} \left( \alpha_{x_0} - \frac{1}{2} \right).$$

Pro  $\tau_k$  dostaneme

$$L_{\tau_k} = \frac{m}{2} \ln \tau_k - \frac{\tau_k}{2} \mathbf{k}^T \mathbf{k} + (\alpha_{k_0} - 1) \ln \tau_k - \beta_{k_0} \tau_k,$$

Odtud

$$\begin{aligned} \frac{\partial L_{\tau_k}}{\partial \tau_k} &= \frac{m}{2} \frac{1}{\tau_k} - \frac{1}{2} \mathbf{k}^T \mathbf{k} + (\alpha_{k_0} - 1) \frac{1}{\tau_k} - \beta_{k_0} \stackrel{!}{=} 0, \\ \tau_k \left( \frac{1}{2} \mathbf{k}^T \mathbf{k} + \beta_{k_0} \right) &= \frac{m}{2} + \alpha_{k_0} - 1, \end{aligned}$$

a bodový odhad  $\tau_k$  je tedy

$$\hat{\tau}_k = \left( \frac{1}{2} \widehat{\mathbf{k}^T \mathbf{k}} + \beta_{k_0} \right)^{-1} \left( \frac{m}{2} + \alpha_{k_0} - 1 \right).$$

A analogicky pro  $\omega$  máme

$$L_\omega = \frac{np}{2} \ln \omega - \frac{\omega}{2} (\mathbf{d} - \mathbf{K}\mathbf{x})^T (\mathbf{d} - \mathbf{K}\mathbf{x}) + (\gamma_0 - 1) \ln \omega - \eta_0 \omega \quad (2.18)$$

a derivace je následující

$$\frac{\partial L_\omega}{\partial \omega} = \left( \frac{np}{2} + \gamma_0 - 1 \right) \frac{1}{\omega} - \left( \frac{1}{2} (\mathbf{d} - \mathbf{K}\mathbf{x})^T (\mathbf{d} - \mathbf{K}\mathbf{x}) + \eta_0 \right) \stackrel{!}{=} 0.$$

MAP odhad  $\omega$  je tedy

$$\hat{\omega} = \left( \frac{np}{2} + \gamma_0 - 1 \right) \left( \frac{1}{2} \mathbb{E}_{\tilde{f}(x,k|d)} \left[ (\mathbf{d} - \mathbf{K}\mathbf{x})^T (\mathbf{d} - \mathbf{K}\mathbf{x}) \right] + \eta_0 \right)^{-1}.$$

V případě, kdy by se hledalo aposteriorní rozdělení pro přesnosti  $\omega, \tau_k, \boldsymbol{\tau}_x$ , z konjugovanosti zvolených apriorních rozdělení by se jednalo o gama-rozdělení. To dosahuje maxima v modu, pokud je čítelel ve zlomku v odhadu kladný, a pokud tomu tak není, aposteriorní rozdělení se limitně blíží maximu v nule. MAP odhady jsou tedy platné pouze v případě kladného čitatele, což musí být zohledněno při nastavování fixních parametrů apriorního rozdělení.



## 2.7 Odvození ELBO

Evidence lower bound (ELBO) je dáno vztahem 2.11, tj.

$$\mathcal{L} = -\mathbb{E}_{\tilde{f}(\boldsymbol{\theta}|\mathbf{d})} \left[ \ln \frac{\tilde{f}(\boldsymbol{\theta}|\mathbf{d})}{f(\boldsymbol{\theta}, \mathbf{d})} \right] = -\mathbb{E}_{\tilde{f}(\boldsymbol{\theta}|\mathbf{d})} [\ln \tilde{f}(\boldsymbol{\theta}|\mathbf{d})] + \mathbb{E}_{\tilde{f}(\boldsymbol{\theta}|\mathbf{d})} [\ln f(\boldsymbol{\theta}, \mathbf{d})], \quad (2.19)$$

takže k vyjádření  $\mathcal{L}$  je třeba najít střední hodnoty logaritmu jednotlivých apriorních hustot, hustoty šumu a aproximací aposteriorních rozdělání vzhledem k těmto aproximacím. Aproximace aposteriorních rozdělání zvolíme tak jako v předchozím případě, tedy

$$\begin{aligned} \mathbf{x} &\sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x), \\ \mathbf{k} &\sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \end{aligned} \quad (2.20)$$

a degenerovaná rozdělání pro  $\boldsymbol{\tau}_x$ ,  $\tau_k$  a  $\omega$  s nenulovými hodnotami v  $\hat{\boldsymbol{\tau}}_x$ ,  $\hat{\tau}_k$  a  $\hat{\omega}$ . Střední hodnota logaritmu hustoty šumu je

$$\begin{aligned} \mathbb{E}_{\tilde{f}} [\ln f(\mathbf{d}|\mathbf{k}, \mathbf{x})] &= -\frac{np}{2} \ln(2\pi) + \frac{np}{2} \ln \hat{\omega} - \\ &- \frac{\hat{\omega}}{2} \text{tr} \left( \mathbf{d}^T \mathbf{d} - \mathbf{d}^T \widehat{\mathbf{K}} \hat{\mathbf{x}} - \hat{\mathbf{x}}^T \widehat{\mathbf{K}}^T \mathbf{d} + \mathbb{E}_{\tilde{f}} [\mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x}] \right), \end{aligned}$$

kde stříška značí střední hodnotu vzhledem k aproximaci sdruženého aposteriorního rozdělání. Využijeme-li aproximaci  $|\boldsymbol{\nabla}^T \text{diag}(\boldsymbol{\tau}_x) \boldsymbol{\nabla}| \approx |\boldsymbol{\nabla}^T \boldsymbol{\nabla}| |\text{diag}(\boldsymbol{\tau}_x)|$ , pak pro střední hodnotu apriorního rozdělání ostrého obrazu dostaneme

$$\mathbb{E}_{\tilde{f}} [\ln f(\mathbf{x}|\boldsymbol{\tau}_x)] = -\frac{np}{2} \ln(2\pi) + \frac{1}{2} \ln |\boldsymbol{\nabla}^T \boldsymbol{\nabla}| + \frac{1}{2} \sum_{j=1}^r \ln \hat{\tau}_{xj} - \frac{1}{2} \mathbb{E}_{\tilde{f}} [\mathbf{x}^T \boldsymbol{\nabla}^T \text{diag}(\boldsymbol{\tau}_x) \boldsymbol{\nabla} \mathbf{x}].$$

Střední hodnota z logaritmu aposteriorního rozdělání  $\mathbf{x}$  je

$$\mathbb{E}_{\tilde{f}} [\ln \tilde{f}(\mathbf{x})] = C_X - \frac{1}{2} \ln |\boldsymbol{\Sigma}_x|,$$

kde  $C_X$  je konstanta vzhledem k parametrům rozdělání. Pro apriorní rozdělání PSF

$$\mathbb{E}_{\tilde{f}} [\ln f(\mathbf{k}|\tau_k)] = -\frac{m}{2} \ln(2\pi) + \frac{m}{2} \ln \hat{\tau}_k - \frac{\hat{\tau}_k}{2} \widehat{\mathbf{k}}^T \mathbf{k},$$

a pro jeho aposteriorní rozdělání dostaneme

$$\mathbb{E}_{\tilde{f}} [\ln \tilde{f}(\mathbf{k})] = C_K - \frac{1}{2} \ln |\boldsymbol{\Sigma}_k|,$$

kde  $C_k$  je opět konstanta. Dále pro apriorní rozdělání přesností

$$\begin{aligned} \mathbb{E}_{\tilde{f}} [\ln f(\tau_x)] &= C_x + (\alpha_{x0} - 1) \ln \hat{\boldsymbol{\tau}}_x - \beta_{x0} \hat{\boldsymbol{\tau}}_x, \\ \mathbb{E}_{\tilde{f}} [\ln f(\tau_k)] &= C_k + (\alpha_{k0} - 1) \ln \hat{\tau}_k - \beta_{k0} \hat{\tau}_k, \\ \mathbb{E}_{\tilde{f}} [\ln f(\omega)] &= C_\omega + (\gamma_0 - 1) \ln \hat{\omega} - \eta_0 \hat{\omega}. \end{aligned}$$

kde  $C_x$ ,  $C_k$  a  $C_\omega$  jsou normalizační konstanty. Střední hodnoty jejich aposteriorních rozdělání budou brány jako konstanty.

## 2.8 Tvary momentů

Aby bylo možné řešit obě úlohy, je nutné nalézt vyjádření všech středních hodnot pomocí parametrů aposteriorních rozdělání. První momenty jsou jednoduše

$$\hat{\mathbf{x}} = \boldsymbol{\mu}_x, \quad \hat{\mathbf{k}} = \boldsymbol{\mu}_k,$$

a druhé momenty mají také jednoduchý tvar

$$\widehat{\mathbf{x}^T \mathbf{x}} = \text{tr} \left( \boldsymbol{\Sigma}_x + \boldsymbol{\mu}_x \boldsymbol{\mu}_x^T \right), \quad \widehat{\mathbf{k}^T \mathbf{k}} = \text{tr} \left( \boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T \right).$$

Dále se v odvození vyskytují střední hodnoty

$$\mathbf{x}^T \nabla^T \mathcal{I}_j \nabla \mathbf{x}, \quad \mathbf{X}^T \mathbf{X}, \quad \mathbf{K}^T \mathbf{K}, \quad \mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x}.$$

K určení první a poslední je třeba znát střední hodnotu  $\mathbf{x}^T \mathbf{A} \mathbf{x}$ , kde  $\mathbf{A}$  je libovolná čtvercová matice. Tato střední hodnota je dána výrazem

$$\text{tr} \left( \left( \boldsymbol{\Sigma}_x + \boldsymbol{\mu}_x \boldsymbol{\mu}_x^T \right) \mathbf{A} \right), \quad (2.21)$$

takže

$$\mathbb{E}_{\tilde{f}} \left[ \mathbf{x}^T \nabla^T \mathcal{I}_j \nabla \mathbf{x} \right] = \text{tr} \left( \left( \boldsymbol{\Sigma}_x + \boldsymbol{\mu}_x \boldsymbol{\mu}_x^T \right) \nabla^T \mathcal{I}_j \nabla \right).$$

Nalezení střední hodnoty  $\mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x}$  ještě vyžaduje určit střední hodnotu  $\mathbf{K}^T \mathbf{K}$ . Matici  $\mathbf{K}$  je možné zkonstruovat pomocí indexové matice  $\mathbf{V}$ , která rozkopíruje prvky  $\mathbf{k}$  na odpovídající pozice v  $\mathbf{K}$ . Pokud je vztah  $\mathbf{V}$  a  $\mathbf{K}$  dán jako  $\text{vec}(\mathbf{K}) = \mathbf{V} \mathbf{k}$ , kde  $\text{vec}(\mathbf{K})$  je  $\mathbf{K}$  vektorizováno po sloupcích, pak lze  $i$ -tý sloupec  $\mathbf{K}$  vyjádřit jako součin matice  $\mathbf{V}_i = \mathbf{V}_{((i-1)m+1:im,:)}$ , kde zápis  $((i-1)m+1 : im, :)$  značí řádky  $(i-1)m+1$  až  $im$  a všechny sloupce, a vektoru  $\mathbf{k}$ .  $(i, j)$ -tý prvek  $\mathbf{K}^T \mathbf{K}$  je pak dán jako skalární součin  $i$ -tého a  $j$ -tého sloupce  $\mathbf{K}$ , tedy

$$\left[ \mathbf{K}^T \mathbf{K} \right]_{(i,j)} = (\mathbf{V}_i \mathbf{k})^T \mathbf{V}_j \mathbf{k},$$

takže střední hodnota tohoto prvku je podle (2.21) rovna

$$\mathbb{E}_{\tilde{f}} \left[ \mathbf{k}^T \mathbf{V}_i^T \mathbf{V}_j \mathbf{k} \right] = \text{tr} \left( \left( \boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T \right) \mathbf{V}_i^T \mathbf{V}_j \right).$$

Díky tomu, že střední hodnota matice je matice středních hodnot jednotlivých prvků, je pomocí tohoto vztahu možné zkonstruovat střední hodnotu  $\mathbf{K}^T \mathbf{K}$  a analogicky i  $\mathbf{X}^T \mathbf{X}$ , jehož indexovou matici označíme  $\mathbf{W}$ .

## Kapitola 3

# Srovnání metod na testovacích datech

Jako testovací snímek byl zvolen výřez z Leny [2], na obrázku 3.2. Ten byl poškozen třemi způsoby: gaussovským rozostřením, rozmazáním lineárním pohybem a špatným zaostřením, všechny jsou vykresleny na obrázku 3.1. Matice PSF je normalizovaná tak, aby součet jejích prvků byl roven jedné. Následně byl k obrázku přidán gaussovský šum s variancí na několika úrovních SNR (signal-to-noise ratio), které je definováno jako

$$\text{SNR} = 10 \log_{10} \left( \frac{\sigma_D^2}{\sigma_N^2} \right),$$

kde  $\sigma_D^2$  je variance obrazu a  $\sigma_N^2$  variance šumu a je udáváno v decibelech (dB).

V následujících sekcích bude ukázáno řešení slepé dekonvoluce pomocí VB a kombinace ELBO a VB, kde ELBO bude použito pro odhad aposteriorního rozdělení konvolučního jádra. Kromě přímého odhadu ostrého obrazu algoritmem, tzn. konečná hodnota  $\mu_x$ , bude provedena i neslepá dekonvoluce rozmazaného obrazu a odhadnuté PSF. Postup při neslepé dekonvoluci byl následující: k diagonále  $\widehat{\mathbf{K}}$  byla přičtena malá hodnota  $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$  a byla provedena operace

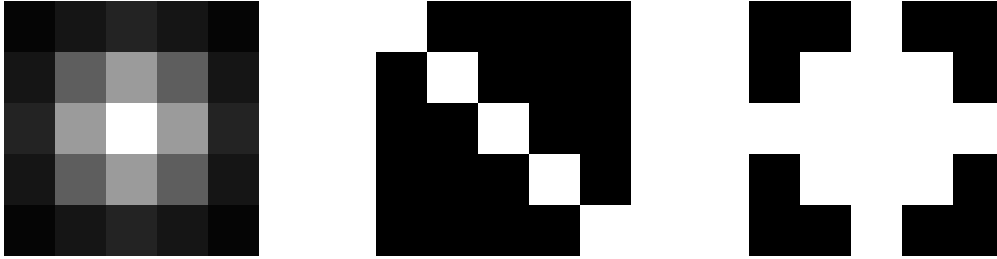
$$(\widehat{\mathbf{K}} + \alpha \mathbb{I})^{-1} \mathbf{d}. \quad (3.1)$$

Poté bylo určeno PSNR (peak signal-to-noise ratio) pro jednotlivé hodnoty  $\alpha$  a na jejich základě zvolena nejlepší rekonstrukce, která je zde pak prezentována. PSNR je dáno následovně

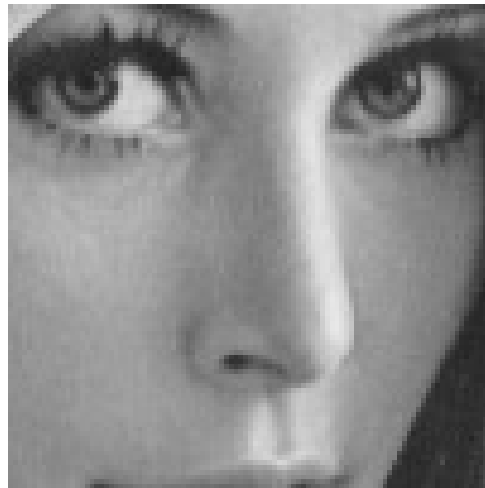
$$\text{PSNR} = 10 \log_{10} \left( \frac{(\max(\mathbf{I}) - \min(\mathbf{I}))^2}{\text{MSE}(\mathbf{I}, \hat{\mathbf{I}})} \right),$$

kde  $\mathbf{I}$  značí ostrý obraz,  $\hat{\mathbf{I}}$  jeho odhad,  $\text{MSE}(\mathbf{I}, \hat{\mathbf{I}})$  jejich střední kvadratickou odchylku a je stejně jako SNR udáváno v decibelech (dB).

V tabulce 3.1 jsou vypsány hodnoty PSNR pro výřez z Leny poškozený rozmazáním všemi třemi PSF a přidáním šumu se SNR 20dB, 30dB, 40dB a 50dB. Rozmazání lineárním pohybem způsobuje podle PSNR větší poškození - váha pixelů, které přímo nesousedí s rozmazávaným pixelem, je větší než u druhých dvou.



Obrázek 3.1: PSF použité v testovacích datech. Zleva gaussovské rozostření, rozostření lineárním pohybem a špatné zaostření.



Obrázek 3.2: Výřez z Leny [2] použitý v testovacích datech.

PSF\SNR	20dB	30dB	40dB	50dB
gauss	26.04054	26.84347	26.92108	26.93129
pohyb	22.81888	23.19967	23.24462	23.24661
zaostření	25.50598	26.15277	26.21793	26.22545

Tabulka 3.1: Tabulka hodnot PSNR poškozených obrazů pro tři druhy PSF a 4 úrovně SNR.

### 3.1 Odhad pomocí VB

Soustava rovnic, jejíž řešení je potřeba nalézt, sestává z odvozených momentů ze sekce 2.6, které jsou zde pro přehlednost znovu vypsány

$$\begin{aligned}
\boldsymbol{\Sigma}_x &= \left( \hat{\omega} \widehat{\mathbf{K}}^T \mathbf{K} + \nabla^T \text{diag}(\hat{\boldsymbol{\tau}}_x) \nabla \right)^{-1}, \\
\boldsymbol{\mu}_x &= \hat{\omega} \boldsymbol{\Sigma}_x \widehat{\mathbf{K}}^T \mathbf{d}, \\
\boldsymbol{\Sigma}_k &= \left( \hat{\omega} \widehat{\mathbf{X}}^T \mathbf{X} + \hat{\boldsymbol{\tau}}_k \mathbb{I} \right)^{-1}, \\
\boldsymbol{\mu}_k &= \hat{\omega} \boldsymbol{\Sigma}_k \widehat{\mathbf{X}}^T \mathbf{d}, \\
\hat{\boldsymbol{\tau}}_{xj} &= \left( \frac{1}{2} \mathbb{E}_{\hat{f}(x|d)} \left[ \mathbf{x}^T \nabla^T \mathcal{I}_j \nabla \mathbf{x} \right] + \beta_{x0} \right)^{-1} \left( \alpha_{x0} - \frac{1}{2} \right), \\
\hat{\boldsymbol{\tau}}_k &= \left( \frac{1}{2} \widehat{\mathbf{k}}^T \mathbf{k} + \beta_{k0} \right)^{-1} \left( \frac{m}{2} + \alpha_{k0} - 1 \right), \\
\hat{\omega} &= \left( \frac{np}{2} + \gamma_0 - 1 \right) \left( \frac{1}{2} \left( \mathbf{d}^T \mathbf{d} - 2 \mathbf{d}^T \widehat{\mathbf{K}} \hat{\mathbf{x}} + \mathbb{E}_{\hat{f}(x,k|d)} \left[ \mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x} \right] \right) + \eta_0 \right)^{-1}.
\end{aligned}$$

a z přirozených momentů ze sekce 2.8, tj.

$$\begin{aligned}
\hat{\mathbf{x}} &= \boldsymbol{\mu}_x, & \hat{\mathbf{k}} &= \boldsymbol{\mu}_k, & \widehat{\mathbf{k}}^T \mathbf{k} &= \text{tr} \left( \boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T \right), \\
\widehat{\mathbf{K}}_j &= \mathbf{V}_j \boldsymbol{\mu}_k, & \widehat{\mathbf{X}}_i &= \mathbf{W}_i \boldsymbol{\mu}_x, & & \forall i \in \hat{n}p, j \in \hat{m}
\end{aligned}$$

kde  $\widehat{\mathbf{K}}_j$ , resp.  $\widehat{\mathbf{X}}_i$  značí  $j$ -tý sloupec matice  $\widehat{\mathbf{K}}$ , resp.  $i$ -tý sloupec matice  $\widehat{\mathbf{X}}$  a  $\mathbf{V}_j$  a  $\mathbf{W}_i$  jsou indexové matice popsané předchozí sekci. Dále

$$\begin{aligned}
\mathbb{E}_{\hat{f}} \left[ \mathbf{x}^T \nabla^T \mathcal{I}_j \nabla \mathbf{x} \right] &= \text{tr} \left( \left( \boldsymbol{\Sigma}_x + \boldsymbol{\mu}_x \boldsymbol{\mu}_x^T \right) \nabla^T \mathcal{I}_j \nabla \right), \\
\mathbb{E}_{\hat{f}} \left[ \mathbf{K}^T \mathbf{K} \right]_{(i,j)} &= \text{tr} \left( \left( \boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T \right) \mathbf{V}_i^T \mathbf{V}_j \right), \quad \forall i, j \in \hat{n}p \\
\mathbb{E}_{\hat{f}} \left[ \mathbf{X}^T \mathbf{X} \right]_{(i,j)} &= \text{tr} \left( \left( \boldsymbol{\Sigma}_x + \boldsymbol{\mu}_x \boldsymbol{\mu}_x^T \right) \mathbf{W}_i^T \mathbf{W}_j \right), \quad \forall i, j \in \hat{m} \\
\mathbb{E}_{\hat{f}} \left[ \mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x} \right] &= \text{tr} \left( \left( \boldsymbol{\Sigma}_x + \boldsymbol{\mu}_x \boldsymbol{\mu}_x^T \right) \mathbb{E}_{\hat{f}} \left[ \mathbf{K}^T \mathbf{K} \right] \right).
\end{aligned}$$

Výpočet  $\boldsymbol{\Sigma}_x$  obnáší inverzi matice o rozměru  $np \times np$ , která je obvykle špatně podmíněná, proto bude brána jako diagonální, stejně jako v [26]. Kovarianční matice  $\mathbf{k}$  je výrazně menších rozměrů, takže by její aproximace nemusela být nutná. Přínos ponechání plné kovariance PSF bude srovnán dvěma verzemi IVB algoritmu. Aby ale diagonalizací  $\boldsymbol{\Sigma}_x$  nedošlo ke ztrátě informace při výpočtu  $\boldsymbol{\mu}_x$ , je  $\boldsymbol{\mu}_x$  hledáno jako řešení

$$\boldsymbol{\Sigma}_x^{-1} \boldsymbol{\mu}_x = \hat{\omega} \widehat{\mathbf{X}}^T \mathbf{d}, \tag{3.2}$$

pomocí konjugovaných gradientů a analogický postup bude použit i v případě diagonální kovariance  $\boldsymbol{\Sigma}_k$ , tedy

$$\boldsymbol{\Sigma}_k^{-1} \boldsymbol{\mu}_k = \hat{\omega} \widehat{\mathbf{K}}^T \mathbf{d}. \tag{3.3}$$

Verzi s plnou kovariancí PSF nazveme IVB-full a je zde rozepsán jako algoritmus 1. Operátor  $\text{diag}(\cdot)$  v algoritmu 1 vrací vektor na diagonále v případě, kdy je vstupem matice, a vytváří

---

**Algoritmus 1:** IVB-full algoritmus
 

---

Inicializace podle tabulky 3.2,  $\boldsymbol{\mu}_x = \mathbf{d}$ ,  $\boldsymbol{\mu}_k = \boldsymbol{\delta}$ ;  
**while** není dosaženo konvergence **do**  
 | Spočti  $\boldsymbol{\Sigma}_x = \text{diag} \left( \left( \text{diag} \left( \widehat{\boldsymbol{\omega}} \widehat{\mathbf{K}}^T \widehat{\mathbf{K}} + \nabla^T \text{diag}(\widehat{\boldsymbol{\tau}}_x) \nabla \right) \right)^{-1} \right)$ ;  
 | Spočti  $\boldsymbol{\mu}_x$  řešením (3.2) konjugovanými gradienty;  
 | Přepočti  $\widehat{\boldsymbol{x}}$ ,  $\widehat{\mathbf{X}}^T \widehat{\mathbf{X}}$ ,  $\mathbb{E}_{\widehat{f}} \left[ \mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x} \right]$  a  $\mathbb{E}_{\widehat{f}} \left[ \mathbf{x}^T \nabla^T \mathcal{L}_j \nabla \mathbf{x} \right]$ ;  
 | **for** všechna  $i \in \widehat{n}_p$  **do**  
 | |  $\widehat{\tau}_{xj} = \left( \frac{1}{2} \mathbb{E}_{\widehat{f}(x|d)} \left[ \mathbf{x}^T \nabla^T \mathcal{L}_j \nabla \mathbf{x} \right] + \beta_{x0} \right)^{-1} \left( \alpha_{x0} - \frac{1}{2} \right)$ ;  
 | **end**  
 | Spočti  $\widehat{\tau}_k = \left( \frac{1}{2} \widehat{\mathbf{k}}^T \widehat{\mathbf{k}} + \beta_{k0} \right)^{-1} \left( \frac{m}{2} + \alpha_{k0} - 1 \right)$ ;  
 | Spočti  $\widehat{\omega} = \left( \frac{np}{2} + \gamma_0 - 1 \right) \left( \frac{1}{2} \left( \mathbf{d}^T \mathbf{d} - 2 \mathbf{d}^T \widehat{\mathbf{K}} \widehat{\boldsymbol{x}} + \mathbb{E}_{\widehat{f}(x,k|d)} \left[ \mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x} \right] \right) + \eta_0 \right)^{-1}$ ;  
 | Spočti  $\boldsymbol{\Sigma}_k = \left( \widehat{\omega} \widehat{\mathbf{X}}^T \widehat{\mathbf{X}} + \widehat{\tau}_k \mathbf{I} \right)^{-1}$ ;  
 | Spočti  $\boldsymbol{\mu}_k = \widehat{\omega} \boldsymbol{\Sigma}_k \widehat{\mathbf{X}}^T \mathbf{d}$ , záporné hodnoty nastav na nulu a poděl  $\sum_i \mu_{ki}$ ;  
 | Přepočti  $\widehat{\mathbf{K}}$ ,  $\widehat{\mathbf{K}}^T \widehat{\mathbf{K}}$  a  $\widehat{\mathbf{k}}^T \widehat{\mathbf{k}}$ ;  
**end**

---

diagonální matici v případě, kdy je vstupem vektor. Hodnoty  $\boldsymbol{\mu}_k$ , které jsou odhadnuté jako záporné jsou v každé iteraci nastaveny na nulu a celý vektor je normalizován tak, aby součet jeho prvků byl roven jedné.

Počáteční hodnoty parametrů jsou vypsány v tabulce 3.2. Ne pro všechny druhy poškození jsou optimální, ale byly zvoleny tak, aby ve všech případech algoritmus našel rozumný výsledek. Kovarianční matice byly inicializovány jako diagonální. Počáteční odhad ostrého obrazu byl nastaven na poškozený obraz  $\mathbf{d}$  a jako počáteční odhad PSF je brána Diracova  $\delta$ -funkce, tedy PSF, která má nenulový prvek pouze ve svém středu. Velikost nosiče (supportu) je nastavena stejná jako u originální PSF, tedy  $5 \times 5$ , výsledky pro počáteční nastavení  $7 \times 7$  budou prezentovány později.

$\boldsymbol{\Sigma}_x$	$\boldsymbol{\Sigma}_k$	$\boldsymbol{\tau}_x$	$\boldsymbol{\tau}_k$	$\alpha_{X0}$	$\beta_{X0}$	$\alpha_{k0}$	$\beta_{k0}$	$\omega$	$\gamma_0$	$\eta_0$
$10^{-5} \mathbf{I}$	$10^{-2} \mathbf{I}$	$10^3 \mathbf{1}$	$10^2$	$10^0$	$10^{-5}$	$10^0$	$10^{-2}$	$10^2$	$10^{-6}$	$10^{-6}$

Tabulka 3.2: Počáteční hodnoty skrytých proměnných a parametrů modelu.  $\mathbf{I}$  značí jednotkovou matici a  $\mathbf{1}$  vektor jedniček.

Prvním testovaným rozmazáním je rozmazání gaussovské. Na obrázku 3.3 jsou v prvním ukázané výsledky pro případ se SNR (Signal to Noise Ratio) 40dB. Přímý výstup z algoritmu má ostřejší hrany než původní obraz a přechody mezi pixely nejsou tak hladké - působí mírně rozkostičkovane. Neslepá dekonvoluce byla provedena s přičtením hodnoty 0.3 k diagonále  $\widehat{\mathbf{K}}$ . Výsledek neslepé dekonvoluce není tak ostrý jako původní obraz, nicméně je ostřejší než obraz rozmazaný. Odhad PSF ukázaný na obrázku je velmi podobný původní PSF. Algoritmus konverguje do cca 100 kroků. PSNR obou odhadů společně s odhady pro další hladiny šumu (20dB, 30dB a 50dB) jsou srovnána v tabulce 3.3.

Dále byl algoritmus testován na obrázku rozmazaném lineárním pohybem se šumem 40dB, výsledky jsou na obrázku (3.3) v druhém řádku. V tomto případě dosahuje lepší rekonstrukce přímý výstup z algoritmu než následná neslepá dekonvoluce, která proběhla s koeficientem 0.9. Odhadnuté jádro je velmi podobné původnímu, ale pixely sousedící s nenulovými pixely nejsou odhadnuty jako nulové. Konvergence je pomalejší, potřebuje cca 300 kroků, dle PSNR rozmazaných obrazů je tento poškozenější než obraz rozmazaný gaussovským rozmazáním (viz tabulka 3.1). PSNR je opět uvedeno v tabulce 3.3 společně s koeficienty pro neslepou dekonvoluci.

Poslední zde zmíněnou variantou je špatné zaostření se SNR 40dB, odhady jsou na obrázku 3.3 ve třetím řádku. Jsou velmi podobné výsledkům nalezeným při slepé dekonvoluci s gaussovským jádrem, odhadnutá PSF více připomíná gaussovskou PSF, i když obrys původní PSF je patrný. I v tomto případě je přímý odhad mírně rozkostičkováný a výsledek neslepé dekonvoluce (s koeficientem 0.3) je pouze lehce rozostřený. Konvergence dosáhne do sta kroků. PSNR odhadů společně s PSNR pro další hodnoty SNR je uvedeno v tabulce 3.3.

Porovnáme-li PSNR v tabulce 3.1 s PSNR poškozených obrazů (tabulka 3.1), vidíme, že s výjimkou gaussovské PSF při SNR 20dB, kde přímý odhad i neslepá dekonvoluce mají PSNR nižší než poškozený obraz, a rozmazání lineárním pohybem při SNR 20dB, kde má výstup z neslepé dekonvoluce nižší PSNR než poškozený obraz, jsou všechny odhady z hlediska PSNR lepší než poškozený obraz.

Algoritmus využívající pouze diagonální kovarianční matici PSF nazveme IVB-diag a je zde rozepsaný jako algoritmus 2. Využívá stejné počáteční hodnoty jako algoritmus IVB-full, které jsou vypsány v tabulce 3.2, z  $\Sigma_k$  je inicializována pouze diagonála. Počáteční odhady obrazu a PSF jsou také stejné, tedy rozmazaný obraz a Diracova  $\delta$ -funkce.

---

#### Algoritmus 2: IVB-diag algoritmus

---

Inicializace podle tabulky 3.2,  $\mu_x = \mathbf{d}$ ,  $\mu_k = \delta$ ;

**while** není dosaženo konvergence **do**

Spočti  $\Sigma_x = \text{diag} \left( \left( \text{diag} \left( \hat{\omega} \widehat{\mathbf{K}}^T \widehat{\mathbf{K}} + \nabla^T \text{diag}(\hat{\tau}_x) \nabla \right) \right)^{-1} \right)$ ;

Spočti  $\mu_x$  řešením (3.2) konjugovanými gradienty;

Přepečti  $\hat{\mathbf{x}}$ ,  $\widehat{\mathbf{X}}^T \widehat{\mathbf{X}}$ ,  $\mathbb{E}_{\hat{f}} \left[ \mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x} \right]$  a  $\mathbb{E}_{\hat{f}} \left[ \mathbf{x}^T \nabla^T \mathcal{I}_j \nabla \mathbf{x} \right]$ ;

**for** všechna  $i \in \widehat{n}_p$  **do**

$\hat{\tau}_{xj} = \left( \frac{1}{2} \mathbb{E}_{\hat{f}(x|d)} \left[ \mathbf{x}^T \nabla^T \mathcal{I}_j \nabla \mathbf{x} \right] + \beta_{x0} \right)^{-1} \left( \alpha_{x0} - \frac{1}{2} \right)$ ;

**end**

Spočti  $\hat{\tau}_k = \left( \frac{1}{2} \widehat{\mathbf{k}}^T \widehat{\mathbf{k}} + \beta_{k0} \right)^{-1} \left( \frac{m}{2} + \alpha_{k0} - 1 \right)$ ;

Spočti  $\hat{\omega} = \left( \frac{np}{2} + \gamma_0 - 1 \right) \left( \frac{1}{2} \left( \mathbf{d}^T \mathbf{d} - 2 \mathbf{d}^T \widehat{\mathbf{K}} \hat{\mathbf{x}} + \mathbb{E}_{\hat{f}(x,k|d)} \left[ \mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x} \right] \right) + \eta_0 \right)^{-1}$ ;

Spočti  $\Sigma_k = \text{diag} \left( \left( \text{diag} \left( \hat{\omega} \widehat{\mathbf{X}}^T \widehat{\mathbf{X}} + \hat{\tau}_k \mathbb{I} \right) \right)^{-1} \right)$ ;

Spočti  $\mu_k$  řešením (3.3) pomocí konjugovaných gradientů, záporné hodnoty nastav na nulu a poděl  $\sum_i \mu_{ki}$ ;

Přepečti  $\widehat{\mathbf{K}}$ ,  $\widehat{\mathbf{K}}^T \widehat{\mathbf{K}}$  a  $\widehat{\mathbf{k}}^T \widehat{\mathbf{k}}$ ;

**end**

---

Opět bude testovaný nejprve na datech s gaussovským rozostřením. První řádek na obrázku 3.4 ukazuje, že odhad přímo z algoritmu je opět lehce rozkostičkováný. Vypadá tmavší než



Obrázek 3.3: Výsledky pro IVB-full algoritmus (algoritmus (1)) a SNR 40dB. V prvním řádku jsou uvedeny výsledky pro gaussovské rozostření, v druhém pro rozmazání lineárním pohybem a v třetím pro špatné zaostření. V každém řádku je na levé straně nahoře ukázáno původní jádro, pod ním je jeho odhad. Následuje zleva původní čistý obraz, rozmazaný obraz s přidáním šumem, přímý výstup z algoritmu a výsledek neslepé dekonvoluce.

původní obraz, ale odhad je škálovaný tak, aby maximální hodnota v obraze byla rovna jedné. Výstup ze slepé dekonvoluce s parametrem 0.3 není tak ostrý jako původní obraz, ale je ostřejší než obraz poškozený. Jádro není tak podobné původnímu jako bylo v případě s plnou kovariancí PSF a není úplně symetrické, nicméně stále silně připomíná jádro původní. Konvergence je v tomto případě stejně rychlá jako u plné kovariance s gaussovským rozmazáním. PSNR je společně s výsledky algoritmu IVB-full uvedeno v tabulce 3.3.

Při rozostření lineárním pohybem pro SNR 40dB je i v případě IVB-diag algoritmu přímý odhad ostrého obrazu původnímu obrazu bližší než ten, který vznikl neslepou dekonvolucí s koeficientem 0.9, jak je vidět v druhém řádku obrázku 3.4. Odhad jádra je výrazně horší než ten nalezený algoritmem IVB-full, konvergence je ale dosaženo rychleji - do sta kroků. PSNR nalezeného ostrého obrazu i neslepé dekonvoluce je uvedeno v tabulce 3.3.

Slepá dekonvoluce špatně zaostřeného obrazu za použití algoritmu IVB-diag má řešení velmi podobné případu s gaussovskou PSF. Jak je vidět z obrázku 3.4, odhad PSF pro tyto dva způsoby rozmazání je velmi podobný, stejně jako mírně rozkostičkovaný odhad ostrého obrazu. Slepá dekonvoluce byla provedena s koeficientem 0.3 a její výstup je ostřejší než poškozený obraz, nicméně není tak ostrý jako obraz původní. Řešení bylo stabilní po 100 iteracích a PSNR



je vypsáno v tabulce 3.3.

Porovnáme-li PSNR odhadů z algoritmu IVB-diag s PSNR poškozených obrazů (tabulka 3.1), ukazuje se, že při gaussovské PSF a SNR 20dB má poškozený obraz vyšší PSNR, stejně jako tomu bylo u algoritmu IVB-full. Pro tuto PSF navíc i přímý odhad při zbylých úrovních SNR nedosahuje PSNR poškozeného obrazu, neslepá dekonvoluce ho však výrazně převyšuje. PSNR přímého odhadu při SNR 20dB a rozmazání lineárním pohybem je opět nižší než PSNR rozmazaného obrazu. Přímý odhad při špatném zaostření a SNR 20 dB také nedosahuje PSNR poškozeného obrazu.



Obrázek 3.4: Výsledky pro IVB-diag algoritmus (algoritmus (2)) a SNR 40dB. V prvním řádku jsou uvedeny výsledky pro gaussovské rozostření, v druhém pro rozmazání lineárním pohybem a v třetím pro špatné zaostření. V každém řádku je na levé straně nahoře ukázáno původní jádro, pod ním je jeho odhad. Následuje zleva původní čistý obraz, rozmazaný obraz s přidáním šumem, přímý výstup z algoritmu a výsledek neslepé dekonvoluce.

Algorismus	PSF	SNR	PSNR $\mu_X$	PSNR dek.	koef.dek.
IVB-full	gauss	20	25.98328	25.93823	0.9
IVB-full	gauss	30	28.22432	29.91891	0.5
IVB-full	gauss	40	29.73543	32.58748	0.3
IVB-full	gauss	50	30.16888	36.51204	0.1
IVB-full	pohyb	20	25.05988	22.60312	0.9
IVB-full	pohyb	30	28.90743	24.08415	0.9
IVB-full	pohyb	40	34.5378	24.27308	0.9
IVB-full	pohyb	50	37.04772	24.29002	0.9
IVB-full	zaostření	20	25.75749	25.42289	0.9
IVB-full	zaostření	30	27.77769	28.7179	0.5
IVB-full	zaostření	40	29.2549	30.14618	0.3
IVB-full	zaostření	50	29.43944	30.56685	0.3
IVB-diag	gauss	20	25.81821	25.67324	0.9
IVB-diag	gauss	30	26.37031	30.25197	0.5
IVB-diag	gauss	40	26.48819	33.04796	0.3
IVB-diag	gauss	50	26.00951	33.76192	0.3
IVB-diag	pohyb	20	24.62589	22.32752	0.9
IVB-diag	pohyb	30	27.76075	23.98974	0.9
IVB-diag	pohyb	40	29.10601	24.2149	0.9
IVB-diag	pohyb	50	29.3045	24.22383	0.9
IVB-diag	zaostření	20	25.81607	25.2112	0.9
IVB-diag	zaostření	30	27.34136	28.8589	0.5
IVB-diag	zaostření	40	27.91142	30.54753	0.3
IVB-diag	zaostření	50	28.10716	30.93815	0.3

Tabulka 3.3: Tabulka hodnot PSNR výsledků algoritmů IVB-full a IVB-diag. PSNR  $\mu_X$  značí PSNR přímého výstupu z algoritmu, koef. dek. je koeficient použitý pro nesleponou dekonvoluci a PSNR dek. PSNR obrazu nalezeného nesleponou dekonvolucí. SNR i PSNR je uvedeno v decibelech.

## 3.2 Odhad pomocí ELBO

Při maximalizaci ELBO se derivuje výraz

$$\begin{aligned}
\mathcal{L} = & -\frac{np}{2} \ln(2\pi) + \frac{np}{2} \ln \hat{\omega} - \frac{\omega}{2} \left( \mathbf{d}^T \mathbf{d} - \mathbf{d}^T \widehat{\mathbf{K}} \hat{\mathbf{x}} - \hat{\mathbf{x}}^T \widehat{\mathbf{K}}^T \mathbf{d} + \mathbb{E}_{\tilde{f}(x,k)} \left[ \mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x} \right] \right) - \\
& -\frac{np}{2} \ln(2\pi) + \frac{1}{2} \ln |\nabla^T \nabla| + \frac{1}{2} \sum_{j=1}^r \ln \hat{\tau}_{xj} - \frac{1}{2} \mathbb{E}_{\tilde{f}} \left[ \mathbf{x}^T \nabla^T \text{diag}(\boldsymbol{\tau}_x) \nabla \mathbf{x} \right] - \\
& -\frac{m}{2} \ln(2\pi) + \frac{m}{2} \ln \hat{\tau}_k - \frac{\hat{\tau}_k}{2} \widehat{\mathbf{k}}^T \mathbf{k} + C_x + (\alpha_{x0} - 1) \ln \hat{\boldsymbol{\tau}}_x - \beta_{x0} \hat{\boldsymbol{\tau}}_x + \\
& + C_k + (\alpha_{k0} - 1) \ln \hat{\tau}_k - \beta_{k0} \hat{\tau}_k + C_\omega + (\gamma_0 - 1) \ln \hat{\omega} - \eta_0 \hat{\omega} - \\
& - C_X + \frac{1}{2} \ln |\boldsymbol{\Sigma}_x| - C_K + \frac{1}{2} \ln |\boldsymbol{\Sigma}_k|. \tag{3.4}
\end{aligned}$$

Motivací pro vyzkoušení tohoto přístupu je prostorově proměnné konvoluční jádro, proto budou v IVB algoritmu nahrazeny odhadem přes ELBO pouze kroky odhadující aposteriorní parametry jádra. Z výrazu (3.4) bude proto potřebná pouze část tyto parametry obsahující, tedy

$$\mathcal{L}_k = \frac{\omega}{2} \left( \mathbf{d}^T \mathbf{d} - 2\mathbf{d}^T \widehat{\mathbf{K}} \hat{\mathbf{x}} + \mathbb{E}_{\tilde{f}(x,k)} \left[ \mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x} \right] \right) - \frac{\hat{\tau}_k}{2} \widehat{\mathbf{k}}^T \widehat{\mathbf{k}} + \frac{1}{2} \ln |\boldsymbol{\Sigma}_k|. \quad (3.5)$$

Nahradíme-li střední hodnoty výrazy ze sekce 2.8, je možné  $\mathcal{L}_k$  derivovat podle  $\boldsymbol{\mu}_k$  a  $\boldsymbol{\Sigma}_k$ . Dalším problémem, který by mohl nastat v případě komplikovaného rozdělení prostorově proměnné PSF, je analyticky nevyjádřitelná střední hodnota nějakého výrazu.  $\mathbf{k}$  bude proto v části (3.5) odpovídající  $\ln p(\mathbf{d}|\mathbf{k}, \mathbf{x}, \omega)$  aproximováno reparametrizačním trikem [25] vysvětleným v předchozí kapitole, takže

$$\mathbf{k} = \boldsymbol{\mu}_k + \mathbf{S}_k \boldsymbol{\epsilon},$$

kde  $\boldsymbol{\Sigma}_k = \mathbf{S}_k \mathbf{S}_k^T$  a  $\boldsymbol{\epsilon}$  má normální rozdělení s nulovou střední hodnotou a jednotkovou kovarianční maticí. Střední hodnota  $\widehat{\mathbf{K}}$  nebude počítána z  $\boldsymbol{\mu}_k$ , ale z  $\mathbf{k}_{rep}(\boldsymbol{\mu}_k, \mathbf{S}_k) = \boldsymbol{\mu}_k + \mathbf{S}_k \bar{\boldsymbol{\epsilon}}$ , kde  $\bar{\boldsymbol{\epsilon}}$  je vzorek  $\boldsymbol{\epsilon}$ , (bude označeno jako  $\mathbf{K}_{rep}(\boldsymbol{\mu}_k, \mathbf{S}_k)$ ) a dále

$$\mathbb{E}_{\tilde{f}(x,k)} \left[ \mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x} \right] \approx \mathbf{k}_{rep}(\boldsymbol{\mu}_k, \mathbf{S}_k)^T \widehat{\mathbf{X}}^T \widehat{\mathbf{X}} \mathbf{k}_{rep}(\boldsymbol{\mu}_k, \mathbf{S}_k).$$

Výraz (3.5) bude mít tvar

$$\begin{aligned} \mathcal{L}_{krep} = \frac{\omega}{2} \left( \mathbf{d}^T \mathbf{d} - 2\mathbf{d}^T \mathbf{K}_{rep}(\boldsymbol{\mu}_k, \mathbf{S}_k) \hat{\mathbf{x}} + \mathbf{k}_{rep}(\boldsymbol{\mu}_k, \mathbf{S}_k)^T \widehat{\mathbf{X}}^T \widehat{\mathbf{X}} \mathbf{k}_{rep}(\boldsymbol{\mu}_k, \mathbf{S}_k) \right) - \\ - \frac{\hat{\tau}_k}{2} \widehat{\mathbf{k}}^T \widehat{\mathbf{k}} + \frac{1}{2} \ln |\boldsymbol{\Sigma}_k|. \end{aligned} \quad (3.6)$$

Tento algoritmus nazveme ELBO a opět budou testovány dvě varianty, jedna s plnou kovarianční maticí  $\boldsymbol{\Sigma}_k = \mathbf{S}_k \mathbf{S}_k^T$  - ELBO-full - a druhá s diagonální  $\boldsymbol{\Sigma}_k = \mathbf{S}_k \mathbf{S}_k^T$  - ELBO-diag. Místo  $\boldsymbol{\Sigma}_k$  bude odhadována  $\mathbf{S}_k$ , takže kovarianční matice bude pozitivně-semidefinitní a symetrická. Oba dva jsou zde rozepsány jako algoritmus 3, v diagonálním případě je pouze  $\mathbf{S}_k$  brána jako diagonální matice, pro kterou jsou odhadovány pouze členy na diagonále.

Počáteční hodnoty byly zvoleny stejné jako pro IVB algoritmus, které jsou vypsány v tabulce 3.2. Počáteční hodnota  $\mathbf{S}_k$  je nastavena na  $10^{-1}$ , takže inicializace  $\boldsymbol{\Sigma}_k$  je diagonální matice s  $10^{-2}$  na diagonále. Počáteční odhad obrazu je  $\mathbf{d}$  a počáteční odhad PSF je Diracova  $\delta$ -funkce.

Algoritmus bude opět testován na datech se třemi PSF. Jako první budou prezentovány výsledky slepé dekonvoluce při použití odhadu přes ELBO pro plnou kovarianční matici  $\boldsymbol{\Sigma}_k$ , výsledky při SNR 40dB jsou na obrázku (3.5). Všechny odhady jsou výrazně horší než ty získané pomocí VB. V případě gaussovské PSF je v odhadu vidět náznak původní struktury, pro zbývající PSF to však neplatí. Ostrý obraz u gaussovského rozostření trochu více připomíná obraz původní, nicméně i tak jsou odhady velmi špatné. Následná neslepá dekonvoluce působí výrazně lépe, protože se provádí na rozmazaném obraze, který je mnohem bližší původnímu ostrému než přímé odhady z algoritmu. Pro gaussovskou PSF vypadá rekonstrukce docela dobře, obraz je jen mírně rozostřený ve srovnání s původním. U výstupu z neslepé dekonvoluce obrazu rozmazaného lineárním pohybem je opět vidět, že je obraz stále rozmazaný ve stejném směru. Neslepá dekonvoluce u špatného zaostření vrací trochu rozostřenější obraz než ta u gaussovského, ale výsledek je lepší než rozmazaný obraz. Ve všech případech algoritmus dosáhne stabilního řešení do sta iterací. V tabulce 3.4 jsou pro srovnání uvedeny PSNR pro všechny varianty rozmazání a různé hladinami šumu.

---

**Algoritmus 3: ELBO algoritmus**

---

Inicializace podle tabulky 3.2,  $\boldsymbol{\mu}_x = \mathbf{d}$ ,  $\boldsymbol{\mu}_k = \boldsymbol{\delta}$ ;  
**while** není dosaženo konvergence **do**  
    Spočti  $\boldsymbol{\Sigma}_x = \text{diag} \left( \left( \text{diag} \left( \widehat{\boldsymbol{\omega}} \widehat{\mathbf{K}}^T \widehat{\mathbf{K}} + \nabla^T \text{diag}(\widehat{\boldsymbol{\tau}}_x) \nabla \right) \right)^{-1} \right)$ ;  
    Spočti  $\boldsymbol{\mu}_x$  řešením (3.2) konjugovanými gradienty;  
    Přepočti  $\widehat{\boldsymbol{x}}$ ,  $\widehat{\mathbf{X}}^T \widehat{\mathbf{X}}$ ,  $\mathbb{E}_{\widehat{f}} \left[ \mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x} \right]$  a  $\mathbb{E}_{\widehat{f}} \left[ \mathbf{x}^T \nabla^T \mathcal{L}_j \nabla \mathbf{x} \right]$ ;  
    **for** všechna  $i \in \widehat{np}$  **do**  
         $\widehat{\tau}_{xj} = \left( \frac{1}{2} \mathbb{E}_{\widehat{f}(x|d)} \left[ \mathbf{x}^T \nabla^T \mathcal{L}_j \nabla \mathbf{x} \right] + \beta_{x0} \right)^{-1} \left( \alpha_{x0} - \frac{1}{2} \right)$ ;  
    **end**  
    Spočti  $\widehat{\tau}_k = \left( \frac{1}{2} \widehat{\mathbf{k}}^T \widehat{\mathbf{k}} + \beta_{k0} \right)^{-1} \left( \frac{m}{2} + \alpha_{k0} - 1 \right)$ ;  
    Spočti  $\widehat{\boldsymbol{\omega}} = \left( \frac{np}{2} + \gamma_0 - 1 \right) \left( \frac{1}{2} \left( \mathbf{d}^T \mathbf{d} - 2 \mathbf{d}^T \widehat{\mathbf{K}} \widehat{\boldsymbol{x}} + \mathbb{E}_{\widehat{f}(x,k|d)} \left[ \mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x} \right] \right) + \eta_0 \right)^{-1}$ ;  
    Nastav optimalizátor Adam na krok  $\boldsymbol{\eta} = 0.001$ ;  
    **for** všechna  $i \in \{1, \dots, 1000\}$  **do**  
        Vezmi jeden vzorek  $\bar{\boldsymbol{\epsilon}}$  z  $\mathcal{N}(\mathbf{0}, \mathbb{I})$ ;  
        Spočti gradienty (3.6) vzhledem k  $\boldsymbol{\mu}_k$  a  $\mathbf{S}_k$ ;  
        Přepočti  $\boldsymbol{\mu}_k$  a  $\mathbf{S}_k$  pomocí optimalizátoru Adam;  
    **end**  
    Záporné hodnoty  $\boldsymbol{\mu}_k$  nastav na nulu a poděl  $\sum_i \mu_{ki}$ ;  
    Přepočti  $\widehat{\mathbf{K}}$ ,  $\widehat{\mathbf{K}}^T \widehat{\mathbf{K}}$  a  $\widehat{\mathbf{k}}^T \widehat{\mathbf{k}}$ ;  
**end**

---

V tomto případě pouze PSNR neslepých dekonvolucí při SNR větším než 20dB převyšují PSNR poškozených obrazů.

Varianta použití ELBO s diagonální kovarianční maticí, tedy algoritmus ELBO-diag, dospěla k výrazně lepším výsledkům než ta s plnou kovariancí na  $\mathbf{k}$ . Odhady obrazu a konvolučních jader jsou na obrázku 3.6. Při dekonvoluci obrazu s gaussovským rozostřením a SNR 40 bylo nalezeno jádro velmi podobné původnímu, působí symetričtěji než jádro nalezené algoritmem IVB-diag. Odhad obrazu není tak hladký jako odhad získaný pouze VB. Výstup z neslepé dekonvoluce je mírně rozmazaný, ale bližší ostrému obrazu než poškozenému. Algoritmus dosáhl stabilního řešení do 200 iterací. PSNR jak přímého odhadu tak neslepé dekonvoluce lze nalézt v tabulce 3.4.

PSF odhadnutá z obrazu rozmazaného lineárním pohybem velmi připomíná původní, ale nepřechází ostře z vysokých hodnot do nulových. Přímý odhad obrazu je velmi podobný odhadu s gaussovskou PSF a nejsou v něm viditelné známky po rozmazání pohybem. Opět není tak hladký jako odhad nalezený algoritmem IVB-diag. Podobně jako u odhadu VB je na výstupu z neslepé dekonvoluce patrné rozmazání. Stabilního řešení bylo dosaženo do sta iterací. PSNR je možné nalézt v tabulce 3.4.

Stejně jako u odhadu pouze přes VB, odhad PSF špatného zaostření připomíná spíše gaussovskou PSF, i když tvar té původní je v ní výraznější. Ostrý obraz je velmi podobný ostrým obrazům nalezeným pro druhé dvě PSF. Výstup z neslepé dekonvoluce je o něco rozostřenější než ten nalezený z gaussovské PSF, ale je ostřejší než poškozený obraz. Stabilní řešení bylo nalezeno do sta iterací. PSNR těchto odhadů i odhadů pro další hladiny šumu lze nalézt v tabulce 3.4.



Obrázek 3.5: Výsledky pro ELBO-diag algoritmus (algoritmus (3)) a SNR 40dB. V prvním řádku jsou uvedeny výsledky pro gaussovské rozostření, v druhém pro rozmazání lineárním pohybem a v třetím pro špatné zaostření. V každém řádku je na levé straně nahoře ukázáno původní jádro, pod ním je jeho odhad. Následuje zleva původní čistý obraz, rozmazaný obraz s přidaným šumem, přímý výstup z algoritmu a výsledek neslepé dekonvoluce.

Narozdíl od algoritmu IVB-diag jsou PSNR přímých odhadů při SNR vyšším než 30 dB při gaussovském rozmazání vyšší než PSNR poškozených obrazů (tabulka 3.1), ale i v tomto případě přímý odhad i neslepá dekonvoluce při SNR 20dB selhávají, stejně jako neslepá dekonvoluce při rozmazání lineárním pohybem a stejné hladině šumu. Podobně jako u algoritmu IVB-diag je i PSNR neslepé dekonvoluce při SNR 20dB nižší než u poškozeného obrazu.

Srovnáním hodnot PSNR v tabulkách 3.3 a 3.4 vidíme, že algoritmus využívající odhad přes ELBO s plnou kovariancí  $\Sigma_k$  dosahuje výrazně nižších hodnot PSNR než všechny ostatní. Pro rozmazání lineárním pohybem je přímý odhad přesnější než neslepá dekonvoluce (s výjimkou ELBO-full), pro druhé dvě PSF to však neplatí. Odhady získané maximalizací ELBO přes parametry  $k$  při rozmazání lineárním pohybem mají výrazně nižší PSNR než odhady nalezené pouze VB. Výsledky získané na obrázcích s gaussovským rozmazáním a špatným zaostřením jsou pro oba algoritmy s diagonální kovariancí srovnatelné a s výjimkou gaussovské PSF a SNR 50dB nedosahuje IVB-full algoritmus lepších výsledků než IVB-diag a ELBO-diag.

To, že odhady ostrého obrazu nejsou úplně hladké ale sestávají z menších oblastí stejných hodnot je důsledek volby apriorního rozdělení na  $x$ . Toto rozdělení společně s rozdělením přesností  $\tau_x$  podle ARD principu předpokládá, že derivace na obrázku jsou řídké, takže obrázek by



Obrázek 3.6: Výsledky pro ELBO-diag algoritmus (algoritmus (3)) a SNR 40dB. V první řádce jsou uvedeny výsledky pro gaussovské rozostření, v druhém pro rozmazání lineárním pohybem a v třetím pro špatné zaostření. V každém řádku je na levé straně nahoře ukázáno původní jádro, pod ním je jeho odhad. Následuje zleva původní čistý obraz, rozmazaný obraz s přidáním šumem, přímý výstup z algoritmu a výsledek neslepé dekonvoluce..

měl být po částech hladký. Tento efekt je výraznější u obrázků s diagonální kovariancí na  $\Sigma_k$ , protože je zanedbána závislost mezi jednotlivými prvky PSF.

Algoritmus	PSF	SNR	PSNR $\mu_X$	PSNR dek.	koef.dek.
ELBO-full	gauss	20	18.056	25.12414	0.9
ELBO-full	gauss	30	17.19244	28.2933	0.9
ELBO-full	gauss	40	17.47427	29.25239	0.7
ELBO-full	gauss	50	17.11647	29.01416	0.9
ELBO-full	pohyb	20	16.98284	22.07161	0.9
ELBO-full	pohyb	30	17.54835	23.35345	0.9
ELBO-full	pohyb	40	17.23765	23.67559	0.9
ELBO-full	pohyb	50	17.76976	23.90629	0.9
ELBO-full	zaostření	20	17.68528	24.72895	0.9
ELBO-full	zaostření	30	17.36233	27.95733	0.9
ELBO-full	zaostření	40	17.37574	28.29448	0.7
ELBO-full	zaostření	50	17.79887	28.28094	0.7
ELBO-diag	gauss	20	25.64079	25.91173	0.9
ELBO-diag	gauss	30	27.45341	29.96938	0.5
ELBO-diag	gauss	40	27.56654	32.57338	0.3
ELBO-diag	gauss	50	28.20815	34.88439	0.1
ELBO-diag	pohyb	20	24.54282	22.59003	0.9
ELBO-diag	pohyb	30	26.48846	24.07055	0.9
ELBO-diag	pohyb	40	26.90665	24.29049	0.9
ELBO-diag	pohyb	50	26.89334	24.30564	0.9
ELBO-diag	zaostření	20	25.72165	25.40854	0.9
ELBO-diag	zaostření	30	27.08185	28.69705	0.5
ELBO-diag	zaostření	40	27.56204	30.19888	0.3
ELBO-diag	zaostření	50	27.45429	30.55224	0.3

Tabulka 3.4: Tabulka hodnot PSNR výsledků algoritmů IVB-full a IVB-diag. PSNR  $\mu_X$  značí PSNR přímého výstupu z algoritmu, koef. dek. je koeficient použitý pro nesleponou dekonvoluci a PSNR dek. PSNR obrazu nalezeného nesleponou dekonvolucí. SNR i PSNR je uvedeno v decibelech.

### 3.3 Tridiagonální $S_k$

Odhad s plnou kovarianční maticí  $\mathbf{k}$  by měl dosahovat lepších výsledků než odhad s diagonální kovariancí, nicméně u zde zmíněných algoritmů, které odhadují  $\mu_k$  a  $S_k$  přes ELBO, to tak není. To může být způsobeno příliš hrubou aproximací reparametrizačním trikem, kde se v odhadu momentů 2.12 používá pouze jeden vzorek  $\epsilon$ . Nejprve proto zkusíme omezit  $S_k$  způsobem, který zachovává vztahy sousedních pixelů, ale stále neobsahuje příliš velké množství prvků k odhadu. Předpokládejme, že  $S_k$  je tridiagonální - má hlavní diagonálu a dvě vedlejší. První vedlejší diagonála představuje přímé sousedy pixelů zprava a druhá přímé sousedy pixelů zespoda.  $\Sigma_k$  má potom šest vedlejších diagonál a mezi nenulovými prvky jsou kovariance sousedů zprava zespoda. Už tato aproximace by v případě použití VB byla problematická, protože inverze matice, která má nenulových pouze 7 diagonál, nemá tu samou strukturu. Algoritmus je potom velmi podobný algoritmu ELBO-full, resp. ELBO-diag, pouze se při hledání odhadu  $S_k$  hledají odhady jejích třech diagonál. Tento algoritmus nazveme ELBO-tdmat.

Inicializace je stejná jako v předchozích případech, tedy podle tabulky 3.1. Počáteční hodnoty hlavní diagonály jsou nastavené stejně jako diagonála  $\mathbf{S}_k$ , tedy na hodnoty  $10^{-1}$ , a vedlejší diagonály jsou na počátku nulové. Počáteční odhad je no-blur řešení.

Na obrázku 3.7 můžeme vidět výsledky dekonvoluce s tridiagonální  $\mathbf{S}_k$ . Odhadnuté PSF i ostré obrázky jsou velmi podobné těm nalezeným algoritmem IBELBO-diag. Pro všechny PSF platí, že přímý odhad ostrého obrázku není úplně hladký, ale sestává z menších regionů stejných hodnot. Porovnáme-li hodnoty PSNR, které jsou pro tridiagonální  $\mathbf{S}_k$  uvedeny v tabulce 3.5, s hodnotami PSNR předchozích testů (tabulky 3.3 a 3.4), vidíme, že výsledky neslepé dekonvoluce jsou zhruba stejné jako při použití algoritmu ELBO-diag. Při rozmazání lineárním jsou hodnoty PSNR nízké jak pro přímý odhad, tak pro slepou dekonvoluci. PSNR přímých výstupů je u algoritmu ELBO-tdmat nižší, než u ELBO-diag.

Ačkoliv přímé odhady ostrých obrazů působí mnohem lépe, PSNR poškozených obrazů (viz. tabulka 3.1) je vyšší v případě gaussovského rozostření pro všechny hladiny šumu a v případě špatného zaostření při SNR 40 dB a 50 dB. Přímé odhady při rozmazání lineárním pohybem mají PSNR vyšší než poškozený obraz pro všechny testované hladiny šumu. Podobně jako u předchozích algoritmů, při neslepé dekonvoluci algoritmus selhává při SNR 20dB.



Obrázek 3.7: Výsledky pro ELBO-tdmat algoritmus (algoritmus (3) s tridiagonální  $\mathbf{S}_k$ ) a SNR 40dB. V první řádce jsou uvedeny výsledky pro gaussovské rozostření, v druhém pro rozmazání lineárním pohybem a v třetím pro špatné zaostření. V každém řádku je na levé straně nahoře ukázáno původní jádro, pod ním je jeho odhad. Následuje zleva původní čistý obraz, rozmazaný obraz s přidáním šumu, přímý výstup z algoritmu a výsledek neslepé dekonvoluce.



Algoritmus	PSF	SNR	PSNR $\mu_X$	PSNR dek.	koef.dek.
ELBO-tdmat	gauss	20	25.57002	25.86696	0.9
ELBO-tdmat	gauss	30	26.43757	29.84517	0.5
ELBO-tdmat	gauss	40	26.46331	32.47787	0.3
ELBO-tdmat	gauss	50	26.09546	32.82565	0.3
ELBO-tdmat	pohyb	20	23.93896	22.59232	0.9
ELBO-tdmat	pohyb	30	24.66366	24.01614	0.9
ELBO-tdmat	pohyb	40	24.74673	24.24821	0.9
ELBO-tdmat	pohyb	50	24.87366	24.28223	0.9
ELBO-tdmat	zaostření	20	25.35019	25.37963	0.9
ELBO-tdmat	zaostření	30	26.21226	28.66438	0.5
ELBO-tdmat	zaostření	40	26.06886	30.08144	0.3
ELBO-tdmat	zaostření	50	25.91231	30.44953	0.3

Tabulka 3.5: Tabulka hodnot PSNR výsledků algoritmů ELBO-tdmat. PSNR  $\mu_X$  značí PSNR přímého výstupu z algoritmu, koef. dek. je koeficient použitý pro neslepu dekonvoluci a PSNR dek. PSNR obrazu nalezeného neslepu dekonvolucí. SNR i PSNR je uvedeno v decibelech.

### 3.4 Větší počet vzorků pro reparametrizaci

Ačkoliv odhady přes ELBO získané s tridiagonální  $\mathbf{S}_k$  jsou výrazně lepší než odhady získané s plnou kovariancí, IVB algoritmus v některých případech stále dosahuje lepších výsledků. V této sekci se vrátíme zpět k reparametrizačnímu triku. Pro aproximaci středních hodnot je v algoritmu ELBO-full použit pouze jeden vzorek  $\epsilon$ , což může být pro odhad plné kovarianční matice nedostatečné. Předpokládáme-li rozměr PSF 5, je rozměr  $\Sigma_k$   $25 \times 25$ , tedy pokud by byla regulární, její hodnota by byla rovna 25. V této sekci proto otestujeme algoritmus, který v reparametrizačním triku využívá 25 vzorů  $\epsilon$  a nazveme ho ELBO-full25, je zde vypsán jako algoritmus 4.

Počáteční hodnoty parametrů jsou nastaveny stejně jako u algoritmu ELBO-full a počáteční odhad ostrého obrazu je  $\mathbf{d}$ , počáteční odhad PSF Diracova  $\delta$ -funkce.

Odhady ostrého obrazu a původních PSF jsou na obrázku 3.8. Srovnáme-li je s výsledky při použití jednoho vzorku (obrázek 3.5), jsou mnohem blíže původním PSF i ostrému obrazu, ale i tak odhady s diagonální kovariancí (obrázek 3.6) a tridiagonální  $\mathbf{S}_k$  působí lépe. Podle tabulky 3.6, kde jsou uvedena PSNR přímých odhadů ostrého obrazu a neslepé dekonvoluce, nedosahuje tento algoritmus takové přesnosti jako algoritmy ELBO-diag (tabulka 3.4) a ELBO-tdmat (tabulka 3.5).

Srovnáme-li tabulky 3.6 a 3.1, vidíme, že přímé odhady při gaussovské PSF a špatném zaostření jsou horší než rozmazaný obraz. V případě rozmazání lineárním pohybem jsou dle PSNR odhady jen mírně lepší. Pro neslepé dekonvoluce opět platí, že poškozený obraz je přesnější než odhad při SNR 20dB.

Dalším navýšením počtu vzorků je sice možné dosáhnout přesnějších odhadů, avšak výpočetní náročnost se s počtem vzorků zvyšuje tak výrazně, že se tento přístup nevyplatí.

---

**Algoritmus 4:** ELBO-full25 algoritmus

---

Inicializace podle tabulky 3.2,  $\boldsymbol{\mu}_x = \mathbf{d}$ ,  $\boldsymbol{\mu}_k = \boldsymbol{\delta}$ ;

**while** není dosaženo konvergence **do**

    Spočti  $\boldsymbol{\Sigma}_x = \text{diag} \left( \left( \text{diag} \left( \widehat{\omega} \widehat{\mathbf{K}}^T \widehat{\mathbf{K}} + \nabla^T \text{diag}(\widehat{\boldsymbol{\tau}}_x) \nabla \right) \right)^{-1} \right)$ ;

    Spočti  $\boldsymbol{\mu}_x$  řešením (3.2) konjugovanými gradienty;

    Přepočti  $\widehat{\boldsymbol{x}}$ ,  $\widehat{\mathbf{X}}^T \widehat{\mathbf{X}}$ ,  $\mathbb{E}_{\tilde{f}} \left[ \mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x} \right]$  a  $\mathbb{E}_{\tilde{f}} \left[ \mathbf{x}^T \nabla^T \mathcal{L}_j \nabla \mathbf{x} \right]$ ;

**for** všechna  $i \in \widehat{n} \widehat{p}$  **do**

$\widehat{\tau}_{xj} = \left( \frac{1}{2} \mathbb{E}_{\tilde{f}(x|d)} \left[ \mathbf{x}^T \nabla^T \mathcal{L}_j \nabla \mathbf{x} \right] + \beta_{x0} \right)^{-1} \left( \alpha_{x0} - \frac{1}{2} \right)$ ;

**end**

    Spočti  $\widehat{\boldsymbol{\tau}}_k = \left( \frac{1}{2} \widehat{\mathbf{k}}^T \widehat{\mathbf{k}} + \beta_{k0} \right)^{-1} \left( \frac{m}{2} + \alpha_{k0} - 1 \right)$ ;

    Spočti  $\widehat{\omega} = \left( \frac{np}{2} + \gamma_0 - 1 \right) \left( \frac{1}{2} \left( \mathbf{d}^T \mathbf{d} - 2 \mathbf{d}^T \widehat{\mathbf{K}} \widehat{\boldsymbol{x}} + \mathbb{E}_{\tilde{f}(x,k|d)} \left[ \mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x} \right] \right) + \eta_0 \right)^{-1}$ ;

    Nastav optimalizátor Adam na krok  $\boldsymbol{\eta} = 0.001$ ;

**for** všechna  $i \in \{1, \dots, 300\}$  **do**

        Nastav průměr gradientů  $\bar{\mathbf{g}}$  funkce (3.6) vzhledem k  $\boldsymbol{\mu}_k$  a  $\mathbf{S}_k$  na nulu;

**for** všechna  $j \in \{1, \dots, 25\}$  **do**

            Vezmi jeden vzorek  $\bar{\boldsymbol{\epsilon}}$  z  $\mathcal{N}(\mathbf{0}, \mathbb{I})$ ;

            Spočti gradienty (3.6) vzhledem k  $\boldsymbol{\mu}_k$  a  $\mathbf{S}_k$ ;

            Přičti gradienty k  $\bar{\mathbf{g}}$

**end**

        Přepočti  $\boldsymbol{\mu}_k$  a  $\mathbf{S}_k$  pomocí optimalizátoru Adam z gradientů  $\bar{\mathbf{g}}/25$ ;

**end**

    Záporné hodnoty  $\boldsymbol{\mu}_k$  nastav na nulu a poděl  $\sum_i \mu_{ki}$ ;

    Přepočti  $\widehat{\mathbf{K}}$ ,  $\widehat{\mathbf{K}}^T \widehat{\mathbf{K}}$  a  $\mathbf{k}^T \mathbf{k}$ ;

**end**

---



Obrázek 3.8: Výsledky pro ELBO-full25 algoritmus (algoritmus (3) s 25 vzorky v reparametrizačním triku) a SNR 40dB. V první řádce jsou uvedeny výsledky pro gaussovské rozostření, v druhém pro rozmazání lineárním pohybem a v třetím pro špatné zaostření. V každém řádku je na levé straně nahoře ukázáno původní jádro, pod ním je jeho odhad. Následuje zleva původní čistý obraz, rozmazaný obraz s přidaným šumem, přímý výstup z algoritmu a výsledek neslepé dekonvoluce.

Algoritmus	PSF	SNR	PSNR $\mu_X$	PSNR dek.	koef.dek.
ELBO-full25	gauss	20	24.24949	25.73553	0.9
ELBO-full25	gauss	30	24.49459	29.52816	0.5
ELBO-full25	gauss	40	24.45025	31.24264	0.3
ELBO-full25	gauss	50	24.85861	32.02233	0.3
ELBO-full25	pohyb	20	23.09611	22.52757	0.9
ELBO-full25	pohyb	30	23.28300	23.94642	0.9
ELBO-full25	pohyb	40	23.26045	24.13534	0.9
ELBO-full25	pohyb	50	23.42590	24.13139	0.9
ELBO-full25	zaostření	20	24.07294	25.28852	0.9
ELBO-full25	zaostření	30	24.32960	28.43265	0.7
ELBO-full25	zaostření	40	24.40191	29.54455	0.5
ELBO-full25	zaostření	50	24.41261	29.82959	0.3

Tabulka 3.6: Tabulka hodnot PSNR výsledků algoritmů ELBO-full25. PSNR  $\mu_X$  značí PSNR přímého výstupu z algoritmu, koef. dek. je koeficient použitý pro neslepou dekonvoluci a PSNR dek. PSNR obrazu nalezeného neslepou dekonvolucí. SNR i PSNR je uvedeno v decibelech.

### 3.5 VADAM

Poslední variantou ELBO algoritmu v této kapitole bude algoritmus, který místo optimalizátoru Adam využívá optimalizátor Vadam [22] popsany na konci první kapitoly. Díky tomu, že aproximace aposterioriho rozdělení PSF je rozdělení normální, není nutné počítat gradient vzhledem k  $\mathbf{S}_k$ . Tento postup obecně předpokládá rozdělení  $\mathbf{k}$  z exponenciální rodiny – v tomto případě normální – a diagonální kovarianční matici  $\mathbf{\Sigma}_k$ . Algoritmus je zde vypsany jako algoritmus 5.

---

#### Algoritmus 5: ELBO-Vadam algoritmus

---

```

Inicializace podle tabulky 3.2,  $\boldsymbol{\mu}_x = \mathbf{d}$ ,  $\boldsymbol{\mu}_k = \boldsymbol{\delta}$ ;
while není dosaženo konvergence do
    Spočti  $\boldsymbol{\Sigma}_x = \text{diag} \left( \left( \text{diag} \left( \widehat{\boldsymbol{\omega}} \widehat{\mathbf{K}}^T \widehat{\mathbf{K}} + \nabla^T \text{diag}(\widehat{\boldsymbol{\tau}}_x) \nabla \right) \right)^{-1} \right)$ ;
    Spočti  $\boldsymbol{\mu}_x$  řešením (3.2) konjugovanými gradienty;
    Přepočti  $\widehat{\mathbf{x}}$ ,  $\widehat{\mathbf{X}}^T \widehat{\mathbf{X}}$ ,  $\mathbb{E}_{\widehat{f}} [\mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x}]$  a  $\mathbb{E}_{\widehat{f}} [\mathbf{x}^T \nabla^T \mathcal{I}_j \nabla \mathbf{x}]$ ;
    for všechna  $i \in \widehat{np}$  do
         $\widehat{\tau}_{xj} = \left( \frac{1}{2} \mathbb{E}_{\widehat{f}(x|d)} [\mathbf{x}^T \nabla^T \mathcal{I}_j \nabla \mathbf{x}] + \beta_{x0} \right)^{-1} \left( \alpha_{x0} - \frac{1}{2} \right)$ ;
    end
    Spočti  $\widehat{\boldsymbol{\tau}}_k = \left( \frac{1}{2} \widehat{\mathbf{k}}^T \widehat{\mathbf{k}} + \beta_{k0} \right)^{-1} \left( \frac{m}{2} + \alpha_{k0} - 1 \right)$ ;
    Spočti  $\widehat{\boldsymbol{\omega}} = \left( \frac{mp}{2} + \gamma_0 - 1 \right) \left( \frac{1}{2} \left( \mathbf{d}^T \mathbf{d} - 2\mathbf{d}^T \widehat{\mathbf{K}} \widehat{\mathbf{x}} + \mathbb{E}_{\widehat{f}(x,k|d)} [\mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x}] \right) + \eta_0 \right)^{-1}$ ;
    Nastav optimalizátor Vadam na krok  $\boldsymbol{\eta} = 0.01$ ;
    for všechna  $i \in \{1, \dots, 1000\}$  do
        Vezmi jeden vzorek  $\mathbf{k}^{(i)}$  z  $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ ;
        Spočti gradient  $\ln p(\mathbf{x}, \mathbf{k}, \boldsymbol{\omega} | \mathbf{d})$  vzhledem k  $\boldsymbol{\mu}_k$  v bodě  $\mathbf{k}^{(i)}$ ;
        Přepočti  $\boldsymbol{\mu}_k$  a  $\boldsymbol{\Sigma}_k$  pomocí optimalizátoru Vadam;
    end
    Záporné hodnoty  $\boldsymbol{\mu}_k$  nastav na nulu a poděl  $\sum_i \mu_{ki}$ ;
    Přepočti  $\widehat{\mathbf{K}}$ ,  $\widehat{\mathbf{K}}^T \widehat{\mathbf{K}}$  a  $\mathbf{k}^T \mathbf{k}$ ;
end

```

---

Odhady pro všechny tři PSF jsou na obrázku 3.9. Přímý odhad ostrého obrazu je velmi podobný původnímu obrazu, konstantní plochy, které se často objevovaly v předchozích odhadech, jsou patrné jen minimálně. Speciálně v případě rozmazání lineárním pohybem vypadá přímý odhad opravdu hladce. Neslepé dekonvoluce působí jen mírně rozostřeně v případě gaussovského rozmazání a špatného zaostření. Při rozmazání pohybem je přímý odhad lepší než výstup z neslepé dekonvoluce, kde je rozmazání stále patrné. Odhad gaussovské PSF je velmi podobný původní PSF, PSF rozmazání pohybem působí také velmi podobně, ale přechod mezi diagonálou a originálně nulovým okolím není tak ostrý. V případě špatného zaostření odhad jádra připomíná spíše gaussovskou PSF, ale obrys původní PSF je tam výrazný. Algoritmus dosáhne stabilního řešení do sta kroků, v případě gaussovské PSF a špatného zaostření už po padesáti krocích.

PSNR těchto odhadů jsou vypsána v tabulce 3.7. Srovnáme-li je s tabulkou odhadů nalezených algoritmem IVB-full, které jsou v tabulce 3.3, vidíme, že algoritmus ELBO-Vadam dosahuje

skoro stejných hodnot. Navíc přímé odhady pro rozmazání lineárním pohybem mají podobné PSNR, čemuž se žádný jiný ze zde prezentovaných algoritmů nepřiblížil.

Porovnáme-li PSNR odhadů uvedené v tabulce 3.7 s PSNR poškozených obrazů v tabulce 3.1, vidíme, že algoritmus ELBO-Vadam selhává pouze při neslepé dekonvoluci obrazu se SNR 20 dB a to pro všechny PSF.



Obrázek 3.9: Výsledky pro ELBO-Vadam algoritmus (algoritmus 5) a SNR 40dB. V první řádce jsou uvedeny výsledky pro gaussovské rozostření, v druhém pro rozmazání lineárním pohybem a v třetím pro špatné zaostření. V každém řádku je na levé straně nahoře ukázáno původní jádro, pod ním je jeho odhad. Následuje zleva původní čistý obraz, rozmazaný obraz s přidaným šumem, přímý výstup z algoritmu a výsledek neslepé dekonvoluce.

Algoritmus	PSF	SNR	PSNR $\mu_X$	PSNR dek.	koef.dek.
ELBO-Vadam	gauss	20	26.14768	25.95086	0.9
ELBO-Vadam	gauss	30	28.16768	29.89638	0.5
ELBO-Vadam	gauss	40	29.61464	32.59987	0.3
ELBO-Vadam	gauss	50	29.91588	36.18125	0.1
ELBO-Vadam	pohyb	20	25.05887	22.60024	0.9
ELBO-Vadam	pohyb	30	28.84046	24.08326	0.9
ELBO-Vadam	pohyb	40	33.5514	24.27814	0.9
ELBO-Vadam	pohyb	50	35.89974	24.28925	0.9
ELBO-Vadam	zaostření	20	25.83936	25.41612	0.9
ELBO-Vadam	zaostření	30	27.7055	28.69195	0.5
ELBO-Vadam	zaostření	40	29.19083	30.11952	0.3
ELBO-Vadam	zaostření	50	29.54973	30.43501	0.3

Tabulka 3.7: Tabulka hodnot PSNR výsledků algoritmů ELBO-Vadam. PSNR  $\mu_X$  značí PSNR přímého výstupu z algoritmu, koef. dek. je koeficient použitý pro neslepou dekonvoluci a PSNR dek. PSNR obrazu nalezeného neslepou dekonvolucí. SNR i PSNR je uvedeno v decibelech.

### 3.6 Zhodnocení experimentu

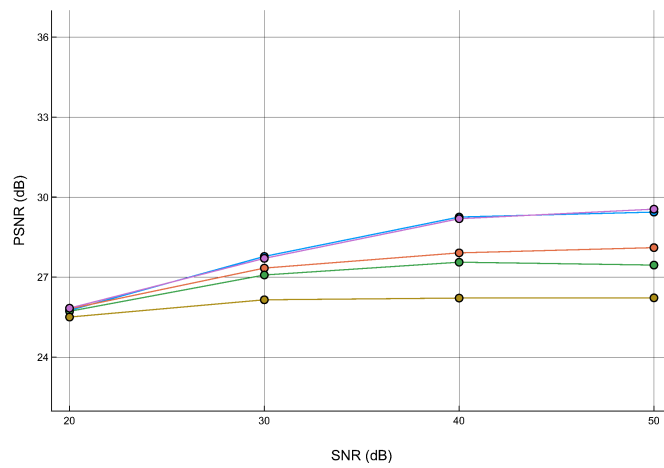
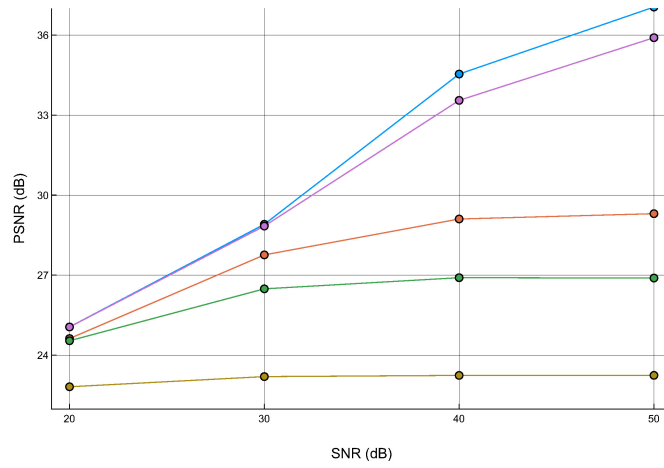
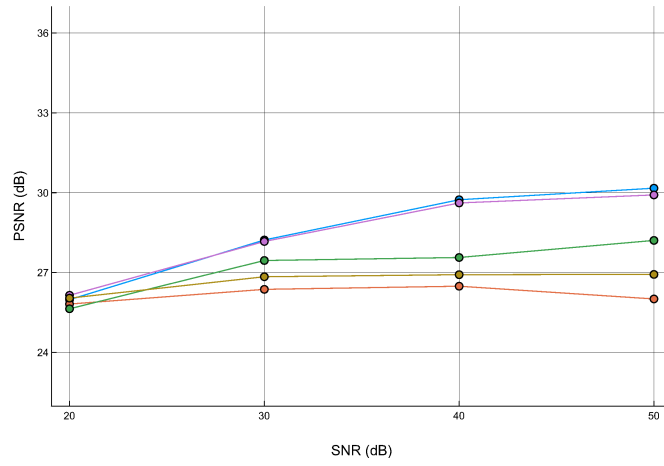
Z experimentů v této kapitole vyplývá následující

- Použití diagonální kovarianční matice  $\Sigma_k$  v IVB algoritmu se v odhadu projevuje a to především v případě rozmazání lineárním pohybem. Přechody v mezi pixely v odhadu PSF nejsou tak ostré jako v originálních PSF a obraz je naopak méně hladký a více se v něm objevují konstantní plochy. V obou případech jsou přímé odhady při rozmazání lineárním pohybem lepší než výstupy z neslepé dekonvoluce, ale hodnoty PSNR varianty s diagonální kovariancí jsou výrazně nižší než varianty s plnou kovariancí. PSNR obrazů z neslepé dekonvoluce jsou pro obě varianty relativně podobné. Tyto závěry jsou dobře vidět na obrázcích 3.10 a 3.11, IVB-full algoritmu odpovídá modrá linka a IVB-diag červená.
- V případě diagonální kovarianční matice PSF je odhad přes ELBO s jedním vzorkem v reparametrizačním triku srovnatelný s výsledky algoritmu IVB-diag, s výjimkou rozmazání lineárním pohybem, kde je přímý odhad přes ELBO výrazně horší, jak je vidět na obrázcích 3.10 a 3.11. Odhad s plnou kovarianční maticí  $\Sigma_k$  je při jednom vzorku v reparametrizačním triku naprosto nedostatečný.
- Omezení struktury  $\Sigma_k$  na několik diagonál při odhadu přes ELBO přineslo ve srovnání s plnou kovariancí zlepšení, ale i tak jsou výsledky mírně horší než u diagonální varianty. Potvrdilo se, že jeden vzorek v reparametrizačním triku není dostatečný pro odhad plné kovariance, nicméně jejich navýšení vyžaduje ve srovnání s ostatními algoritmy příliš vysoký výpočetní čas.
- Použití optimalizátoru Vadam bylo přínosné a i přes to, že uvažuje diagonální kovarianční matici PSF, dosáhl stejně dobrých výsledků jako algoritmus IVB-full. Od ostatních zde uvedených ELBO algoritmů se liší především v odhadu  $\Sigma_k$ , v případě Vadam se nepočítá přímo gradient ELBO vzhledem k  $S_k$ , ale je aproximován pomocí gradientu vzhledem  $\mu_k$ .

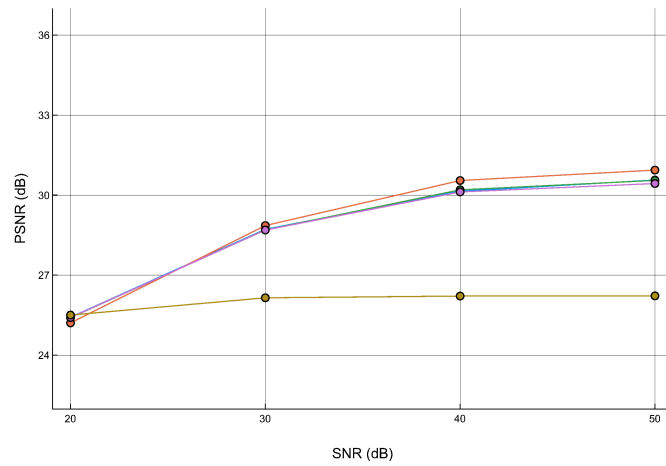
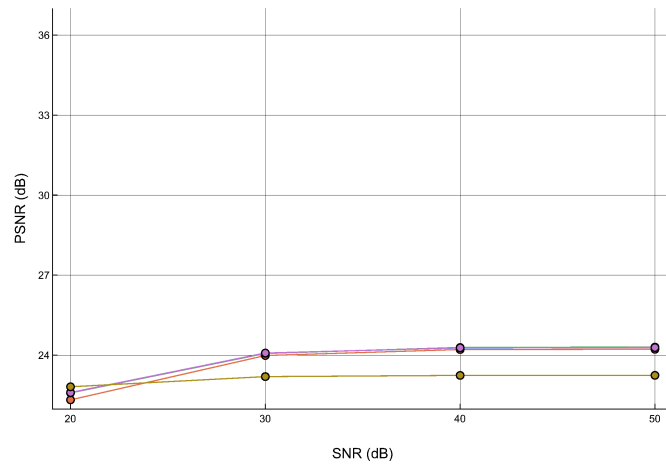
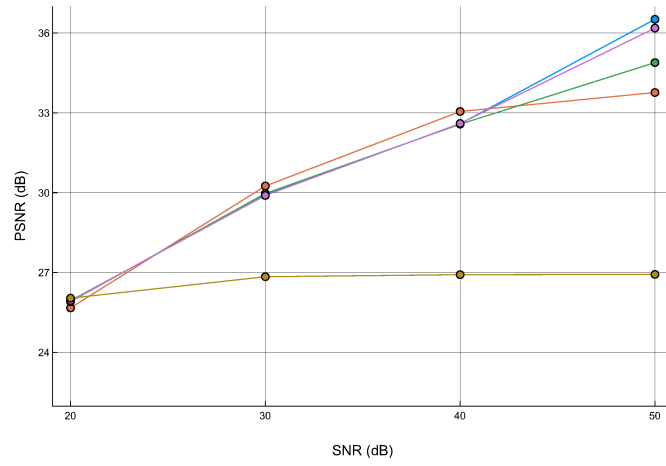
Podíváme-li se do sekce 3.5, je hodnota nové  $\Sigma_k$  daná hodnotou v předchozím kroku a kvadrátem změny věrohodnosti vzhledem k  $\mathbf{k}$ , neuchovávají se tedy předchozí změny na  $\Sigma_k$ , ale pouze její předchozí hodnoty. Vliv předchozích změn by se dal ověřit použitím jiného optimalizátoru při maximalizaci ELBO, například RMSprop [49], který neuchovává předchozí hodnoty gradientů, pouze jejich druhé mocniny. Navíc při použití optimalizátoru Adam se gradient počítaný v náhodném bodě vyskytuje v odhadu  $\mathbf{S}_k$  dvakrát, takže by mohl mít větší vliv.

- Při SNR 20dB ve většině případů algoritmy nejsou schopny nalézt lepší odhad ostrého obrazu než je obraz rozmazaný.





Obrázek 3.10: Graf PSNR pro přímé výstupy z algoritmů. První graf je pro gaussovskou PSF, druhý pro rozmazání lineárním pohybem a třetí pro špatné zaostření. Modrá křivka značí IVB-full algoritmus, červená IVB-diag, zelená ELBO-diag, fialová ELBO-Vadam a hnědá PSNR rozmazaného obrazu.



Obrázek 3.11: Graf PSNR pro neslepou dekonvoluci. První graf je pro gaussovskou PSF, druhý pro rozmazání lineárním pohybem a třetí pro špatné zaostření. Modrá křivka značí IVB-full algoritmus, červená IVB-diag, zelená ELBO-diag, fialová ELBO-Vadam a hnědá PSNR rozmazaného obrazu.

### 3.7 Odhad nosiče

V této sekci bude otestována schopnost správně odhadnout nosič (support) u nejlepších algoritmů, tj. IVB-full a ELBO-Vadam. Na stejných obrázcích se stejnými PSF bude hledán rozklad za předpokladu, že rozměr PSF není  $5 \times 5$  pixelů, ale  $7 \times 7$ .

Na obrázcích 3.12 a 3.13 jsou vyobrazeny odhady ostrých obrazů a PSF. Obrázky původních PSF byly pro lepší srovnání doplněny o nuly. PSF jsou oběma dvěma algoritmy odhadnuty velmi podobně. Gaussovská PSF a špatné zaostření jsou v obou případech velmi podobné výsledkům získaným při znalosti velikosti nosiče, pro rozmazání lineárním pohybem to však neplatí, pixely okolo nenulové diagonály jsou odhadnuty s vyššími hodnotami než když se hledá PSF o velikosti  $5 \times 5$ . Přímý odhad vypadá v obou případech tmavší, ale to je důsledek škálování obrazu tak, aby maximální hodnota byla rovna jedné, a toho, že se pixel v levém horním rohu odhadl s extrémně vysokou hodnotou.



Obrázek 3.12: Výsledky pro IVB-full algoritmus (algoritmus (1)) při hledání větší PSF než byla původní, SNR je 40dB. V první řádce jsou uvedeny výsledky pro gaussovské rozostření, v druhém pro rozmazání lineárním pohybem a v třetím pro špatné zaostření. V každém řádku je na levé straně nahoře ukázáno původní jádro, pod ním je jeho odhad. Následuje zleva původní čistý obraz, rozmazaný obraz s přidáním šumu, přímý výstup z algoritmu a výsledek neslepé dekonvoluce.

V tabulce 3.8 jsou vypsány PSNR odhadů nalezených při předpokladu většího nosiče PSF. Rozdíly mezi oběma algoritmy pro výstup ze slepé dekonvoluce jsou minimální, při přímém odhadu se liší více, ale úspěšnost rekonstrukce při srovnání s PSNR poškozeného obrazu je stejná.

U gaussovské PSF došlo k vylepšení při SNR 40 a 50 dB, u rozmazání lineárním pohybem je PSNR přímého odhadu vždy vyšší než PSNR poškozeného a u špatného zaostření nedošlo k vylepšení ani v jednom případě. Odhad z neslepé dekonvoluce je podle PSNR lepší než původní obraz pro SNR vyšší než 20 dB u obou algoritmu.

Ve srovnání s případem, kdy je velikost nosiče známá, je odhad gaussovského jádra a špatného zaostření podobně úspěšný, ale pokud je obraz rozmazaný lineárním pohybem, přímý odhad nedosahuje zdaleka tak dobrých výsledků. Při nastavení počátečních hodnot podle tabulky 3.2 je střední hodnota  $\tau_x 10^5$ , tzn. variance derivací obrazu je podle apriorního rozdělení  $10^{-5}$ . Diference na obraze jsou rozdíly hodnot mezi 0 a 1, takže apriorně jsou předpokládány malé změny. Pokud je střední hodnota  $\tau_x$  apriorně nastavena na  $10^4$ , tedy variance na derivacích obrazu je větší, PSF se odhadne ostřeji, a větší hladkost je v nalezeném obraze. Pokud by velikost nosiče nebyla známá, bylo by na místě prozkoumat lepší nastavení počátečních hodnot.



Obrázek 3.13: Výsledky pro ELBO-Vadam algoritmus (algoritmus (5)) při hledání větší PSF než byla původní, SNR je 40dB. V první řádce jsou uvedeny výsledky pro gaussovské rozostření, v druhém pro rozmazání lineárním pohybem a v třetím pro špatné zaostření. V každém řádku je na levé straně nahoře ukázáno původní jádro, pod ním je jeho odhad. Následuje zleva původní čistý obraz, rozmazaný obraz s přidaným šumem, přímý výstup z algoritmu a výsledek neslepé dekonvoluce.

Algoritmus	PSF	SNR	PSNR $\mu_X$	PSNR dek.	koef.dek.
IVB-full	gauss	20	24.48493	25.76123	0.9
IVB-full	gauss	30	25.57153	29.99723	0.5
IVB-full	gauss	40	27.78306	32.79321	0.3
IVB-full	gauss	50	28.28783	34.47357	0.1
IVB-full	pohyb	20	23.3982	22.59123	0.9
IVB-full	pohyb	30	23.5698	24.11939	0.9
IVB-full	pohyb	40	25.12464	24.33914	0.9
IVB-full	pohyb	50	25.60605	24.35405	0.9
IVB-full	zaostření	20	24.13294	25.39351	0.9
IVB-full	zaostření	30	25.18908	28.92669	0.5
IVB-full	zaostření	40	26.11123	30.39558	0.3
IVB-full	zaostření	50	25.81348	30.83922	0.3
ELBO-Vadam	gauss	20	24.42038	25.7742	0.9
ELBO-Vadam	gauss	30	25.82459	30.01543	0.5
ELBO-Vadam	gauss	40	27.3828	32.76683	0.3
ELBO-Vadam	gauss	50	28.70521	34.8226	0.1
ELBO-Vadam	pohyb	20	23.27054	22.57489	0.9
ELBO-Vadam	pohyb	30	24.15666	24.12297	0.9
ELBO-Vadam	pohyb	40	25.68684	24.32642	0.9
ELBO-Vadam	pohyb	50	25.83678	24.34057	0.9
ELBO-Vadam	zaostření	20	24.29496	25.41041	0.9
ELBO-Vadam	zaostření	30	25.07704	28.91728	0.5
ELBO-Vadam	zaostření	40	26.09822	30.45038	0.3
ELBO-Vadam	zaostření	50	26.19788	30.82875	0.3

Tabulka 3.8: Tabulka hodnot PSNR výsledků algoritmů IVB-full a ELBO-Vadam při hledání větší PSF než byla původní. PSNR  $\mu_X$  značí PSNR přímého výstupu z algoritmu, koef. dek. je koeficient použitý pro neslepu dekonvoluci a PSNR dek. PSNR obrazu nalezeného neslepu dekonvolucí. SNR i PSNR je uvedeno v decibelech.



## Kapitola 4

# Odhad z nekompletního snímku

V rastrovacím elektronovém mikroskopu je obraz tvořen skenováním pixel po pixelu, ale vědce, kteří jej používají, často zajímá jen určitá oblast, ostatní je bráno jako pozadí. Mohlo by proto být přínosné v průběhu skenování určit, které oblasti jsou zajímavé a které ne, aby se množství skenovaných pixelů snížilo pouze na ty, které přinášejí nějakou důležitou informaci. Výrazné změny v obrázku – hrany – se dají za tuto důležitou informaci považovat.

V této kapitole bude nejprve model z předchozí kapitoly rozšířen tak, aby se dal použít pro odhad změn v obraze. Následují experimenty, které ukážou, jak se model chová v případě, že v naměřeném obraze chybí určité pixely. Pro tyto testy budou použity dva algoritmy z předchozí kapitoly: IVB-full a ELBO-Vadam, jejichž schopnost pracovat s nekompletním obrazem bude vzájemně porovnávána. Na konci kapitoly budou pak nová zjištění aplikována na snímek z environmentálního rastrovacího elektronového mikroskopu.

### 4.1 Vícerozměrná variance šumu

Pro lepší odhad aposteriorní variance v ostrém obraze budeme předpokládat, že naměřené pixely mají různou přesnost, tzn. přesnost  $\omega$  nebude skalární, ale vektorová  $\boldsymbol{\omega}$  a rozdělení dat pak bude následující

$$\mathbf{d}|\mathbf{k}, \mathbf{x}, \boldsymbol{\omega} \sim \mathcal{N}(\mathbf{K}\mathbf{x}, \text{diag}(\boldsymbol{\omega})^{-1}). \quad (4.1)$$

Jako apriorní rozdělení  $\boldsymbol{\omega}$  opět zvolíme gama-rozdělení, tedy

$$\omega_i \sim \mathcal{G}(\gamma_0, \eta_0), \quad \forall i \in \widehat{np}.$$

Změny ve srovnání s modelem v kapitole 2 jsou pouze ve třech aposteriorních rozděleních – v rozdělení ostrého obrazu  $\mathbf{x}$ , v rozdělení PSF  $\mathbf{k}$  a v rozdělení samotné přesnosti šumu  $\boldsymbol{\omega}$ .

Pro odvození aposteriorního rozdělení  $\mathbf{x}$  je potřeba rozepsat vícerozměrné normální rozdělení (4.1) jako součin jednorozměrných, což je možné díky tomu, že kovarianční matice 4.1 je diagonální a tedy jsou jednotlivé pixely nezávislé náhodné proměnné. Logaritmus součinu hustot těchto rozdělení společně s logaritmem hustoty apriorního rozdělení  $\mathbf{x}$  má tvar

$$\begin{aligned} L_x &= -\frac{1}{2} \sum_{i \in \widehat{np}} (d_i - \bar{\mathbf{K}}_i \mathbf{x})^T \omega_i (d_i - \bar{\mathbf{K}}_i \mathbf{x}) - \frac{1}{2} \mathbf{x}^T (\nabla^T \text{diag}(\boldsymbol{\tau}_x) \nabla) \mathbf{x} = \\ &= -\frac{1}{2} \left( \sum_{i \in \widehat{np}} \omega_i d_i^2 - 2 \sum_{i \in \widehat{np}} \omega_i d_i \bar{\mathbf{K}}_i \mathbf{x} + \sum_{i \in \widehat{np}} \omega_i \mathbf{x}^T \bar{\mathbf{K}}_i^T \bar{\mathbf{K}}_i \mathbf{x} \right) - \frac{1}{2} \mathbf{x}^T (\nabla^T \text{diag}(\boldsymbol{\tau}_x) \nabla) \mathbf{x}. \end{aligned}$$

kde  $d_i$  je  $i$ -tý prvek  $\mathbf{d}$  a  $\bar{\mathbf{K}}_i$  značí  $i$ -tý řádek matice  $\mathbf{K}$  (řádkový vektor). Podobně jako v kapitole 2 dostaneme, že aproximace aposterioriho rozdělení je normální rozdělení se střední hodnotou  $\boldsymbol{\mu}_x$  a kovariancí  $\boldsymbol{\Sigma}_x$

$$\mathbf{x}|\mathbf{d} \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x),$$

kde

$$\boldsymbol{\Sigma}_x = \left( \sum_{i \in \hat{n}p} \hat{\omega}_i \bar{\mathbf{K}}_i^T \bar{\mathbf{K}}_i + \boldsymbol{\nabla}^T \text{diag}(\hat{\boldsymbol{\tau}}_x) \boldsymbol{\nabla} \right)^{-1},$$

$$\boldsymbol{\mu}_x = \boldsymbol{\Sigma}_x \left( \sum_{i \in \hat{n}p} d_i \hat{\omega}_i \bar{\mathbf{K}}_i^T \right),$$

a  $\bar{\mathbf{X}}_i$  značí  $i$ -tý řádek  $\mathbf{X}$  (řádkový vektor) a štríška střední hodnotu vzhledem k aproximaci aposterioriho rozdělení odpovídající veličiny.

Analogicky lze odvodit i rozdělení  $\mathbf{k}$

$$\mathbf{k}|\mathbf{d} \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

$$\boldsymbol{\Sigma}_k = \left( \sum_{i \in \hat{n}p} \hat{\omega}_i \bar{\mathbf{X}}_i^T \bar{\mathbf{X}}_i + \hat{\tau}_k \mathbb{I} \right)^{-1},$$

$$\boldsymbol{\mu}_k = \boldsymbol{\Sigma}_k \left( \sum_{i \in \hat{n}p} d_i \hat{\omega}_i \bar{\mathbf{X}}_i^T \right),$$

Logaritmus součinu rozdělení obsahujících  $\boldsymbol{\omega}$  je následující

$$L_\omega = \frac{1}{2} \sum_{i \in \hat{n}p} \ln \omega_i - \frac{1}{2} \sum_{i \in \hat{n}p} (d_i - \bar{\mathbf{K}}_i \mathbf{x})^T \omega_i (d_i - \bar{\mathbf{K}}_i \mathbf{x}) + \sum_{i \in \hat{n}p} ((\gamma_0 - 1) \ln \omega_i - \eta_0 \omega_i),$$

takže

$$\frac{\partial L_\omega}{\partial \omega_i} = \frac{1}{2} \frac{1}{\omega_i} - \frac{1}{2} (d_i - \bar{\mathbf{K}}_i \mathbf{x})^T (d_i - \bar{\mathbf{K}}_i \mathbf{x}) + (\gamma_0 - 1) \frac{1}{\omega_i} - \eta_0.$$

MAP odhad  $\boldsymbol{\omega}$  je proto dán

$$\hat{\boldsymbol{\omega}} = \left( \gamma_0 - \frac{1}{2} \right) \left( \frac{1}{2} \mathbf{d} \circ \mathbf{d} - \widehat{\mathbf{K}} \hat{\mathbf{x}} \circ \mathbf{d} + \frac{1}{2} \mathbb{E}_{\hat{f}} \left[ \sum_{i \in \hat{n}p} \mathbf{x}^T \bar{\mathbf{K}}_i^T \bar{\mathbf{K}}_i \mathbf{x} \right] + \eta_0 \right)^{-1},$$

$\circ$  je symbol pro Hadamardův součin a  $\gamma_0$  musí být větší než jedna polovina, jinak je maxima dosaženo v nule.

## 4.2 Odvození pro ELBO

V rámci algoritmu ELBO-Vadam se provádí derivace  $-\ln p(\mathbf{d}|\mathbf{x}, \mathbf{k}, \boldsymbol{\omega})$ , což má tvar

$$-\frac{1}{2} \sum_{i \in \hat{n}p} \ln \hat{\omega}_i + \frac{1}{2} \sum_{i \in \hat{n}p} \mathbb{E}_{\hat{f}} \left[ (d_i - \bar{\mathbf{X}}_i \mathbf{k})^T \omega_i (d_i - \bar{\mathbf{X}}_i \mathbf{k}) \right] =$$

$$= -\frac{1}{2} \sum_{i \in \hat{n}p} \ln \hat{\omega}_i + \frac{1}{2} \sum_{i \in \hat{n}p} \hat{\omega}_i \left( \mathbf{d}_i^2 - 2 \widehat{\mathbf{X}}_i \mathbf{k} d_i + \mathbf{k}^T \bar{\mathbf{X}}_i^T \bar{\mathbf{X}}_i \mathbf{k} \right).$$



Derivace vzhledem k  $\mathbf{k}$  je nalezena pomocí automatické diferenciaci v jazyce julia a je vyčíslena v bodě  $\boldsymbol{\mu}_k + \mathbf{S}_k \boldsymbol{\epsilon}$ , kde  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ . Momenty se určí stejně jako v předchozí kapitole.

### 4.3 Chybějící pixely

V případě, kdy v obraze některé pixely chybí a jejich poloha je známá, vypadá odvození velmi podobně. Označíme-li vektor naměřených pixelů o velikosti  $u < np$  jako  $\tilde{\mathbf{d}}$ , datový člen potom sestává pouze z těchto  $u$  pixelů, tzn. z  $\mathbf{K}$  se použije pouze odpovídajících  $u$  řádků a  $\boldsymbol{\omega}$  má  $u$  prvků. Vypadá pak následovně

$$\frac{1}{2} \sum_{i \in \hat{u}} \ln \omega_i - \frac{1}{2} \sum_{i \in \hat{u}} (\tilde{d}_i - \bar{\mathbf{K}}_i \mathbf{x})^T \omega_i (\tilde{d}_i - \bar{\mathbf{K}}_i \mathbf{x}).$$

Odhady parametrů odvozené v předchozí sekci se změní pouze nahrazením  $\mathbf{d}$  nekompletním obrazem  $\tilde{\mathbf{d}}$  a v množině indexů, přes které sčítají sumy. Místo všech  $np$  pixelů obrazu se sčítá jen přes  $u$  naměřených pixelů. Parametry aposteriorních rozdělání jsou pak následující

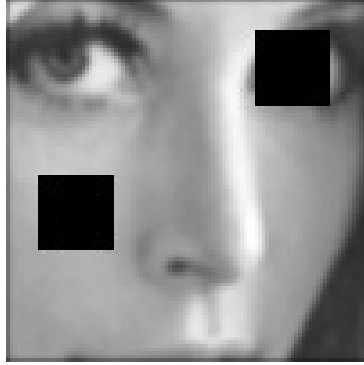
$$\begin{aligned} \boldsymbol{\Sigma}_x &= \left( \sum_{i \in \hat{u}} \hat{\omega}_i \widehat{\bar{\mathbf{K}}}_i^T \widehat{\bar{\mathbf{K}}}_i + \boldsymbol{\nabla}^T \text{diag}(\hat{\boldsymbol{\tau}}_x) \boldsymbol{\nabla} \right)^{-1}, \\ \boldsymbol{\mu}_x &= \boldsymbol{\Sigma}_x \left( \sum_{i \in \hat{u}} \tilde{d}_i \hat{\omega}_i \widehat{\bar{\mathbf{K}}}_i^T \right), \\ \boldsymbol{\Sigma}_k &= \left( \sum_{i \in \hat{u}} \hat{\omega}_i \widehat{\bar{\mathbf{X}}}_i^T \widehat{\bar{\mathbf{X}}}_i + \hat{\tau}_k \mathbb{I} \right)^{-1}, \\ \boldsymbol{\mu}_k &= \boldsymbol{\Sigma}_k \left( \sum_{i \in \hat{u}} \tilde{d}_i \hat{\omega}_i \widehat{\bar{\mathbf{X}}}_i^T \right), \\ \hat{\omega} &= \left( \gamma_0 - \frac{1}{2} \right) \left( \frac{1}{2} \tilde{\mathbf{d}} \circ \tilde{\mathbf{d}} - \widehat{\bar{\mathbf{K}}} \hat{\mathbf{x}} \circ \tilde{\mathbf{d}} + \frac{1}{2} \mathbb{E}_{\tilde{\mathbf{f}}} \left[ \sum_{i \in \hat{u}} \mathbf{x}^T \bar{\mathbf{K}}_i^T \bar{\mathbf{K}}_i \mathbf{x} \right] + \eta_0 \right)^{-1}. \end{aligned}$$

Tvary momentů se určí stejným způsobem jako v předchozí kapitole.

### 4.4 Odhad přesnosti v chybějících oblastech

Nejprve budou algoritmy IVB-full a ELBO-Vadam testovány na obrázku, ve kterém jsou vymazané dvě oblasti. Jedná se o výřez z Leny [2], který byl rozmázan gaussianovskou PSF a byl k němu přidán šum o SNR 40 dB, na obrázku 4.1. Obě oblasti mají rozměr  $20 \times 20$  pixelů, takže při velikosti PSF  $5 \times 5$  by vnitřek těchto oblastí měl být pro algoritmy naprosto neznámý. První výřez je umístěn v oblasti levého oka Leny, kde se vyskytuje hodně hran a měla by to tedy být zajímavá oblast obrázku. Druhý výřez je naproti tomu umístěn na pravé tváři, kde jsou rozdíly mezi pixely minimální. Sledován bude odhad aposteriorní přesnosti (variance) ostrého obrázku, tedy  $\boldsymbol{\Sigma}_x$ , která je brána jako diagonální, takže přesnosti jednotlivých pixelů jsou přímo převrácené hodnoty variance.

Počáteční hodnoty musely být změněny. Parametry apriorního rozdělání  $\boldsymbol{\omega}$  musely být nastaveny tak, aby aposteriorní gama-rozdělání nemělo maximum v nule. Dále byla zvýšena střední



Obrázek 4.1: Výřez z Leny rozmazaný gaussovskou PSF, kde SNR přidaného šumu je 40dB. Oblasti o rozměru  $20 \times 20$  pixelů na levém oku a pravé tváři byly odstraněny kvůli zkoumání variance ostrého obrazu.

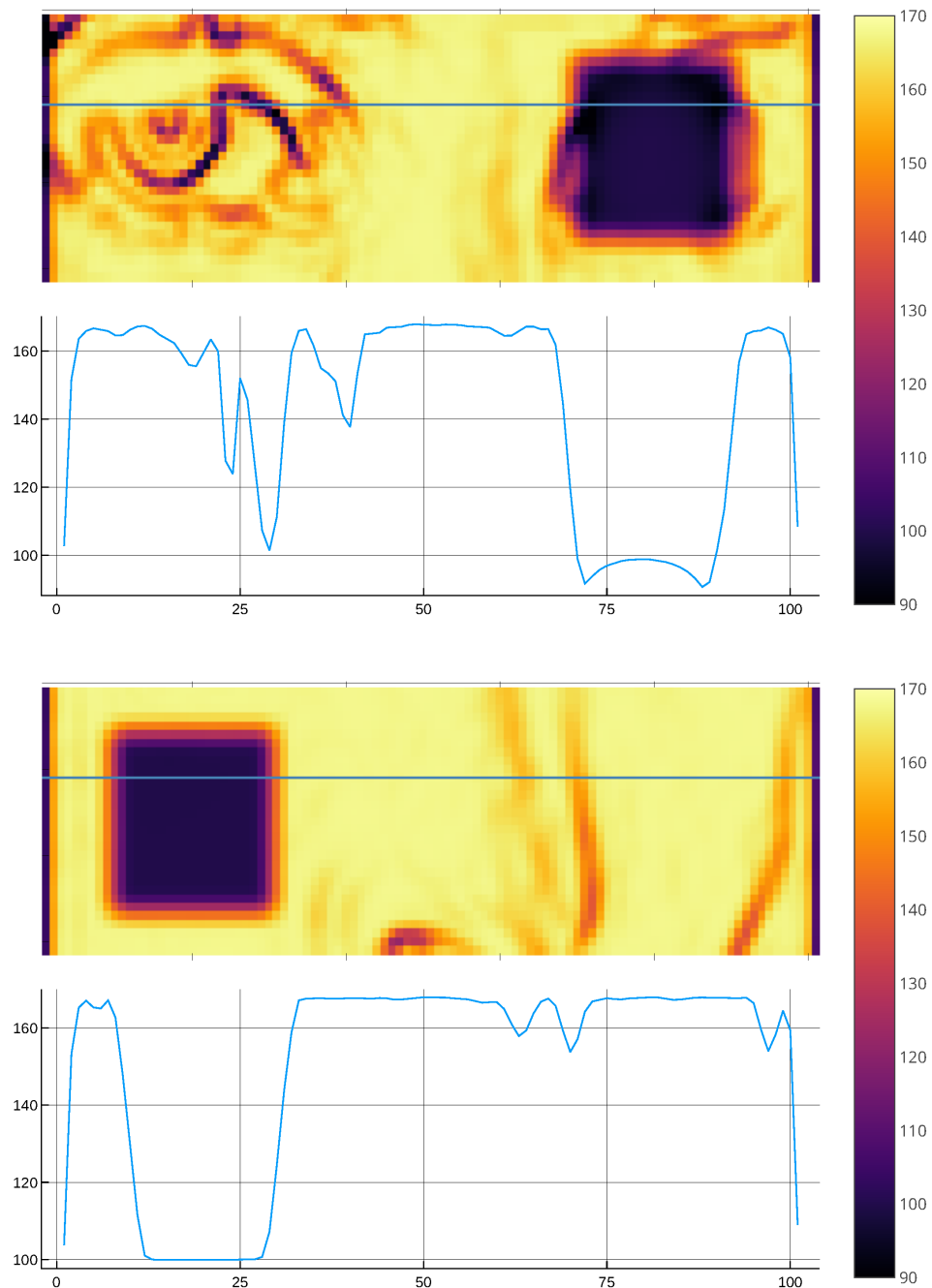
hodnota apriorní variance na derivacích ostrého obrazu a to z  $10^{-5}$  na  $10^{-2}$  kvůli ostrým přechodům způsobeným vymazanými pixely. Počáteční odhad variance derivací na ostrém obraze byl zvýšen a počáteční odhad variance na PSF snížen. Tyto počáteční hodnoty jsou vypsány v tabulce 4.1.

$\Sigma_x$	$\Sigma_k$	$\tau_x$	$\tau_k$	$\alpha_{X0}$	$\beta_{X0}$	$\alpha_{k0}$	$\beta_{k0}$	$\omega$	$\gamma_0$	$\eta_0$
$10^{-3}\mathbb{I}$	$10^{-4}\mathbb{I}$	$10^3\mathbf{1}$	$10^2$	$10^0$	$10^{-2}$	$10^0$	$10^{-2}$	$10^0$	$10^1$	$10^{-2}$

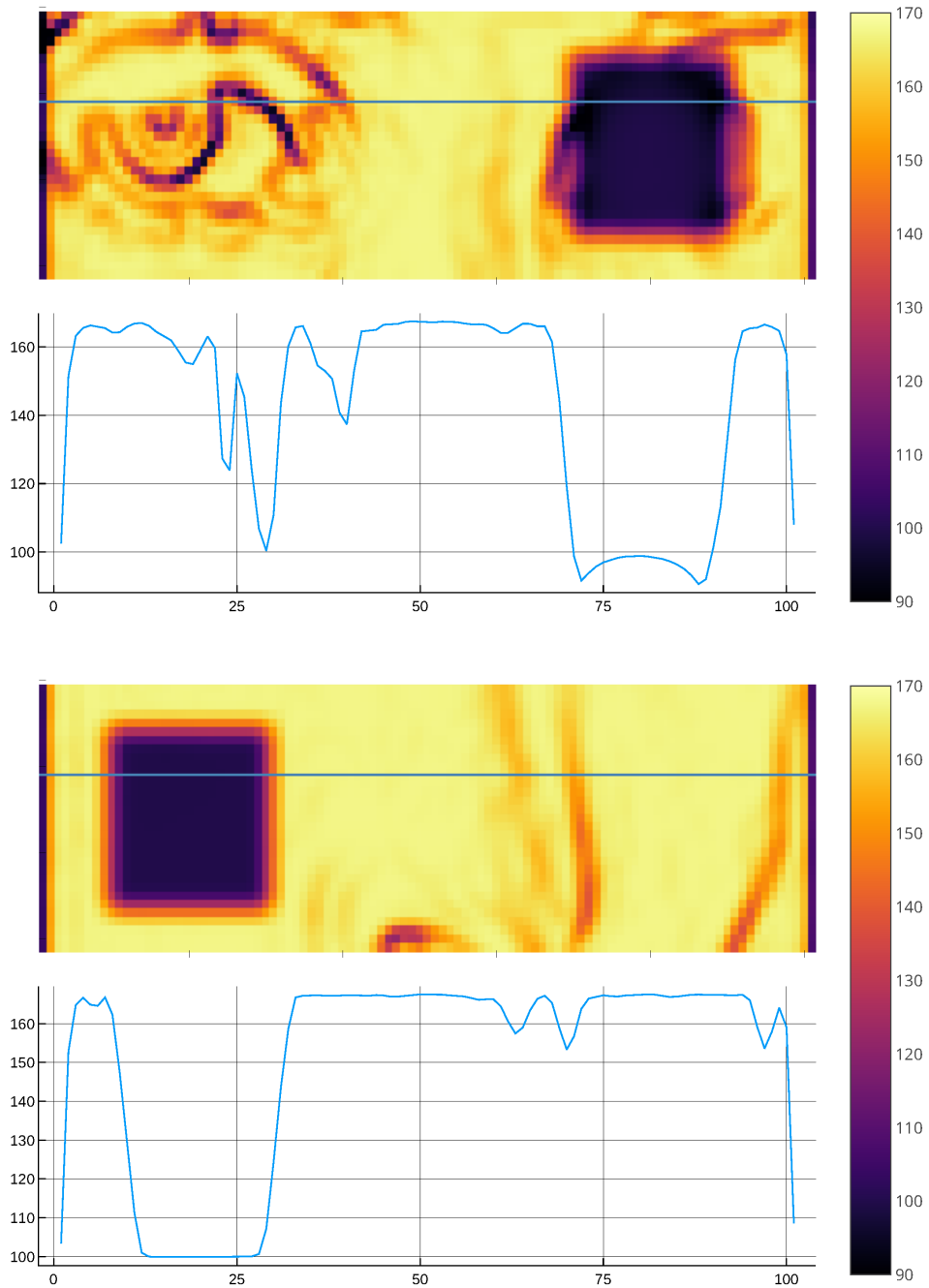
Tabulka 4.1: Počáteční hodnoty skrytých proměnných a parametrů modelu pro odhad variance chybějících pixelů.  $\mathbb{I}$  značí jednotkovou matici a  $\mathbf{1}$  vektor jedniček.

Na obrázku 4.2 je odhad aposteriorní přesnosti ostrého obrazu (tj. převrácené hodnoty diagonály  $\Sigma_x$ ) pomocí algoritmu IVB-full. Z obrázků je vidět, že v případě, kdy v okolí vymazané oblasti nejsou velké změny v intenzitě, aposteriorní přesnost pixelů se se vzdáleností od pixelů, které jsou známé, snižuje až dosáhne konstantní hodnoty uvnitř oblasti. Pokud jsou v okolí chybějící oblasti změny, projeví se hodnotou přesnosti v uvnitř této oblasti. Z přesnosti výřezu na oku je patrný obrys oka a má nižší přesnost, než jaká se objevuje ve výřezu z tváře a v části oblasti na oku, o které okolní hrany žádnou informaci neposkytují. Tento rozdíl je nejlépe vidět na grafech řezu obrázkem, v případě, kdy v okolí žádné změny nejsou, přesnost skoro lineárně klesne k hodnotě 100 a zůstane na této hodnotě pro většinu pixelů ve vymazané oblasti (pixely cca 10 až 30). U řezu v oblasti oka přesnost uvnitř čtverce s chybějícími pixely nejdříve poklesne pod hodnotu 100, následně po konkávní křivce vzroste ke stovce a s blížícím se okrajem chybějící oblasti zase klesá (pixely cca 70 až 90). Stejné chování lze pozorovat i na odhadu získaném pomocí algoritmu ELBO-Vadam, jehož výsledky jsou na obrázku 4.3.

Tyto výsledky naznačují, že by pomocí VB i kombinací VB a maximalizace ELBO mělo být možné stanovit, kde se v obrázku objevují změny, tedy potenciálně zajímavé oblasti, a kde by bylo přínosné doměřit další pixely.



Obrázek 4.2: Obrázky ukazují odhad přesnosti ve dvou vymazaných oblastech algoritmem IVB-full. Na teplotní mapě je výřez z aposteriori přesnosti, modrá linka značí řez, který je zobrazený pod teplotní mapou. Na vertikální ose řezu jsou hodnoty intenzity, na horizontální pozice pixelu v řezu.



Obrázek 4.3: Obrázky ukazují odhad přesnosti ve dvou vymazaných oblastech algoritmem ELBO-Vadam. Na teplotní mapě je výřez z aposteriorní přesnosti, modrá linka značí řez, který je zobrazený pod teplotní mapou. Na vertikální ose řezu jsou hodnoty intenzity, na horizontální pozice pixelu v řezu.

## 4.5 Míra poškození

Dále by bylo dobré znát, jak velké množství pixelů může ve snímku chybět, aby byl odhad stále dobrý. Testy budou opět provedeny na výřezu z Leny s gaussovskou PSF a SNR 40dB. Tentokrát nebudou vymazány velké oblasti, ale náhodné pixely tak, aby chybělo 10%, 50%, 80%, 90%, 95%, 99% pixelů v obraze. Na každé úrovni bude schopnost odhadu měřena pomocí PSNR přímého odhadu a neslepé dekonvoluce obrazu bez chybějících pixelů a odhadu PSF z algoritmu. Bylo zvoleno pouze gaussovské rozostření a rozmazání lineárním pohybem, neboť odhady špatného zaostření jsou velmi podobné gaussovské PSF, jak bylo ukázáno v předchozí kapitole.

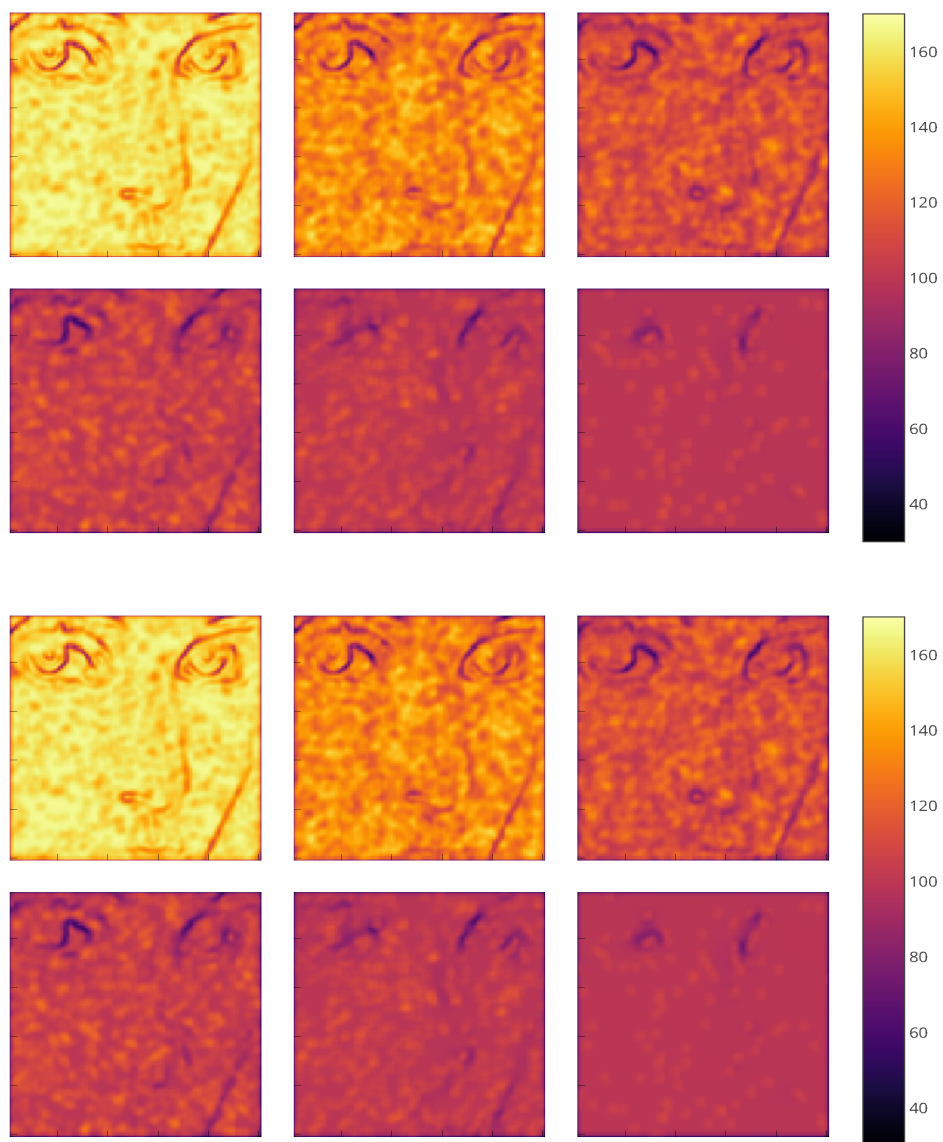
V tabulce 4.2 jsou vypsány průměrné hodnoty PSNR, při pěti různých náhodně zvolených rozloženích chybějících pixelů v obraze. Z hodnot pro neslepou dekonvoluci je vidět, že i když jsou výsledky pro všechny míry chybějících pixelů horší než ty, kdy je obraz plný, vždy odhad převyšuje PSNR poškozeného obrazu. Tato hodnota mluví spíše o kvalitě odhadu PSF, narozdíl od přímého odhadu, jehož hodnoty PSNR tak dobré nejsou, pro gaussovskou PSF je odhad horší než rozmazaný obraz při 95% a 99% chybějících pixelů. Přímý odhad pro rozmazání pohybem je lepší pro 95% a více chybějících pixelů.

Kromě průměrných PSNR jsou v tabulce i jejich směrodatné odchylky. Pro oba dva algoritmy jsou jejich hodnoty prakticky stejné, takže generování náhodných čísel v rámci algoritmu ELBO-Vadam nepřináší významnější rozdíly.

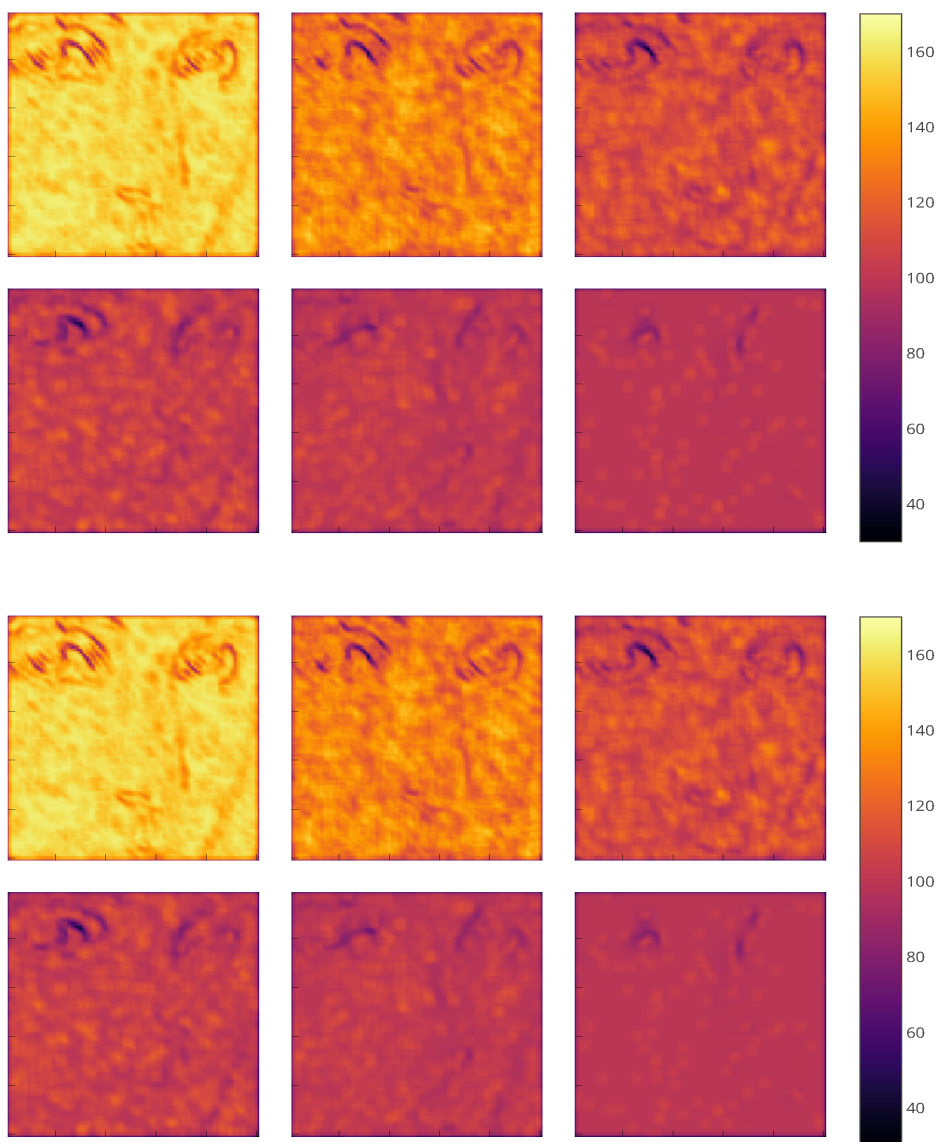
Na obrázku 4.4 jsou vykresleny aposteriorní přesnosti při volbě gaussovské PSF a na obrázku 4.5 pro rozmazání lineárním pohybem. Při 99% chybějících pixelů už není okem patrný nos ani okraj tváře. Ty se objevují už při 95% chybějících pixelů, při 90% jsou ještě o něco výraznější. Přímý odhad obrazu je více spjatý s hodnotami aposteriorní přesnosti obrazu, proto se nastavení hodnoty potřebných pixelů bude zakládat na jeho PSNR. Minimální počet pixelů bude stanoven na 10% pro obrázek s nízkou mírou šumu (test je proveden na obrázku se SNR 40dB). U obou testovaných PSF při tomto počtu pixelů průměr PSNR přesahuje PSNR rozmazaného obrazu. Na obrázcích s přesností jsou vidět významné oblasti.

Algoritmus	PSF	chybí	$\mu_x$	std $\mu_x$	dek.	std dek
VADAM	pohyb	10%	27.62108	0.029	24.24657	0.00088
VADAM	pohyb	50%	27.90389	0.14072	24.23882	0.00822
VADAM	pohyb	80%	27.42076	0.20723	24.22348	0.02131
VADAM	pohyb	90%	26.04978	0.13793	24.19906	0.03477
VADAM	pohyb	95%	23.90078	0.21392	24.15447	0.02527
VADAM	pohyb	99%	18.31374	0.43294	24.09717	0.03172
VADAM	gauss	10%	29.02108	0.02953	31.01145	0.02223
VADAM	gauss	50%	29.49296	0.07948	30.86288	0.04349
VADAM	gauss	80%	28.57507	0.10868	30.57731	0.05985
VADAM	gauss	90%	26.40205	0.18776	30.14039	0.03301
VADAM	gauss	95%	23.87021	0.02042	30.38016	0.08176
VADAM	gauss	99%	18.17995	0.44693	30.05169	0.26691
IVB-full	pohyb	10%	27.64024	0.02585	24.24398	0.00094
IVB-full	pohyb	50%	27.91739	0.13929	24.23786	0.00817
IVB-full	pohyb	80%	27.4233	0.20718	24.22187	0.02029
IVB-full	pohyb	90%	26.04762	0.13792	24.19742	0.03497
IVB-full	pohyb	95%	23.89391	0.21644	24.1499	0.0277
IVB-full	pohyb	99%	18.30106	0.43621	24.07951	0.04097
IVB-full	gauss	10%	29.09813	0.02463	30.99969	0.02302
IVB-full	gauss	50%	29.52675	0.08161	30.86347	0.04231
IVB-full	gauss	80%	28.59345	0.10585	30.57092	0.056
IVB-full	gauss	90%	26.41262	0.19015	30.12758	0.02689
IVB-full	gauss	95%	23.87549	0.02099	30.37096	0.07756
IVB-full	gauss	99%	18.17474	0.44855	30.02845	0.32455

Tabulka 4.2: Tabulka průměrných hodnot PSNR pro počet 5 různých náhodných pozic chybějících pixelů. Sloupec chybí obsahuje procento chybějících pixelů, sloupec  $\mu_x$  obsahuje průměr PSNR přímého odhadu, std  $\mu_x$  směrodatnou odchylku těchto měření. Sloupec dek. obsahuje průměr PSNR neslepé dekonvoluce, std dek. pak směrodatnou odchylku tohoto průměru.



Obrázek 4.4: Teplotní mapy aposterioriálních přesností na ostrém obraze při různé míře náhodného poškození a gaussovské PSF. Horních 6 map je výstup z algoritmu IVB-full, dolních 6 z algoritmu ELBO-Vadam. V každé šestici chybí na prvním řádku po řadě 10%, 50% a 80% pixelů, na druhém 90%, 95%, 99%.

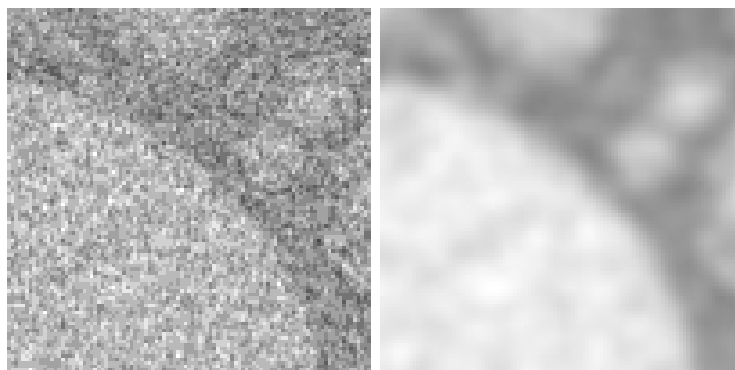


Obrázek 4.5: Teplotní mapy aposteriorních přesností na ostrém obraze při různé míře náhodného poškození a rozmazáním lineárním pohybem. Horních 6 map je výstup z algoritmu IVB-full, dolních 6 z algoritmu ELBO-Vadam. V každé šestici chybí na prvním řádku po řadě 10%, 50% a 80% pixelů, na druhém 90%, 95%, 99%.



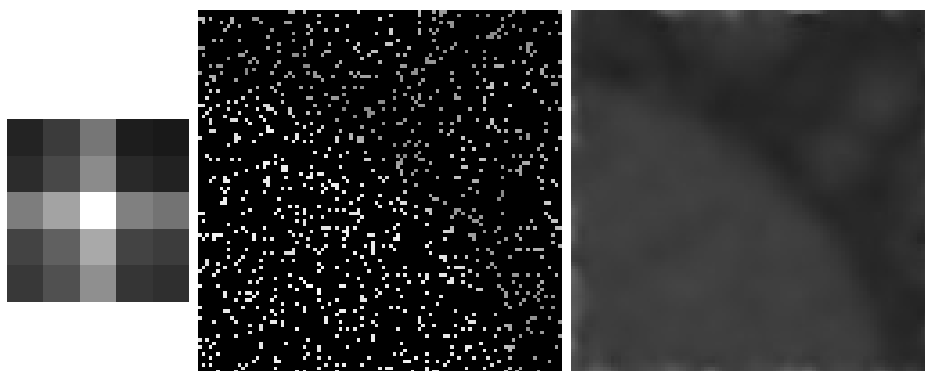
## 4.6 Snímek z mikroskopu

Navržený model bude otestován na odhadu z nekompletního obrázku z environmentálního rastrovacího elektronového mikroskopu, který laskavě poskytl Ing. Vilém Neděla, Ph.D. z Ústavu přístrojové techniky AV v Brně. Výřez ze snímku je na obrázku 4.6 vpravo. Pixely na výřezu z originálního snímku mají pouze 9 různých hodnot a hrany nejsou příliš výrazné, proto byl zvýšen kontrast a snímek byl následně rozmazán gaussovským filtrem, výsledek tohoto předzpracování je na obrázku 4.6 vlevo.



Obrázek 4.6: Na levém obrázku je výřez z originálního snímku z rastrovacího elektronového mikroskopu. Na pravém obrázku je výřez ze snímku, u kterého byl zvýšen kontrast a byl rozostřen kvůli snížení šumu.

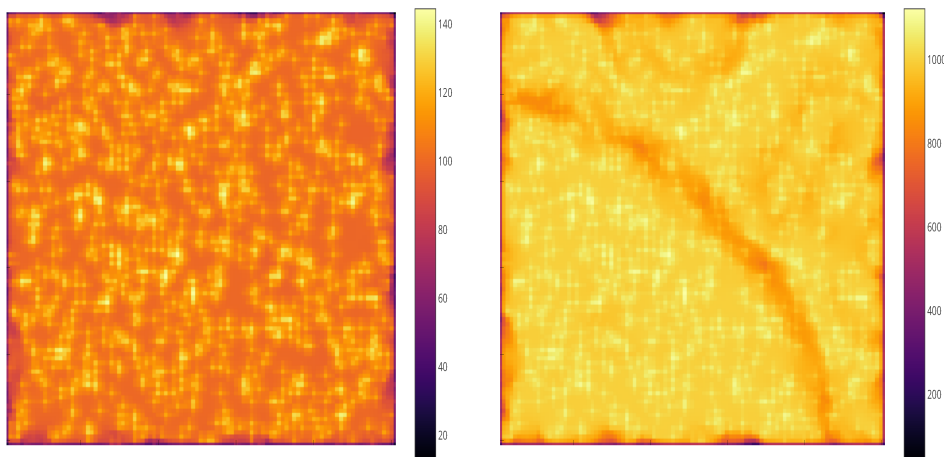
Na základě analýzy z předchozí sekce, bylo z obrázku náhodně odstraněno 90% pixelů. Odhad byl proveden pomocí algoritmu ELBO-Vadam. Počáteční hodnoty byly nastaveny podle tabulky 4.1. Odhad PSF a obrazu je společně s nekompletním snímekem ukázán na obrázku 4.7. Odhad PSF připomíná gaussovskou PSF, neboť při předzpracování byl použit gaussovský filtr. V odhadu obrázku jsou patrné obrysy z originálního obrázku. Pixel v pravém horním rohu se odhadl s vysokou hodnotou, proto působí tmavě.



Obrázek 4.7: Zprava odhad PSF, vstupní obrázek s chybějícími pixely a odhad obrázku při původním nastavení počátečních hodnot (tabulka 4.1).

Na obrázku 4.8 vlevo je vykreslena aposteriorní přesnost odhadu obrázku. Hrany z původního obrázku tam vůbec nejsou patrné, a to ani v případě, kdy je neznámých pixelů výrazně méně. Proto byly změněny parametry apriorního rozdělení přesností na derivacích obrázku, parametr  $\beta_{X_0}$  byl nastaven na hodnotu  $10^{-3}$ , tedy apriorní střední hodnota prvků  $\tau_x$  se zvýšila z  $10^2$  na

$10^3$  a zvýšila se i jejich variance. Na obrázku se proto snížila střední hodnota variance rozdílů sousedních pixelů a je hledán obraz s většími hladkými plochami. Takto upravené počáteční hodnoty jsou vypsány v tabulce 4.3.

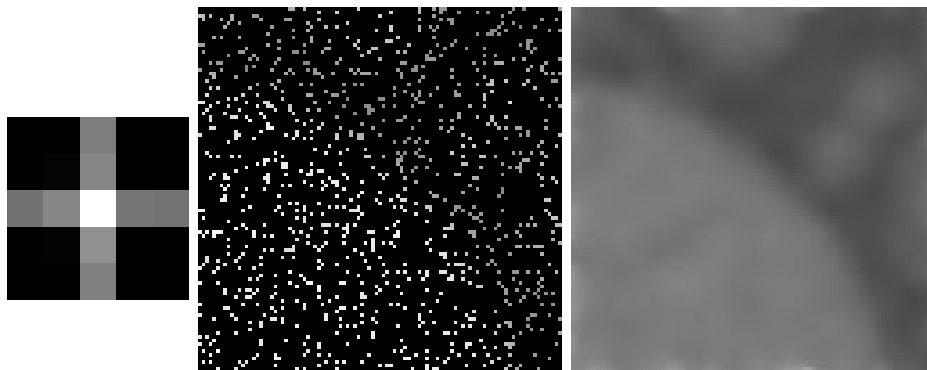


Obrázek 4.8: Na levém obrázku je teplotní mapa aposteriorní přesnosti odhadu z původních počátečních hodnot (tabulka 4.1). Na pravém z nových počátečních hodnot (tabulka 4.3).

$\Sigma_x$	$\Sigma_k$	$\tau_x$	$\tau_k$	$\alpha_{X0}$	$\beta_{X0}$	$\alpha_{k0}$	$\beta_{k0}$	$\omega$	$\gamma_0$	$\eta_0$
$10^{-3}\mathbb{I}$	$10^{-4}\mathbb{I}$	$10^3\mathbf{1}$	$10^2$	$10^0$	$10^{-3}$	$10^0$	$10^{-2}$	$10^0$	$10^1$	$10^{-2}$

Tabulka 4.3: Počáteční hodnoty skrytých proměnných a parametrů modelu pro odhad variance chybějících pixelů pro snímky z mikroskopu.  $\mathbb{I}$  značí jednotkovou matici a  $\mathbf{1}$  vektor jedniček.

Výsledky pro nové počáteční hodnoty jsou na obrázku 4.9. Odhad PSF je v tomto případě symetrický, ale už tolik nepřipomíná gaussovskou PSF. To nemusí být špatně, neboť PSF původního obrázku není známá. Obrázek je odhadnut prakticky stejně, jen s vyšším kontrastem mezi světlými a tmavými oblastmi obrázku. Na obrázku 4.8 je vykreslena teplotní mapa aposteriorní přesnosti obrázku. V tomto případě je už v odhadu velmi dobře vidět nejvýraznější hrana a i některé méně výrazné.



Obrázek 4.9: Zprava odhad PSF, vstupní obrázek s chybějícími pixely a odhad obrázku při novém nastavení počátečních hodnot (tabulka 4.3).



# Závěr

Cílem diplomové práce bylo nalézt algoritmus pro slepou dekonvoluci, který by byl méně restriktivní vůči volbě pravděpodobnostního rozdělení rozmazání obrazu. Toto by bylo přínosné například v případech, kdy PSF závisí na poloze pixelu v obrazu. V environmentální rastrovací elektronové mikroskopii se obraz měří vysíláním paprsku pixel po pixelu, ale zdaleka ne všechny pixely obrázku jsou pro vědce zajímavé. Proto byla dále zkoumána schopnost takto nalezeného algoritmu odhadnout, které oblasti v nekompletním obraze by byly potenciálně zajímavé a bylo by přínosné je doměřit.

V první kapitole byl popsán model rozmazaného snímku a metody, které se pro řešení slepé dekonvoluce používají. Důraz byl kladen na pravděpodobnostní modely a volby rozdělení jednotlivých proměnných modelu – ostrého snímku a PSF.

V druhé kapitole byla hlouběji představena variační Bayesova metoda a algoritmy, které se používají k hledání odhadů. Tyto algoritmy byly použity již dříve, jsou považovány za spolehlivé a rychlé, ale jejich použití předpokládá, že apriorní rozdělení parametrů budou volena z konjugovaného systému hustot, aby bylo možné pouze řešit lineární soustavu rovnic. Tento požadavek je omezující, proto byla teorie variační Bayesovy metody propojena s odhadem přes ELBO, které díky optimalizaci pomocí gradientního sestupu dává větší volnost pro volbu apriorních rozdělení. Následně byl navržen model pro řešení slepé dekonvoluce obrazu. Byla zvolena taková apriorní rozdělení, aby bylo možné řešit úlohu pomocí variační Bayesovy metody i přes ELBO a porovnat obě řešení. Apriorní rozdělení ostrého obrazu bylo zvoleno tak, aby rozdíly sousedních horizontálních a vertikálních pixelů byly řídké, tedy ostrý obraz by měl být po částech hladký.

Třetí kapitola obsahuje návrhy algoritmu s optimalizací přes ELBO a srovnání jejich výsledků na testovacích datech s výsledky algoritmu, který kompletně stojí na variační Bayesově metodě. V rámci prvního navrženého algoritmu byl použit reparametrizační trik pro odhad některých středních hodnot a to s využitím pouze jednoho vzorku náhodné proměnné z reparametrizace. To se ukázalo být dostatečné při aproximaci aposteriorní kovarianční matice PSF diagonálou, ale ne pro plnou kovarianci. Při diagonální kovarianci PSF byly odhady srovnatelné s výsledky VB, ale bez této aproximace to neplatilo, proto byl navržen takový tvar kovarianční matice, který zachovával pouze některé prvky. Odhad byl sice lepší, ale zdaleka ne tak dobrý, jaký byl nalezen VB s plnou kovariancí PSF. Jeden vzorek v rámci reparametrizačního triku nemusí být dostatečný pro odhad plné kovarianční matice, proto jich v dalším experimentu bylo bráno 25. Ačkoliv se odhad vylepšil, tento přístup byl zamítnut protože je výpočetně velmi náročný. Poslední navržená varianta algoritmu s odhadem přes ELBO využívala optimalizátor Vadam namísto klasického optimalizátoru Adam. Tento algoritmus dosáhl přesnosti variační Bayesovy metody s plnou kovarianční maticí, ačkoliv sám uvažuje pouze diagonální, a z testovaných algoritmů používajících ELBO konvergoval nejrychleji. Tato srovnání byla provedena za předpokladu znalosti nosiče PSF, proto byl otestován i případ, kdy velikost nosiče známá není. Algoritmus s optimalizátorem Vadam dosáhl stejné přesnosti jako ten založený variační Bayesově metodě s plnou kovariancí

PSF.

Na závěr byla zkoumána schopnost nejlepších algoritmů odhadnout z nekompletního obrazu, které oblasti by mohly být potenciálně zajímavé. Odhady algoritmu s optimalizátorem Vadam a algoritmu založeném plně na variační Bayesově byly v této kapitole opět prakticky stejné. Na testech se dvěma chybějícími oblastmi, jednou s malými změnami a jednou s výraznými změnami, se ukázalo, že algoritmy rozpoznají, která oblast má větší varianci a je tedy potenciálně zajímavější. Následně byl na základě experimentů s různým množstvím chybějících pixelů v obraze stanoven minimální počet pixelů pro dobrý odhad variance v obraze a tento výsledek byl aplikován na snímek z environmentální rastrovací elektronové mikroskopie. Ukázalo se, že pro dobrý odhad variance je nutné citlivě nastavit parametry apriorního rozdělení přesnosti na derivacích ostrého obrazu. Už u řešení problému s neznámým nosičem PSF bylo potřeba tyto parametry upravit, proto by mohlo být do budoucna přínosné odhadovat i parametry tohoto apriorního rozdělení.

# Literatura

- [1] Environmentální rastrovací elektronová mikroskopie a detekční systémy [online]. Dostupné z: <https://www.isibrno.cz/cs/environmentalni-rastrovaci-elektronova-mikroskopie-detekcni-systemy>
- [2] Lena [online]. Dostupné z: <https://www.cosy.sbg.ac.at/~pmeerw/Watermarking/lena.html>, cited 2020-10-12
- [3] AGARWAL, Chirag a et al., DEEP-URL: A Model-Aware Approach to Blind Deconvolution Based on Deep Unfolded Richardson-Lucy Network. In '2020 IEEE International Conference on Image Processing (ICIP)', IEEE, 2020, 3299-3303.
- [4] ATTIAS, Hagai, A Variational Bayesian framework for graphical models. *Advances in Neural Information Processing Systems* 12, 2000, 209-215.
- [5] BABACAN, S. Derin, Rafael MOLINA a Aggelos K. KATSAGGELOS, Variational Bayesian Blind Deconvolution Using a Total Variation Prior. *IEEE Transactions on Image Processing* 18(1), 2009, 12-26.
- [6] BABACAN, S. D., R. MOLINA, M. N. DO a A. K. KATSAGGELOS, Bayesian Blind Deconvolution with General Sparse Image Priors. In 'ECCV'12: Proceedings of the 12th European conference on Computer Vision - Volume Part VI', Springer-Verlag, Berlin, Heidelberg, 2012, 341-355.
- [7] BERGER, James O., *Statistical Decision Theory and Bayesian Analysis*, Springer-Verlag, New York, 1985.
- [8] BERNARDO, José M. a Adrian F. M. SMITH, *Bayesian Theory*. John Wiley, Chichester, 1994.
- [9] BISHOP, Christopher, *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, 2006.
- [10] BISHOP, T. E. et al.. Blind image deconvolution: problem formulation and existing approaches. In 'Blind image deconvolution: Theory and Applications', CRC Press, 2007, 1-41.
- [11] COLE, E. R., *The removal of unknown image blurs by homomorphic filtering*. Disertace, Department of Electrical Engineering, University of Utah, 1973.
- [12] CONDAT, Laurent, Discrete Total Variation: New Definition and Minimization. *SIAM J. Imaging Sci.* 10(3), 2017, 1258–1290.

- [13] DEMPSTER, A. P., N. M. LAIRD a D. B. RUBIN, Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of Royal Statistical Society: Series B* 39(1), 1977, 1-38.
- [14] FERGUS, R. et al., Removing camera shake from a single photograph. In 'SIGGRAPH '06: ACM SIGGRAPH 2006 Papers", Association for Computing Machinery, New York, NY, United States, 2006, 787-794.
- [15] FISH, D. A., A. M. BRINICOMBE, E. R. PIKE a J. G. WALKER, Blind deconvolution by means of the Richardson–Lucy algorithm. *Journal of the Optical Society of America A* 12(1), 1995, 58-65.
- [16] GONZALEZ, Rafael C. a Richard E. WOODS, *Digital image processing*, 3rd ed. Upper Saddle River: Prentice-Hall, 2008.
- [17] HIRSCH M., S. SUVRIT, B. SCHÖLKOPF, a S. HARMELING, Efficient filter flow for space-variant multiframe blind deconvolution. In '2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition', IEEE, 2010, 607-614.
- [18] HIRSCH, M., Ch. J. SCHULER, S. HARMELING a B. SCHÖLKOPF, Fast Removal of Non-uniform Camera Shake. In '2011 International Conference on Computer Vision', IEEE, 2011, 463-470.
- [19] HUNT, B. R., The application of constrained least squares estimation to image restoration by digital computer. *IEEE Transactions on Computers* C-22(9), 1973, 805-812
- [20] CHAN, Tony F. a Chiu-Kwong WONG, Total variation blind deconvolution. *IEEE Transactions on Image Processing* 7(3), 1998, 370-375.
- [21] JOSHI, N., S. B. KANG, C. L. ZITNICK a R. SZELISKI, Image deblurring using inertial measurement sensors. *ACM Transactions on Graphics* 29(4), 2010, 1-9.
- [22] KHAN, Mohammad Emtiyaz a et al., Fast and Scalable Bayesian Deep Learning by Weight-Perturbation in Adam. *ArXiv:1806.04854v3 [stat.ML]*, 2018.
- [23] KHAN, M. E. a W. LIN, Conjugate-Computation Variational Inference : Converting Variational Inference in Non-Conjugate Models to Inferences in Conjugate Models. *ArXiv:1703.04265 [cs.LG]*, 2017.
- [24] KINGMA, Diederik P. a Jimmy Lei BA, Adam: A Method for Stochastic Optimization. *ArXiv:1412.6980 [cs.LG]*, 2015.
- [25] KINGMA, D. a M. WELLING, Auto-Encoding Variational Bayes. *ArXiv:1312.6114v10 [stat.ML]*, 2014.
- [26] KOTERA, J., V. ŠMÍDL a F. ŠROUBEK, Blind Deconvolution With Model Discrepancies. *IEEE Transactions on Image Processing* 26(5), 2017, 2533-2544.
- [27] LAGENDIJK, Reginald L., Jan BIEMOND a Dick E. BOEKEE, Identification and Restoration of Noisy Blurred Images Using the Expectation-Maximization Algorithm. *IEEE Transactions on Signal Processing* 38(7), 1990, 1180-1191.
- [28] LEVIN, A., W. YAIR, D. FREDO a W. T. FREEMAN, Understanding Blind Deconvolution Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(12), 2011, 2354-2367.



- [29] LIFSHIN, Eric, Yudhishtir P. KANDEL a Richard L. MOORE, Improving Scanning Electron Microscope Resolution for Near Planar Samples Through the Use of Image Restoration. *Microsc. Microanal.* 19, 2013, 1-12.
- [30] LUCY, L. B., An iterative technique for the rectification of observed distributions. *Astronomical Journal* 79(6), 1974, 745-754.
- [31] MOLINA, R., A. K. KATSAGGELOS, J. ABAD a J. MATEOS, 1997. A Bayesian approach to blind deconvolution based on Dirichlet distributions. In '1997 IEEE International Conference on Acoustics, Speech, and Signal Processing', IEEE, 1997, 2809-2812.
- [32] MOLINA, Rafael, Javier MATEOS a Aggelos K. KATSAGGELOS, Blind Deconvolution Using a Variational Approach to Parameter, Image, and Blur Estimation. *IEEE Transactions on Image Processing* 15(12), 2006, 3715-3727.
- [33] PALMER, J. A., K. KREUTZ-DELGADO a S. MAKEIG, Strong Sub- and Super-Gaussianity. In 'LVA/ICA'10: Proceedings of the 9th international conference on Latent variable analysis and signal separation', Springer-Verlag, Berlin, Heidelberg, 2010, 303-310.
- [34] PERRONE, Daniele a Paolo FAVARO, A Clearer Picture of Total Variation Blind Deconvolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(6), 2016, 1041-1055.
- [35] PRATT, William K., *Digital image processing: PIKS inside*, 3rd ed. Wiley, New York, 2001.
- [36] REN, Dongwei a et al., Neural Blind Deconvolution Using Deep Priors. In '2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)', IEEE, 2020, 3338-3347.
- [37] RICHARDSON, William Hadley, Bayesian-Based Iterative Method of Image Restoration\*. *J. Opt. Soc. Am.* 62(1), 1972, 55-59.
- [38] RUDIN, Leonid I., Stanley OSHER a Emad FATEMI, Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena* 60(1-4), 1992, 259-268.
- [39] SATO, Masa-Aki, Online Model Selection Based on the Variational Bayes. *Neural Computation* 13(7), 2001, 1649-1681.
- [40] SHAO, Wen-Ze et al., Regularized motion blur-kernel estimation with adaptive sparse image prior learning. *Pattern Recognition.* 51, 2016, 402-424.
- [41] SHOCHER, Assaf, Nadav COHEN a Michal IRANI, Zero-Shot Super-Resolution Using Deep Internal Learning. In '2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition', IEEE, 2018, 3118-3126.
- [42] ŠEVČÍK, Jakub, Václav ŠMÍDL a Filip ŠROUBEK, An Adaptive Correlated Image Prior for Image Restoration Problems. *IEEE Signal Processing Letters* 25(7), 2018, 1024-1028.
- [43] ŠMÍDL, Václav a Anthony QUINN, *The Variational Bayes Method in Signal Processing.* Springer-Verlag, Berlin, Heidelberg, 2006.
- [44] ŠMÍDL, Václav a Ondřej TICHÝ, Sparsity in Bayesian Blind Source Separation and Deconvolution. In 'Machine Learning and Knowledge Discovery in Databases.: ECML PKDD 2013. Lecture Notes in Computer Science. 8189', Springer, Berlin, Heidelberg, 2013, 548-563.

- [45] ŠOREL, Michal a Filip ŠROUBEK, Space-variant deblurring using one blurred and one underexposed image. In '2009 16th IEEE International Conference on Image Processing (ICIP)', IEEE, 2009, 157-160.
- [46] ŠROUBEK, Filip, Václav ŠMÍDL a Jan KOTERA, Understanding Image Priors in Blind Deconvolution. In '2014 IEEE International Conference on Image Processing', IEEE, 2014, 4492-4496.
- [47] TAI, Yu-Wing, Ping TAN a Michael S. BROWN, Richardson-Lucy Deblurring for Scenes under a Projective Motion Path. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(8), 2011, 1603-1618.
- [48] TALLÓN, M. et al., Space-variant blur deconvolution and denoising in the dual exposure problem. *Information Fusion* 14(2013), 2013, 369-409.
- [49] TIELEMAN, T. a G. HINTON, Lecture 6.5-RMSprop: Divide the gradient by a running average of its recent magnitude. In 'COURSERA: Neural Networks for Machine Learning 4', 2012.
- [50] TZIKAS, D., A. LIKAS a N. GALATSANOS, Variational Bayesian sparse kernel-based blind image deconvolution with student's-t priors. *IEEE Transactions on Image Processing* 18(4), 2009, 753-764.
- [51] ULYANOV, Dmitry, Andrea VEDALDI a Victor LEMPITSKY, Deep Image Prior. In '2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition', IEEE, 2018, 9446-9454.
- [52] VER HOEF, Jay M., Ephraim M. HANKS a Mevin B. HOOTEN, On the relationship between conditional (CAR) and simultaneous (SAR) autoregressive models. *Spatial Statistics* 25, 2018, 68-85.
- [53] WANG, Zhou, Alan C. BOVIK, Hamid R. SHEIKH a Eero P. SIMONCELLI, Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing* 13(4), 2004, 600-612.
- [54] WHYTE, O., J. SIVIC, A. ZISSERMAN a J. PONCE, Non-uniform Deblurring for Shaken Images. *Int J Comput Vis* 98, 2012, 168-186.
- [55] WILSON, Ashia C. a et al., The marginal value of adaptive gradient methods in machine learning. In 'NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems', Curran Associates Inc., 57 Morehouse Lane, Red Hook, NY, United States, 2017, 4151-4161.
- [56] WIPF, David a Haichao ZHANG, Revisiting Bayesian Blind Deconvolution. *Journal of Machine Learning Research* 15, 2014, 3775-3814.
- [57] ZHANG, Bo, Josiane ZERUBIA a Jean-Christophe OLIVO-MARIN, Gaussian approximations of fluorescence microscope point-spread function models. *Applied Optics* 46(10), 2007, 1819-1829.
- [58] ZHANG, Haichao, a David WIPF, Non-uniform camera shake removal using a spatially-adaptive sparse penalty. In 'Advances in Neural Information Processing Systems 26', Curran Associates Inc., 57 Morehouse Lane, Red Hook, NY, United States, 2013, 1556-1564.