BAYESIAN BLIND DECONVOLUTION OF IMAGES COMPARING JMAP, EM AND BVA WITH A STUDENT-T A PRIORI MODEL

A. Mohammad-Djafari

Laboratoire des signaux et systèmes (L2S)
UMR 8506 CNRS-SUPELEC-UNIV PARIS SUD
plateau de Moulon, 3 rue Joliot-Curie, 91192 GIF-SUR-YVETTE Cedex, France

ABSTRACT

Blind image deconvolution consists in restoring a blurred and noisy image when the point spread function of the blurring system is not known a priori. This inverse problem is ill-posed and need prior information to obtain a satisfactory solution. Regularization methods, well known, for simple image deconvolution is not enough. Bayesian inference approach with appropriate priors on the image as well as on the PSF has been used successfully, in particular with a Gaussian prior on the PSF and a sparsity enforcing prior on the image. Joint Maximum A posteriori (JMAP), Expectation-Maximization (EM) algorithm for marginalized MAP and Variational Bayesian Approximation (VBA) are the methods which have been considered recently with some advantages for the last one. In this paper, first we review these methods and give some original insights by comparing them, in particular for their respective properties, advantages and drawbacks and their computational complexity. Then we propose to look at these methods in two cases: A simple one which is using Gaussian priors for both the PSF and the image and a more appropriate case which is a Student-t prior for the image to enhance the sharpness (sparsity) of the image while keeping Gaussian prior for the PSF. We take advantages of the Infinite Gaussian Mixture (IGM) property of the Student-t to consider a hierarchical Gaussian-Inverse Gamma prior model for the image. We give detailed comparison of these three methods for this case.

Keywords

Blind Deconvolution; Bayesian JMAP; Expectation-Maximization (EM); Variational Bayesian Approximation (BVA); Student-t prior models; Blind Image restoration.

1. INTRODUCTION

A blurred image $g(x,y)$ can be modeled as the convolution of the original sharp image $f(x,y)$ with a point spread function (pdf) $h(x,y)$:

$$g(x,y) = f(x,y) * h(x,y) + \epsilon(x,y), \tag{1}$$

where * represents the convolution operation and $\epsilon(x,y)$ the errors. The inverse problem of the deconvolution consists in estimating $f(x,y)$ from the blurred and noisy image $g(x,y)$ when the Point Spread Function (PSF) $h(x,y)$ of the blurring system is known a priori. This inverse problem is ill-posed and needs prior information on the original image. Regularization theory and the Bayesian inversion have been successful for this task. See for example [1, 2] and [3, 4, 5, 6, 7, 8, 9, 10].

Blind Deconvolution consists in restoring the blurred and noisy image $g(x,y)$ when the PSF $h(x,y)$ is not known a priori. This inverse problem is still more ill-posed and need strong prior information to obtain a satisfactory solution. Regularization theory and simple Bayesian inversion, well known, for simple deconvolution are no more enough [11, 12]. Bayesian inference approach with appropriate priors on the image as well as on the PSF has been used successfully [2, 13, 14, 15, 16]. In particular, a Gaussian prior on the PSF and a sparsity enforcing prior on the image has been used successfully [17, 12, 18].

Joint Maximum A posteriori (JMAP) estimation of the image $f(x,y)$ and the PSF $h(x,y)$, Expectation-Maximization algorithm for marginal MAP and the Variational Bayesian Approximation (BVA) are three main methods which have been considered recently with some advantages for the last one [19, 20, 21, 15, 22, 23].

In this paper, first we review the basic ideas of these methods and give some original insights by comparing these three methods and their associated algorithms. Then, we discuss more in detail their properties as well as their computational costs and complexities for two cases: one for the case of Gaussian priors for both image and the PSF and the second for the case where we still keep a Gaussian prior for the PSF but we propose to use the Student-t prior for the image. The Student-t model has the advantage of sparsity enforcing property and its Infinite Gaussian Mixture property gives the possibility of proposing a hierarchical structure generative graphical model for the output data. Finally, we give details of the three estimation methods of JMAP, BEM and VBA for this prior model and discuss more in detail their properties as well as their computational costs and complexities.

2. BACKGROUND ON BAYESIAN APPROACH FOR BLIND DECONVOLUTION

Assuming a forward convolution model, additive noise, and discretized model, we have:

$$\boldsymbol{g} = \boldsymbol{h} * \boldsymbol{f} + \boldsymbol{\epsilon} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\epsilon} = \boldsymbol{F}\boldsymbol{h} + \boldsymbol{\epsilon}, \tag{2}$$

where $\boldsymbol{f}$ represents the unknown sharp image, $\boldsymbol{h}$ the unknown PSF, $\boldsymbol{\epsilon}$ the errors, $\boldsymbol{H}$ the 2D convolution matrix (Toeplitz-Bloc-Toeplitz) obtained from the PSF $\boldsymbol{h}$ and $\boldsymbol{F}$ the 2D convolution matrix obtained from the image $\boldsymbol{f}$ [24, 25, 26].

Using this forward model and assigning the forward $p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{h})$ and the prior laws $p(\boldsymbol{f})$ and $p(\boldsymbol{h})$, the Bayesian approach starts with the expression of the joint posterior law

$$p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g}) = \frac{p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{h})\, p(\boldsymbol{f})\, p(\boldsymbol{h})}{p(\boldsymbol{g})}. \tag{3}$$

From here, basically, two approaches have been proposed to estimate both $\boldsymbol{f}$ and $\boldsymbol{h}$:

- JMAP:

$$(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) = \arg\max_{(\boldsymbol{f},\boldsymbol{h})} \{p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g})\} \tag{4}$$

  and

- Marginal likelihood estimate of $\boldsymbol{h}$:

$$\widehat{\boldsymbol{h}} = \arg\max_{\boldsymbol{h}} \{p(\boldsymbol{h}|\boldsymbol{g})\} \tag{5}$$

  followed by the marginal MAP estimate of $\boldsymbol{f}$:

$$\widehat{\boldsymbol{f}} = \arg\max_{\boldsymbol{f}} \left\{p(\boldsymbol{f}|\widehat{\boldsymbol{h}},\boldsymbol{g})\right\}. \tag{6}$$

The first one is easily understood and linked to the classical regularization theory, if we note that:

$$(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) = \arg\max_{(\boldsymbol{f},\boldsymbol{h})} \{p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g})\} = \arg\min_{(\boldsymbol{f},\boldsymbol{h})} \left\{J_{\text{MAP}}(\boldsymbol{f},\boldsymbol{h})\right\}$$

$$\text{with} \quad J_{\text{MAP}}(\boldsymbol{f},\boldsymbol{h}) = -\ln p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{h}) - \ln p(\boldsymbol{f}) - \ln p(\boldsymbol{h}) \tag{7}$$

which, with the following Gaussian priors: $p(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{\epsilon}|0, v_\epsilon \boldsymbol{I})$, $p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}|0, v_f \boldsymbol{I})$ and $p(\boldsymbol{h}) = \mathcal{N}(\boldsymbol{h}|0, v_h(\boldsymbol{C}_h'\boldsymbol{C}_h)^{-1})$ becomes:

$$J_{\text{MAP}}(\boldsymbol{f},\boldsymbol{h}) = \frac{1}{v_\epsilon}\|\boldsymbol{g} - \boldsymbol{h} * \boldsymbol{f}\|_2^2) + \frac{1}{v_f}\|\boldsymbol{f}\|_2^2 + \frac{1}{v_h}\|\boldsymbol{C}_h\boldsymbol{h}\|_2^2. \tag{8}$$

2.1. Joint MAP estimation:

Noting that $\|\boldsymbol{g} - \boldsymbol{h} * \boldsymbol{f}\|_2^2 = \|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|_2^2 = \|\boldsymbol{g} - \boldsymbol{F}\boldsymbol{h}\|_2^2$, the JMAP Criterion (8) can be written as:

$$\begin{aligned} J_{\text{MAP}}(\boldsymbol{f},\boldsymbol{h}) &= \tfrac{1}{v_\epsilon}\|\boldsymbol{g} - \boldsymbol{H} * \boldsymbol{f}\|_2^2) + \tfrac{1}{v_f}\|\boldsymbol{f}\|_2^2 + \tfrac{1}{v_h}\|\boldsymbol{C}_h\boldsymbol{h}\|_2^2 \\ &= \tfrac{1}{v_\epsilon}\|\boldsymbol{g} - \boldsymbol{F} * \boldsymbol{h}\|_2^2) + \tfrac{1}{v_f}\|\boldsymbol{f}\|_2^2 + \tfrac{1}{v_h}\|\boldsymbol{C}_h\boldsymbol{h}\|_2^2. \end{aligned} \tag{9}$$

So, its alternate optimization with respect to $\boldsymbol{f}$ (with fixed $\boldsymbol{h}$) and $\boldsymbol{h}$ (with fixed $\boldsymbol{f}$) result to the following iterative algorithm:

JMAP Algorithm:

Initialization:
$\boldsymbol{h}^{(0)} = \boldsymbol{h}_0, \quad \boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}^{(0)})$
Iterations:
$\boldsymbol{f}^{(k)} = \arg\min_{\boldsymbol{f}} \{J_{\text{MAP}}(\boldsymbol{f},\boldsymbol{h})\} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f \boldsymbol{I})^{-1}\boldsymbol{H}'\boldsymbol{g}$
$\boldsymbol{F} = \texttt{Convmtx}(\boldsymbol{f}^{(k-1)})$
$\boldsymbol{h}^{(k)} = \arg\min_{\boldsymbol{h}} \{J_{\text{MAP}}(\boldsymbol{f},\boldsymbol{h})\} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_h \boldsymbol{C}_h'\boldsymbol{C}_h)^{-1}\boldsymbol{F}'\boldsymbol{g}$
$\boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}^{(k-1)})$

(10)

where $\lambda_f = \frac{v_f}{v_\epsilon}$ and $\lambda_h = \frac{v_h}{v_\epsilon}$. This algorithm can be visualized as in the following figure:

$$\begin{array}{ccc} \widehat{\boldsymbol{h}}^{(0)} \rightarrow \widehat{\boldsymbol{h}} \rightarrow & \boxed{\begin{array}{l} \boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}) \\ \widehat{\boldsymbol{f}} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f \boldsymbol{I})^{-1}\boldsymbol{H}'\boldsymbol{g} \end{array}} & \rightarrow \widehat{\boldsymbol{f}} \\ \uparrow & & \downarrow \\ \widehat{\boldsymbol{h}} \leftarrow & \boxed{\begin{array}{l} \boldsymbol{F} = \texttt{Convmtx}(\boldsymbol{f}) \\ \widehat{\boldsymbol{h}} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_h \boldsymbol{C}_h'\boldsymbol{C}_h)^{-1}\boldsymbol{F}'\boldsymbol{g} \end{array}} & \leftarrow \widehat{\boldsymbol{f}} \end{array}$$

2.2. Bayesian Expectation-Maximization (BEM)

The second method, needs first the integration (marginalization):

$$p(\boldsymbol{h}|\boldsymbol{g}) = \int p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g})\, \mathrm{d}\boldsymbol{f} \tag{11}$$

which can not often be done analytically and needs approximation methods to obtain the solution. The Expectation-Maximization (EM) and its Bayesian version (BEM) try to find this solution by alternate maximizing of some lower bound $p^*(\boldsymbol{h}|\boldsymbol{g})$ to it. In summary, the BEM algorithm can be written as a two step iterative algorithm:

- E step:Compute the expected value:

$$Q(\boldsymbol{h},\boldsymbol{h}^{(k-1)}) = \langle \ln p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g})\rangle_{p(\boldsymbol{f}|\boldsymbol{h}^{(k-1)},\boldsymbol{g})} \tag{12}$$

- M step:

$$\boldsymbol{h}^{(k)} = \arg\max_{\boldsymbol{h}} \left\{Q(\boldsymbol{h},\boldsymbol{h}^{(k-1)})\right\} \tag{13}$$

For the Gaussian case, noting that

$$\begin{aligned} -\ln p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g}) &= c + \tfrac{1}{2} J_{\text{MAP}}(\boldsymbol{f},\boldsymbol{h}) \\ &= c + \tfrac{1}{2}\left[\tfrac{1}{v_\epsilon}\|\boldsymbol{g}-\boldsymbol{h}*\boldsymbol{f}\|_2^2) + \tfrac{1}{v_f}\|\boldsymbol{f}\|_2^2 + \tfrac{1}{v_h}\|\boldsymbol{C}_h\boldsymbol{h}\|_2^2\right] \end{aligned} \tag{14}$$

where c is a constant which will be eliminated since after, and that

$$\begin{aligned} &< -\ln p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g}) > \\ &=< \|\boldsymbol{g}-\boldsymbol{h}*\boldsymbol{f}\|_2^2) > +\lambda_f < \|\boldsymbol{f}\|_2^2) > +\lambda_h\|\boldsymbol{C}_h\boldsymbol{h}\|_2^2 \\ &= \tfrac{1}{v_\epsilon}\Big[\|\boldsymbol{g}\|^2 - 2\boldsymbol{g}' < \boldsymbol{F} > \boldsymbol{h} + \| < \boldsymbol{F} > \boldsymbol{h}\|^2 + \\ &\text{Tr}\left\{\boldsymbol{H}\text{Cov}\left[\boldsymbol{f}\right]\boldsymbol{H}'\right\}\Big] + \lambda_h\|\boldsymbol{C}_h\boldsymbol{h}\|_2^2 \\ &= \left[\|\boldsymbol{g}- < \boldsymbol{F} > \boldsymbol{h}\|^2 + \|\boldsymbol{D}_f\boldsymbol{h}\|^2\right] + \lambda_h\|\boldsymbol{C}_h\boldsymbol{h}\|_2^2 \end{aligned} \tag{15}$$

where we assumed that $\text{Tr}\left\{\boldsymbol{H}\text{Cov}\left[\boldsymbol{f}\right]\boldsymbol{H}'\right\}$ can be written as $\|\boldsymbol{D}_f\boldsymbol{h}\|^2$ which is possible. Then, with this relation, it is easy to write down the Bayesian EM algorithm as follows:

Bayesian EM Algorithm:

Initialization:
$\boldsymbol{h}^{(0)} = \boldsymbol{h}_0, \quad \boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}^{(0)})$

Iterations:
$$\begin{aligned} &\boldsymbol{\Sigma}_f = v_\epsilon(\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{I})^{-1} \\ &\boldsymbol{f}^{(k)} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{I})^{-1}\boldsymbol{H}'\boldsymbol{g} \\ &\boldsymbol{F} = \texttt{Convmtx}(\boldsymbol{f}^{(k-1)}) \\ &\text{Tr}\left\{\boldsymbol{H}\boldsymbol{\Sigma}_f\boldsymbol{H}'\right\} = \|\boldsymbol{D}_f\boldsymbol{h}\|_2^2 \\ &\boldsymbol{h}^{(k)} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_h\boldsymbol{C}_h'\boldsymbol{C}_h + \boldsymbol{D}_f'\boldsymbol{D}_f)^{-1}\boldsymbol{F}'\boldsymbol{g} \\ &\boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}^{(k-1)}) \end{aligned} \tag{16}$$

This algorithm can be visualized as follows:

$$\widehat{\boldsymbol{h}}^{(0)} \rightarrow \widehat{\boldsymbol{h}} \rightarrow \boxed{\begin{aligned} &\boldsymbol{H} = \texttt{Convmtx}(\widehat{\boldsymbol{h}}) \\ &\widehat{\boldsymbol{f}} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{I})^{-1}\boldsymbol{H}'\boldsymbol{g} \\ &\widehat{\boldsymbol{\Sigma}} = v_\epsilon(\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{I})^{-1} \end{aligned}} \Rightarrow \begin{matrix}\widehat{\boldsymbol{f}} \\ \widehat{\boldsymbol{\Sigma}}\end{matrix}$$

$$\uparrow \qquad\qquad\qquad\qquad \Downarrow$$

$$\widehat{\boldsymbol{h}} \leftarrow \boxed{\begin{aligned} &\text{Tr}\left\{\boldsymbol{H}\boldsymbol{\Sigma}_f\boldsymbol{H}'\right\} = \|\boldsymbol{D}_f\boldsymbol{h}\|_2^2 \\ &\boldsymbol{F} = \texttt{Convmtx}(\widehat{\boldsymbol{f}}) \\ &\widehat{\boldsymbol{h}} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_h\boldsymbol{C}_h'\boldsymbol{C}_h + \boldsymbol{D}_h'\boldsymbol{D}_h)^{-1}\boldsymbol{F}'\boldsymbol{g} \end{aligned}} \Leftarrow \begin{matrix}\widehat{\boldsymbol{f}} \\ \widehat{\boldsymbol{\Sigma}}\end{matrix}$$

2.3. Variational Bayesian Approximation (VBA)

The third approach which, in some way, generalizes BEM, is the VBA method which consists in approximating the joint posterior law $p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g})$ by a separable one $q(\boldsymbol{f},\boldsymbol{h}) = q_1(\boldsymbol{f}|\boldsymbol{h})\, q_2(\boldsymbol{h}|\boldsymbol{f})$ by minimizing the Kullback-Leibler KL$(q : p)$. It is easily shown that the alternate optimization of this criterion results to the following iterative algorithm:

- E step:Compute the expected values $\langle \ln p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g})\rangle_{q_1}$ and $\langle \ln p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g})\rangle_{q_2}$ and deduce:

$$\left\{\begin{array}{l} q_1(\boldsymbol{f}|\boldsymbol{h}^{(k)}) \propto \exp\left\{\langle \ln p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g})\rangle_{q_2(\boldsymbol{h}|\boldsymbol{f}^{(k-1)})}\right\} \\ q_2(\boldsymbol{h}|\boldsymbol{f}^{(k)}) \propto \exp\left\{\langle \ln p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g})\rangle_{q_1(\boldsymbol{f}|\boldsymbol{h}^{(k-1)})}\right\} \end{array}\right. \tag{17}$$

- M step:

$$\left\{\begin{array}{l} \boldsymbol{f}^{(k+1)} = \arg\max_{\boldsymbol{f}}\left\{q_2(\boldsymbol{f}|\boldsymbol{h}^{(k)})\right\} \\ \boldsymbol{h}^{(k+1)} = \arg\max_{\boldsymbol{h}}\left\{q_2(\boldsymbol{h}|\boldsymbol{f}^{(k)})\right\} \end{array}\right. \tag{18}$$

Here too, it can be shown that with the Gaussian priors, we obtain the following algorithm:

VBA Algorithm :

Initialization:
$$\begin{aligned} &\boldsymbol{h}^{(0)} = \boldsymbol{h}_0; \quad \boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}^{(0)}) \\ &\boldsymbol{\Sigma}_f = v_\epsilon(\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{I})^{-1} \\ &\boldsymbol{f} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{I})^{-1}\boldsymbol{H}'\boldsymbol{g} \\ &\boldsymbol{F} = \texttt{Convmtx}(\boldsymbol{f}) \\ &\text{Tr}\left\{\boldsymbol{H}\boldsymbol{\Sigma}_f\boldsymbol{H}'\right\} = \|\boldsymbol{D}_f\boldsymbol{h}\|_2^2 \end{aligned}$$

Iterations:
$$\begin{aligned} &\boldsymbol{\Sigma}_h = v_\epsilon(\boldsymbol{F}'\boldsymbol{F} + \lambda_h\boldsymbol{C}_h'\boldsymbol{C}_h + \boldsymbol{D}_f'\boldsymbol{D}_f)^{-1} \\ &\boldsymbol{h}^{(k)} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_h\boldsymbol{C}_h'\boldsymbol{C}_h + v_\epsilon\boldsymbol{D}_f'\boldsymbol{D}_f)^{-1}\boldsymbol{F}'\boldsymbol{g} \\ &\boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}^{(k-1)}) \\ &\text{Tr}\left\{\boldsymbol{F}\boldsymbol{\Sigma}_h\boldsymbol{F}'\right\} = \|\boldsymbol{D}_h\boldsymbol{f}\|_2^2 \\ &\boldsymbol{\Sigma}_f = v_\epsilon(\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{I} + v_\epsilon\boldsymbol{D}_h'\boldsymbol{D}_h)^{-1} = \|\boldsymbol{D}_h\boldsymbol{f}\|^2 \\ &\boldsymbol{f}^{(k)} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{I} + v_\epsilon\boldsymbol{D}_h'\boldsymbol{D}_h)^{-1}\boldsymbol{H}'\boldsymbol{g} \\ &\boldsymbol{F} = \texttt{Convmtx}(\boldsymbol{f}^{(k-1)}) \\ &\text{Tr}\left\{\boldsymbol{H}\boldsymbol{\Sigma}_f\boldsymbol{H}'\right\} = \|\boldsymbol{D}_f\boldsymbol{h}\|_2^2 \end{aligned} \tag{19}$$

$$\begin{matrix}\widehat{\boldsymbol{h}}^{(0)} \\ \widehat{\boldsymbol{\Sigma}}_f^{(0)}\end{matrix} \Rightarrow \begin{matrix}\widehat{\boldsymbol{h}} \\ \widehat{\boldsymbol{\Sigma}}_f\end{matrix} \Rightarrow \boxed{\begin{aligned} &\boldsymbol{H} = \texttt{Convmtx}(\widehat{\boldsymbol{h}}) \\ &\text{Tr}\left\{\boldsymbol{F}\boldsymbol{\Sigma}_h\boldsymbol{F}'\right\} = \|\boldsymbol{D}_f\boldsymbol{f}\|_2^2 \\ &\widehat{\boldsymbol{\Sigma}}_f = v_\epsilon(\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{I} + \boldsymbol{D}_f'\boldsymbol{D}_f)^{-1} \\ &\widehat{\boldsymbol{f}} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{I} + \boldsymbol{D}_f'\boldsymbol{D}_f)^{-1}\boldsymbol{H}'\boldsymbol{g} \end{aligned}} \Rightarrow \begin{matrix}\widehat{\boldsymbol{f}} \\ \widehat{\boldsymbol{\Sigma}}_f\end{matrix}$$

$$\Uparrow \qquad\qquad\qquad\qquad \Downarrow$$

$$\begin{matrix}\widehat{\boldsymbol{h}} \\ \widehat{\boldsymbol{\Sigma}}_h\end{matrix} \Leftarrow \boxed{\begin{aligned} &\boldsymbol{F} = \texttt{Convmtx}(\boldsymbol{f}) \\ &\text{Tr}\left\{\boldsymbol{H}\boldsymbol{\Sigma}_f\boldsymbol{H}'\right\} = \|\boldsymbol{D}_h\boldsymbol{h}\|_2^2 \\ &\widehat{\boldsymbol{h}} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_h\boldsymbol{C}_h'\boldsymbol{C}_h + \boldsymbol{D}_h'\boldsymbol{D}_h)^{-1}\boldsymbol{F}'\boldsymbol{g} \\ &\widehat{\boldsymbol{\Sigma}}_h = v_\epsilon(\boldsymbol{H}'\boldsymbol{H} + \lambda_h\boldsymbol{C}_h'\boldsymbol{C}_h + \boldsymbol{D}_h'\boldsymbol{D}_h)^{-1} \end{aligned}} \Leftarrow \begin{matrix}\widehat{\boldsymbol{f}} \\ \widehat{\boldsymbol{\Sigma}}_f\end{matrix}$$

2.4. Comparison JMAP1, BEM1 and VBA1

Comparing the three algorithms JMAP (10), BEM (16) and VBA (16), we can make the following remarks:

- In JMAP, there is no need to matrix inversion. At each step, we can find $\boldsymbol{f}^{(k)}$ and $\boldsymbol{h}^{(k)}$ using an optimization algorithm.
- In BEM, at each step, wee need to compute $\boldsymbol{\Sigma}_f$ and do the matrix decomposition $\text{Tr}\left\{\boldsymbol{H}\boldsymbol{\Sigma}_f\boldsymbol{H}'\right\} = \|\boldsymbol{D}'_f\boldsymbol{h}\|^2$. This is a very costly operation due to the size of the matrices $\boldsymbol{H}'$ and $\boldsymbol{\Sigma}_f$.
- In VBA, at each step, wee need to compute $\boldsymbol{\Sigma}_f$ and do the matrix decomposition $\text{Tr}\left\{\boldsymbol{H}\boldsymbol{\Sigma}_f\boldsymbol{H}'\right\} = \|\boldsymbol{D}'_f\boldsymbol{h}\|^2$ and also to compute $\boldsymbol{\Sigma}_f$ and do the matrix decomposition $\text{Tr}\left\{\boldsymbol{F}\boldsymbol{\Sigma}_h\boldsymbol{F}'\right\} = \|\boldsymbol{D}'_h\boldsymbol{f}\|^2$. There are two very costly operations.

For practical applications, we have to write specialized algorithm taking account of the particular structures of the matrix operators $\boldsymbol{H}$ and $\boldsymbol{F}$. In particular, in Blind deconvolution, these matrices are Toeplitz (or Block-Toeplitz) and we can approximate them with appropriate circulant (or Bloc-circulant) matrices and use the Fast Fourier Transform (FFT) to write appropriate algorithms.

3. JMAP, BEM AND VBA WITH A STUDENT-T PRIOR

As we are, in general, looking for a sharp image, a Gaussian prior is not very appropriate. We may use any sparsity enforcing priors. Between those prior law, one is very interesting, the Student-t prior:

$$\mathcal{T}(f_j|\nu,\mu_j,v_f) = \int_0^\infty \mathcal{N}(f_j|\mu_j, z_j^{-1}v_f)\,\mathcal{G}(z_j|\nu/2,\nu/2)\,\mathrm{d}z_j$$

where

$$\mathcal{N}(f_j|\mu_j, z_j^{-1}v_f) = |2\pi v_f/z_j|^{-1/2}\exp\left\{-\frac{1}{2v_f}z_j(x_j-\mu_j)^2\right\}$$

and
$$\mathcal{G}(z_j|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}z_j^{\alpha-1}\exp\left\{-\beta z_j\right\}.$$

Now, using the forward model (2) and the following priors

$$\begin{array}{l} p(\boldsymbol{\epsilon}|v_\epsilon) = \mathcal{N}(\boldsymbol{\epsilon}|0, v_\epsilon\boldsymbol{I}) \to p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{h},v_\epsilon) = \mathcal{N}(\boldsymbol{g}|\boldsymbol{h}*\boldsymbol{f}, v_\epsilon\boldsymbol{I}),\\ p(\boldsymbol{h}|v_h) = \mathcal{N}(\boldsymbol{h}|0, v_h(\boldsymbol{C}'_h\boldsymbol{C}_h)^{-1})\\ p(\boldsymbol{f}|\boldsymbol{z},v_f) = \mathcal{N}(\boldsymbol{f}|0, v_f\boldsymbol{Z}^{-1}) \text{ with } \boldsymbol{Z} = \text{Diag}[z_1,\cdots,z_N]\\ p(\boldsymbol{z}|\alpha,\beta) = \prod_{j=1}^N \mathcal{G}(z_j|\alpha,\beta) \end{array} \quad (20)$$

we have

$$\begin{aligned} p(\boldsymbol{f},\boldsymbol{z},\boldsymbol{h}|\boldsymbol{g},) &\propto p(\boldsymbol{g}|\boldsymbol{h},\boldsymbol{f})\,p(\boldsymbol{h}|v_h)\,p(\boldsymbol{f}|\boldsymbol{z},v_f)\,p(\boldsymbol{z}|\alpha,\beta)\\ &\propto \mathcal{N}(\boldsymbol{g}|\boldsymbol{h}*\boldsymbol{f}, v_\epsilon\boldsymbol{I})\,\mathcal{N}(\boldsymbol{h}|0, v_h(\boldsymbol{C}'_h\boldsymbol{C}_h)^{-1})\\ &\quad \mathcal{N}(\boldsymbol{f}|0, v_f\boldsymbol{Z}^{-1})\prod_{j=1}^N \mathcal{G}(z_j|\alpha,\beta)\\ &\propto \exp\left\{-\tfrac{1}{v_\epsilon}J_{\text{MAP}}(\boldsymbol{f},\boldsymbol{z},\boldsymbol{h})\right\} \end{aligned} \quad (21)$$

which results to:

$$\begin{aligned} J_{\text{MAP}}(\boldsymbol{f},\boldsymbol{z},\boldsymbol{h}) &= \|\boldsymbol{g}-\boldsymbol{h}*\boldsymbol{f}\|^2 + \lambda_h\|\boldsymbol{C}_h\boldsymbol{h}\|^2\\ &+\lambda_f\|\boldsymbol{Z}^{1/2}\boldsymbol{f}\|^2 + 2v_\epsilon\left[\sum_{j=1}^N(\alpha-1)\ln z_j + \beta z_j\right] \end{aligned} \quad (22)$$

Using this expression, we can obtain easily the necessary developments to describe the the algorithms JMAP, BEM and VBA with this prior model.

JMAP Blind Deconvolution Algorithm with Student-t prior:

Initialization:

$\boldsymbol{h}^{(0)} = \boldsymbol{h}_0, \quad \boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}^{(0)}), \quad \boldsymbol{z}^{(0)} = 1$

Iterations:

$\boldsymbol{f}^{(k)} = \arg\min_{\boldsymbol{f}}\left\{J_{\text{MAP}}(\boldsymbol{f},\boldsymbol{z},\boldsymbol{h})\right\} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{Z})^{-1}\boldsymbol{H}'\boldsymbol{g}$

$\boldsymbol{z}^{(k)} = \arg\min_{\boldsymbol{z}}\left\{J_{\text{MAP}}(\boldsymbol{f},\boldsymbol{z},\boldsymbol{h})\right\} \to \widehat{z}_j^{(k)} = \frac{\widehat{\beta}_j}{\widehat{\alpha}_j}$

with $\widehat{\alpha}_j = \frac{1}{2}+\alpha$ and $\widehat{\beta}_j = \beta + \frac{1}{2} + \|\boldsymbol{g}-\boldsymbol{h}*\boldsymbol{f}\|^2$

$\boldsymbol{F} = \texttt{Convmtx}(\boldsymbol{f}^{(k-1)})$

$\boldsymbol{h}^{(k)} = \arg\min_{\boldsymbol{h}}\left\{J_{\text{MAP}}(\boldsymbol{f},\boldsymbol{z},\boldsymbol{h})\right\} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_h\boldsymbol{C}'_h\boldsymbol{C}_h)^{-1}\boldsymbol{F}'\boldsymbol{g}$

$\boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}^{(k-1)})$

(23)

where $\lambda_f = \frac{v_f}{v_\epsilon}$ and $\lambda_h = \frac{v_h}{v_\epsilon}$. This is illustrated in the following:

$\widehat{\boldsymbol{h}}^{(0)}, \widehat{\boldsymbol{z}}^{(0)} \Rightarrow \widehat{\boldsymbol{h}}, \widehat{\boldsymbol{z}} \Rightarrow$ $\boxed{\boldsymbol{Z} = \text{diag}\left[\widehat{\boldsymbol{z}}\right], \quad \boldsymbol{H} = \texttt{Convmtx}(\widehat{\boldsymbol{h}}); \quad \widehat{\boldsymbol{f}} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{Z})^{-1}\boldsymbol{H}'\boldsymbol{g}}$ $\to \widehat{\boldsymbol{f}}$

$\Uparrow \qquad\qquad \downarrow$

$\widehat{\boldsymbol{h}}, \boldsymbol{z} \Leftarrow$ $\boxed{\widehat{\alpha}_j = \frac{1}{2}+\alpha, \widehat{\beta}_j = \beta + \frac{1}{2} + \|\boldsymbol{g}-\boldsymbol{h}*\boldsymbol{f}\|^2; \quad \widehat{z}_j^{(k)} = \frac{\widehat{\beta}_j}{\widehat{\alpha}_j}; \quad \boldsymbol{F} = \texttt{Convmtx}(\boldsymbol{f}); \quad \widehat{\boldsymbol{h}} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_h\boldsymbol{C}'_h\boldsymbol{C}_h)^{-1}\boldsymbol{F}'\boldsymbol{g}}$ $\leftarrow \widehat{\boldsymbol{f}}$

Following the same approach, for BEM we obtain:

BEM Blind Deconvolution Algorithm with Studet-t prior:

Initialization:

$\boldsymbol{h}^{(0)} = \boldsymbol{h}_0, \quad \boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}^{(0)}), \quad \boldsymbol{z}^{(0)} = 1$

Iterations:

$\boldsymbol{\Sigma}_f = v_\epsilon(\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{Z})^{-1}$

$\boldsymbol{f}^{(k)} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{Z})^{-1}\boldsymbol{H}'\boldsymbol{g}$

$\boldsymbol{F} = \texttt{Convmtx}(\boldsymbol{f}^{(k-1)})$

$\text{Tr}\left\{\boldsymbol{H}\boldsymbol{\Sigma}_f\boldsymbol{H}'\right\} = \|\boldsymbol{D}_f\boldsymbol{h}\|^2$

$\widehat{z}_j^{(k)} = \frac{\widehat{\beta}_j}{\widehat{\alpha}_j}$

with $\widehat{\alpha}_j = \frac{1}{2}+\alpha$ and $\widehat{\beta}_j = \beta + \frac{1}{2} + \boldsymbol{g}-\boldsymbol{h}*\boldsymbol{f}\|^2$

$\boldsymbol{h}^{(k)} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_h\boldsymbol{C}'_h\boldsymbol{C}_h + \boldsymbol{D}'_f\boldsymbol{D}_f)^{-1}\boldsymbol{F}'\boldsymbol{g}$

$\boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}^{(k-1)})$

(24)

The flow diagram of this algorithm is shown in the following:

$$\begin{array}{c}\widehat{h}^{(0)}\\ \widehat{z}^{(0)}\end{array} \Rightarrow \begin{array}{c}\widehat{h}\\ \widehat{z}\end{array} \Rightarrow \boxed{\begin{array}{l} Z=\text{diag}\left[\widehat{z}\right], \quad H = \texttt{Convmtx}(\widehat{h}) \\ \widehat{\Sigma}_f = v_\epsilon(H'H+\lambda_f Z)^{-1} \\ \widehat{f} = (H'H+\lambda_f Z)^{-1}H'g\end{array}} \Rightarrow \begin{array}{c}\widehat{f}\\ \widehat{\Sigma}_f\end{array}$$

$$\Uparrow \qquad\qquad\qquad\qquad \Downarrow$$

$$\begin{array}{c}\widehat{h}\\ \widehat{z}\end{array} \Leftarrow \boxed{\begin{array}{l} \widehat{\alpha}_j = \frac{1}{2}+\alpha, \widehat{\beta}_j = \beta + \frac{1}{2} + \|g - h * f\|^2 \\ \widehat{z}_j^{(k)} = \frac{\widehat{\beta}_j}{\widehat{\alpha}_j} \\ F = \texttt{Convmtx}(f) \\ \text{Tr}\left\{H\Sigma_f H'\right\} = \|D_h h\|^2 \\ \widehat{h} = (F'F + \lambda_h C_h' C_h = D_h' D_h)^{-1} F'g \end{array}} \Leftarrow \begin{array}{c}\widehat{f}\\ \widehat{\Sigma}_f\end{array}$$

Again, following the same steps, we obtain for VBA:

VBA Blind Deconvolution Algorithm with Studet-t prior:

Initialization:

$h^{(0)} = h_0; \quad H = \texttt{Convmtx}(h^{(0)}), \quad z^{(0)} = 1;$

$\Sigma_f = v_\epsilon(H'H + \lambda_f I)^{-1}$

$f = (H'H + \lambda_f I)^{-1} H'g$

$F = \texttt{Convmtx}(f)$

$\text{Tr}\left\{H\Sigma_f H'\right\} = \|D_f h\|^2$

Iterations:

$\Sigma_h = v_\epsilon(F'F + \lambda_h C_h' C_h + D_h' D_h)^{-1}$

$h^{(k)} = (F'F + \lambda_h C_h' C_h + v_\epsilon D_h' D_h)^{-1} F'g$

$H = \texttt{Convmtx}(h^{(k-1)})$

$\text{Tr}\left\{F\Sigma_h F'\right\} = \|D_h f\|^2$

$\Sigma_f = v_\epsilon(H'H + \lambda_f I + v_\epsilon D_f' D_f)^{-1} = \|D_h f\|^2$

$f^{(k)} = (H'H + \lambda_f I + v_\epsilon D_f' D_f)^{-1} H'g$

$F = \texttt{Convmtx}(f^{(k-1)})$

$\text{Tr}\left\{H\Sigma_f H'\right\} = \|D_f h\|^2$

$\widehat{z}_j^{(k)} = \frac{\widehat{\beta}_j}{\widehat{\alpha}_j}$

with $\widehat{\alpha}_j = \frac{1}{2} + \alpha$ and $\widehat{\beta}_j = \beta + \frac{1}{2} + \|g - h * f\|^2$ (25)

The flow diagram of this algorithm is shown in the following:

$$\begin{array}{c}\widehat{h}^{(0)}\\ \Sigma_h^{(0)}\\ \widehat{z}^{(0)}\end{array} \Rightarrow \begin{array}{c}\widehat{h}\\ \Sigma_h\\ \widehat{z}\end{array} \Rightarrow \boxed{\begin{array}{l} Z=\text{diag}\left[\widehat{z}\right], \quad H = \texttt{Convmtx}(\widehat{h}) \\ \text{Tr}\left\{F\Sigma_h F'\right\} = \|D_h f\|^2 \\ \widehat{\Sigma}_f = v_\epsilon(H'H+\lambda_f Z + D_f' D_f)^{-1} \\ \widehat{f} = (H'H+\lambda_f Z + D_f' D_f)^{-1}H'g\end{array}} \Rightarrow \begin{array}{c}\widehat{f}\\ \widehat{\Sigma}_f\end{array}$$

$$\Uparrow \qquad\qquad\qquad\qquad \Downarrow$$

$$\begin{array}{c}\widehat{h}\\ \widehat{\Sigma}_h\\ \widehat{z}\end{array} \Leftarrow \boxed{\begin{array}{l} \widehat{\alpha}_j = \frac{1}{2}+\alpha, \widehat{\beta}_j = \beta + \frac{1}{2} + \|g - h * f\|^2 \\ \widehat{z}_j^{(k)} = \frac{\widehat{\beta}_j}{\widehat{\alpha}_j} \\ F = \texttt{Convmtx}(f) \\ \text{Tr}\left\{H\Sigma_f H'\right\} = \|D_h h\|^2 \\ \widehat{h} = (F'F + \lambda_h C_h' C_h + D_h' D_h)^{-1} F'g \\ \widehat{\Sigma}_h = v_\epsilon(H'H + \lambda_h C_h' C_h)^{-1} \end{array}} \Leftarrow \begin{array}{c}\widehat{f}\\ \widehat{\Sigma}_f\end{array}$$

4. CONCLUSIONS

In this paper, we considered the blind deconvolution problem in a Bayesian framework. Then, first, using Gaussian priors, we compared three main algorithms based on JMAP, BEM and VBA giving some insight and comparison between these methods. Then, using a Gaussian prior for the PSF but a Student-t prior for the images to enhance the sparsity, again, we compared those three methods.

The main conclusion can be summarized as follows:

- In JMAP, at each iteration, only the value of the previous estimate of h and f are transferred to the next step. So, in one hand, the computational cost of this approach is low because there is no need for matrix inversion. At the other hand, we do not know a lot about the convergence and the properties of the obtained solution.

- In EM, the value of the IRF h is transferred, but for f, its expected value and its uncertainty (covariance matrix) are transferred for the next iteration computation of h. So, in one hand, the computational cost of this approach is higher than JMAP because here we need the computation of Σ_f which needs a huge matrix inversion. At the other hand, we know a little more about the convergence (to local maximum of the marginal likelihood) and the properties of the obtained solution.

- In VBA, at each step, not only the values of the estimates, but also theirs uncertainties (in fact the whole approximated marginal laws) are transferred. So, in one hand, the computational cost of this approach is still higher than BEM because here we need the computation of Σ_f and Σ_h which needs two huge dimensional matrix inversion. At the other hand, not only we get the estimates of f and h but also their approximated marginals $q_1(f)$ and $q_2(h)$, from which, we can compute any statistical properties of these estimates.

5. REFERENCES

[1] G. Demoment and R. Reynaud, "Fast minimum-variance deconvolution," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. ASSP-33, pp. 1324–1326, 1985.

[2] T. F. Chan and C. K. Wong, "Total variation blind deconvolution.," *IEEE Trans Image Process*, vol. 7, no. 3, pp. 370–375, 1998.

[3] Simone Fiori, "Fast fixed-point neural blind-deconvolution algorithm.," *IEEE Trans Neural Netw*, vol. 15, no. 2, pp. 455–459, Mar 2004.

[4] Liang Wei, Lei Hua-ming, and Que Pei-wen, "Sparsity enhancement for blind deconvolution of ultrasonic

signals in nondestructive testing application.," *Rev Sci Instrum*, vol. 79, no. 1, pp. 014901, Jan 2008.

[5] Haiyong Liao and Michael K Ng, "Blind deconvolution using generalized cross-validation approach to regularization parameter estimation.," *IEEE Trans Image Process*, vol. 20, no. 3, pp. 670–680, Mar 2011.

[6] Filip Sroubek and Peyman Milanfar, "Robust multichannel blind deconvolution via fast alternating minimization.," *IEEE Trans Image Process*, vol. 21, no. 4, pp. 1687–1700, Apr 2012.

[7] Yu-Wing Tai, Xiaogang Chen, Sunyeong Kim, Seon Joo Kim, Feng Li, Jie Yang, Jingyi Yu, Yasuyuki Matsushita, and Michael S Brown, "Nonlinear camera response functions and image deblurring: theoretical analysis and practice.," *IEEE Trans Pattern Anal Mach Intell*, vol. 35, no. 10, pp. 2498–2512, Oct 2013.

[8] Xiang Zhu and Peyman Milanfar, "Removing atmospheric turbulence via space-invariant deconvolution.," *IEEE Trans Pattern Anal Mach Intell*, vol. 35, no. 1, pp. 157–170, Jan 2013.

[9] Hanjie Pan and Thierry Blu, "An iterative linear expansion of thresholds for -based image restoration.," *IEEE Trans Image Process*, vol. 22, no. 9, pp. 3715–3728, Sep 2013.

[10] Thibault Lelore and Frdric Bouchara, "Fair: a fast algorithm for document image restoration.," *IEEE Trans Pattern Anal Mach Intell*, vol. 35, no. 8, pp. 2039–2048, Aug 2013.

[11] Jianlin Zhang, Qiheng Zhang, and Guangming He, "Blind deconvolution of a noisy degraded image.," *Appl Opt*, vol. 48, no. 12, pp. 2350–2355, Apr 2009.

[12] Joao Oliveira, Mario Figueiredo, and Jose Bioucas-Dias, "Parametric blur estimation for blind restoration of natural images: Linear uniform motion and out-of-focus.," *IEEE Trans Image Process*, Oct 2013.

[13] J. Idier and Y. Goussard, "Markov modeling for bayesian multi-channel deconvolution," *Proceedings of IEEE ICASSP*, p. 2, 1990.

[14] J. Zhang, "The mean field theory in em procedures for blind markov random field image restoration.," *IEEE Trans Image Process*, vol. 2, no. 1, pp. 27–40, 1993.

[15] Sevket Babacan, Jingnan Wang, Rafael Molina, and Aggelos Katsaggelos, "Bayesian blind deconvolution from differently exposed image pairs.," *IEEE Trans Image Process*, vol. 19, no. 11, Nov 2010.

[16] Hacheme Ayasso and Ali Mohammad-Djafari, "Joint NDT image restoration and segmentation using Gauss–Markov–Potts prior models and variational bayesian computation," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2265–2277, 2010.

[17] A Mohammad-Djafari, "Bayesian approach with prior models which enforce sparsity in signal and image processing," *EURASIP Journal on Advances in Signal Processing*, vol. Special issue on Sparse Signal Processing, pp. 2012:52, 2012.

[18] Esteban Vera, Miguel Vega, Rafael Molina, and Aggelos K Katsaggelos, "Iterative image restoration using nonstationary priors.," *Appl Opt*, vol. 52, no. 10, pp. D102–D110, Apr 2013.

[19] Rafael Molina, Javier Mateos, and Aggelos K Katsaggelos, "Blind deconvolution using a variational approach to parameter, image, and blur estimation.," *IEEE Trans Image Process*, vol. 15, no. 12, pp. 3715–3727, Dec 2006.

[20] Se Un Park, Nicolas Dobigeon, and Alfred O Hero, "Semi-blind sparse image reconstruction with application to mrfm.," *IEEE Trans Image Process*, vol. 21, no. 9, pp. 3838–3849, Sep 2012.

[21] Zhimin Xu and Edmund Y Lam, "Maximum a posteriori blind image deconvolution with huber-markov randomfield regularization.," *Opt Lett*, vol. 34, no. 9, pp. 1453–1455, May 2009.

[22] L. Blanco and L. M. Mugnier, "Marginal blind deconvolution of adaptive optics retinal images.," *Opt Express*, vol. 19, no. 23, pp. 23227–23239, Nov 2011.

[23] Siamak Yousefi, Nasser Kehtarnavaz, and Yan Cao, "Computationally tractable stochastic image modeling based on symmetric markov mesh random fields.," *IEEE Trans Image Process*, vol. 22, no. 6, pp. 2192–2206, Jun 2013.

[24] N. N. Abdelmalek, T. Kasvand, and J. P. Croteau, "Image restoration for space invariant pointspread functions.," *Appl Opt*, vol. 19, no. 7, pp. 1184–1189, Apr 1980.

[25] Wided Souidene, Karim Abed-Meraim, and Azeddine Beghdadi, "A new look to multichannel blind image deconvolution.," *IEEE Trans Image Process*, vol. 18, no. 7, pp. 1487–1500, Jul 2009.

[26] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman, "Understanding blind deconvolution algorithms.," *IEEE Trans Pattern Anal Mach Intell*, Jul 2011.