Empirical Bayesian estimation for semi-blind inverse problems: Application to image deblurring with total variation regularisation

Charlesquin Kemajou Mbakam

Joint work with Marcelo Pereyra and Jean-Francois Giovannelli

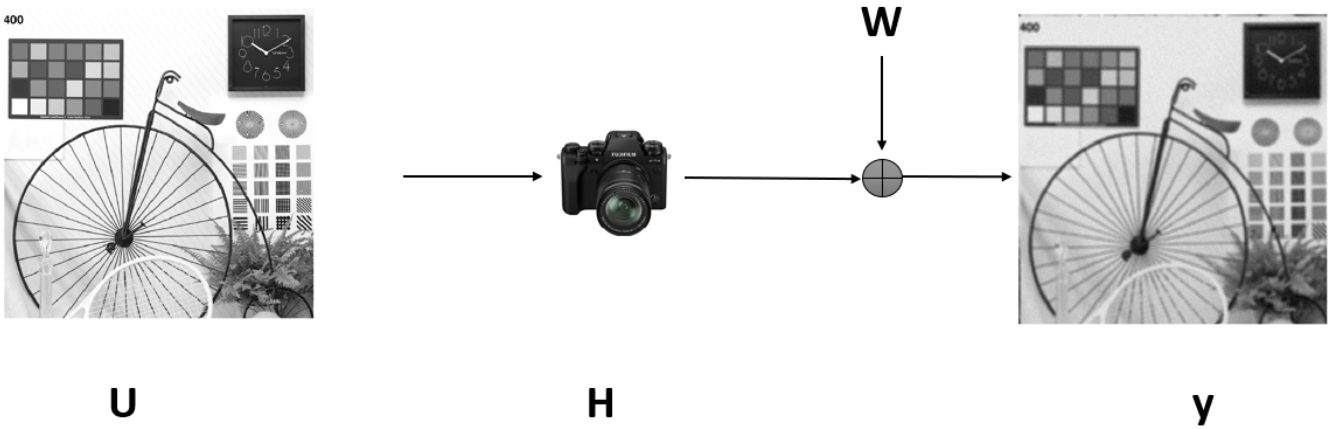
Maxwell Institue for Mathematical Sciences, Heriot-Watt University

Imaging with Uncertainty Quantification (IUQ), September 2022, Konventum

Outline

Introduction

- Image deblurring



Introduction

- Forward model:

$$y = Hu + w, \tag{1}$$

where,

  - $u \in \mathbb{R}^d$ unknown image, $y \in \mathbb{R}^d$ observed data and $d \in \mathbb{N}$,
  - H a circulant block matrix of dimension $d \times d$ obtained from a blur kernel h and ...
  - $w \sim \mathcal{N}(0, \sigma^2 Id)$ noise, $\sigma^2 > 0$.

- Deconvolution problems: Estimating u from y.

Deconvolution problems can be broadly classified in 3 groups: Non-blind, blind and semi-blind.

Introduction

$$y = Hu + w,$$

- Non-blind problems: The operator H is completely known and the image u is unknown.

  $\rightsquigarrow$ The problem is ill-posed so it requires regularisation to estimate u.

Introduction

$$y = Hu + w,$$

- Non-blind problems: The operator H is completely known and the image u is unknown.

  $\rightsquigarrow$ The problem is ill-posed so it requires regularisation to estimate u.

- Blind problems: The operator H and u are completely unknown.
  The problem is:

  $\rightsquigarrow$ Bilinear

  $\rightsquigarrow$ Ill-posed so it requires regularisation on both H and u.

Semi-blind deconvolution problem

- Semi-blind problems: $y = H(\alpha)u + w,$

  $\rightsquigarrow$ $H(\alpha) \in \mathcal{K}$ where,

$$\mathcal{K} = \left\{ H(\alpha) : \mathbb{R}^d \longrightarrow \mathbb{R}^d, \alpha \in \Theta_\alpha \right\}. \tag{2}$$

  - **Pros:** Specifying a parametric family introduces more structure into the estimation problem.
  - **Cons:** The problem is non linear w.r.t α

  $\rightsquigarrow$ The problem is ill-posed and regularisation on u is required.

Problem formulation in Bayesian framework

We formulated the problem in the Bayesian framework:

$\rightsquigarrow$ Likelihood function:

$$p(y|u,\alpha,\sigma^2) \propto \exp\left(-f^y_{\alpha,\sigma^2}(u)\right), \tag{3}$$

with the data fidelity term $f^y_{\alpha,\sigma^2}(u) = \frac{||y-H(\alpha)u||^2}{2\sigma^2}$.

$\rightsquigarrow$ Prior knowledge :

$$p(u|\theta) = \exp\left(-\theta g(u)\right)/Z(\theta), \tag{4}$$

where $Z(\theta) = \int_{\mathbb{R}^d} e^{-\theta g(\tilde{u})} du$, $g(u) = ||u||_{TV} = ||(D^h u, D^v u)||$

- θ is the regularisation parameter of the model
- D^h and D^v are horizontal and vertical finite difference operators.

Problem formulation in Bayesian framework

$\rightsquigarrow$ Form Bayes theorem, the posterior distribution is

$$p(u|y,\theta,\alpha,\sigma^2) = p(y|u,\alpha,\sigma^2)p(u|\theta)/p(y|\theta,\alpha,\sigma^2), \tag{5}$$

where, $p(y|\theta,\alpha,\sigma^2)$ is the marginal likelihood defined by

$$p(y|\theta,\alpha,\sigma^2) = \int_{\mathbb{R}^d} p(y|\tilde{u},\alpha,\sigma^2)p(\tilde{u}|\theta)d\tilde{u}. \tag{6}$$

Problem formulation in Bayesian framework

$\rightsquigarrow$ Form Bayes theorem, the posterior distribution is

$$p(u|y,\theta,\alpha,\sigma^2) = p(y|u,\alpha,\sigma^2)p(u|\theta)/p(y|\theta,\alpha,\sigma^2), \tag{5}$$

where, $p(y|\theta,\alpha,\sigma^2)$ is the marginal likelihood defined by

$$p(y|\theta,\alpha,\sigma^2) = \int_{\mathbb{R}^d} p(y|\tilde{u},\alpha,\sigma^2)p(\tilde{u}|\theta)d\tilde{u}. \tag{6}$$

Goals

- Estimate θ, α and σ^2 from the observed data y.
- Restore the image u from the observed data y, given the estimates of θ, α and σ^2.

Importance of the forward operator

- Deconvolution with mismatch blur operator $H(\alpha)$

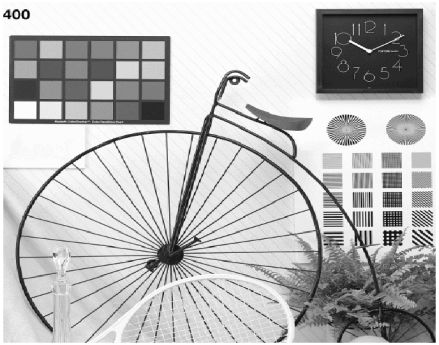
Importance of the forward operator

- Deconvolution with mismatch blur operator $H(\alpha)$

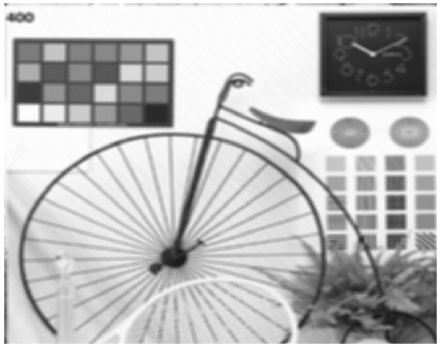
(a) u^* (b) blurred image, $\alpha = (0.4, 0.3)$

Importance of the forward operator

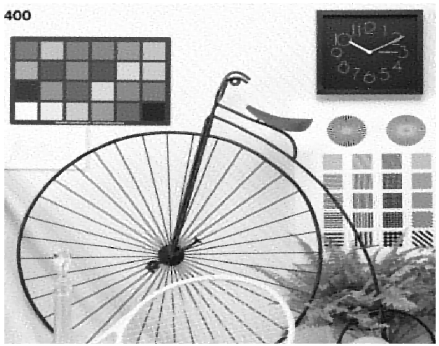
- Deconvolution with mismatch blur operator $H(\alpha)$



(a) u^*



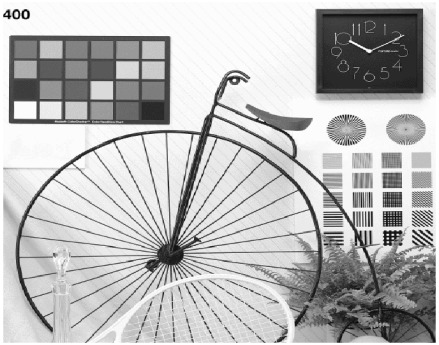
(b) blurred image, $\alpha = (0.4, 0.3)$



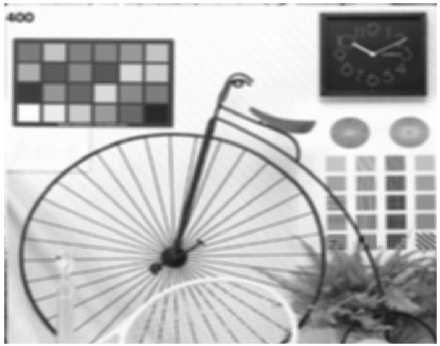
(c) MSE $= 26.31$, $\alpha = (0.4, 0.3)$

Importance of the forward operator

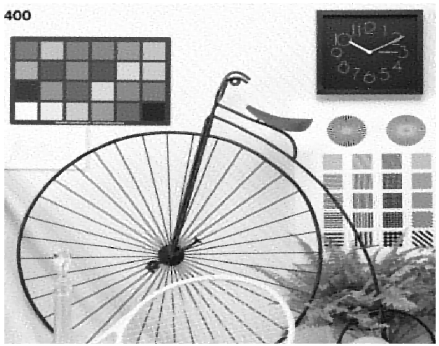
- Deconvolution with mismatch blur operator $H(\alpha)$



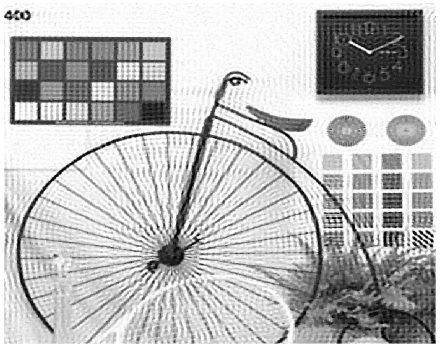
(a) u^*



(b) blurred image, $\alpha = (0.4, 0.3)$



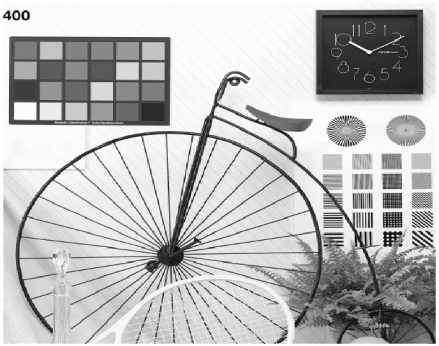
(c) MSE = 26.31, $\alpha = (0.4, 0.3)$



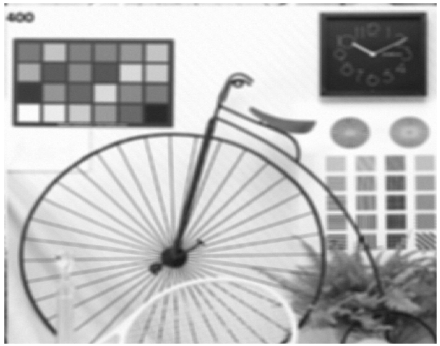
(d) MSE = 28.92, $\alpha = (0.2, 0.2)$

Importance of the forward operator

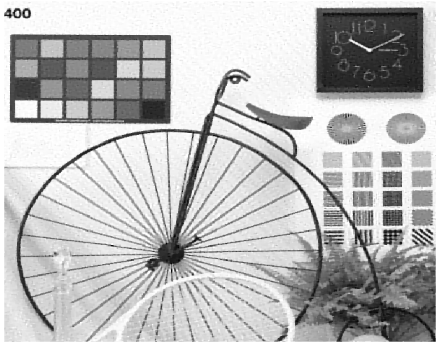
- Deconvolution with mismatch blur operator $H(\alpha)$



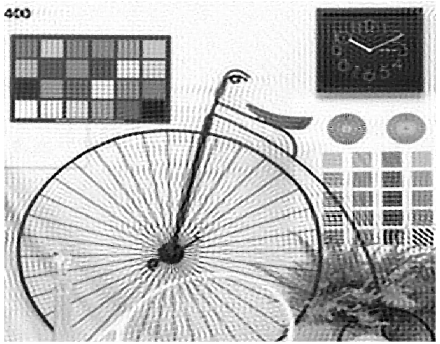
(a) u^*



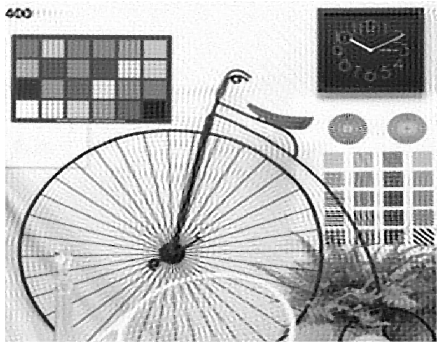
(b) blurred image, $\alpha = (0.4, 0.3)$



(c) MSE = 26.31, $\alpha = (0.4, 0.3)$



(d) MSE = 28.92, $\alpha = (0.2, 0.2)$



(e) MSE = 30.17, $\alpha = (0.1, 0.1)$

Methodology

Our proposed method is based on the empirical Bayesian approach:

1. We evaluate the Maximum Marginal likelihood estimator from y,

$$(\hat{\theta}, \hat{\alpha}, \hat{\sigma}^2) \in \underset{\theta \in \Theta_\theta, \alpha \in \Theta_\alpha, \sigma \in \Theta_{\sigma^2}}{\operatorname{argmax}} p(y|\theta, \alpha, \sigma^2).$$

where

$$p(y|\theta, \alpha, \sigma^2) = \int_{\mathbb{R}^d} p(y|\tilde{u}, \alpha, \sigma^2) p(\tilde{u}|\theta) d\tilde{u}.$$

Methodology

Our proposed method is based on the empirical Bayesian approach:

1. We evaluate the Maximum Marginal likelihood estimator from y,

$$(\hat{\theta}, \hat{\alpha}, \hat{\sigma}^2) \in \underset{\theta \in \Theta_\theta, \alpha \in \Theta_\alpha, \sigma \in \Theta_{\sigma^2}}{\operatorname{argmax}} p(y|\theta, \alpha, \sigma^2).$$

where

$$p(y|\theta, \alpha, \sigma^2) = \int_{\mathbb{R}^d} p(y|\tilde{u}, \alpha, \sigma^2) p(\tilde{u}|\theta) d\tilde{u}.$$

2. Given $\hat{\theta}$, $\hat{\alpha}$ and $\hat{\sigma}^2$, we estimate u by maximum-a-posteriori estimation using the pseudo posterior $p(u|y, \hat{\theta}, \hat{\alpha}, \hat{\sigma}^2)$

$$\hat{u}_{MAP} = \underset{\tilde{u} \in \mathbb{R}^d}{\operatorname{argmin}} \left\{ f^y_{\hat{\alpha}, \hat{\sigma}^2}(\tilde{u}) + \hat{\theta} g(\tilde{u}) \right\}. \tag{7}$$

Step 1: Projected gradient descent (PGD)

Assuming $\Theta_\theta, \Theta_\alpha$ and Θ_{σ^2} are convex sets, and Π_S the projection operator on the convex set S. The projected gradient descent (PGD) algorithm can be used to estimate θ, α and σ^2 as follows:

$$\begin{cases} \theta_{n+1} &= \Pi_{\Theta_\theta}\left[\theta_n + \delta_{n+1}\nabla_\theta \log(p(y|\theta,\alpha,\sigma^2))\right] \\ \alpha_{n+1} &= \Pi_{\Theta_\alpha}\left[\alpha_n + \delta_{n+1}\nabla_\alpha \log(p(y|\theta,\alpha,\sigma^2))\right] \\ \sigma^2_{n+1} &= \Pi_{\Theta_{\sigma^2}}\left[\sigma^2_n + \delta_{n+1}\nabla_{\sigma^2} \log(p(y|\theta,\alpha,\sigma^2))\right] \end{cases}, \qquad \forall n \in \mathbb{N}.$$

Step 1: Projected gradient descent (PGD)

Assuming $\Theta_\theta, \Theta_\alpha$ and Θ_{σ^2} are convex sets, and Π_S the projection operator on the convex set S. The projected gradient descent (PGD) algorithm can be used to estimate θ, α and σ^2 as follows:

$$\begin{cases} \theta_{n+1} &= \Pi_{\Theta_\theta}\left[\theta_n + \delta_{n+1}\nabla_\theta \log(p(y|\theta,\alpha,\sigma^2))\right] \\ \alpha_{n+1} &= \Pi_{\Theta_\alpha}\left[\alpha_n + \delta_{n+1}\nabla_\alpha \log(p(y|\theta,\alpha,\sigma^2))\right] \\ \sigma^2_{n+1} &= \Pi_{\Theta_{\sigma^2}}\left[\sigma^2_n + \delta_{n+1}\nabla_{\sigma^2} \log(p(y|\theta,\alpha,\sigma^2))\right] \end{cases}, \qquad \forall n \in \mathbb{N}.$$

Question : Can we apply the PGD algorithm?

Step 1: Projected gradient descent (PGD)

- $p(y|u, \theta, \alpha, \sigma^2)$ is analytically and computationally intractable.

$$p(y|u, \theta, \alpha, \sigma^2) = \int_{\mathbb{R}^d} p(y|\tilde{u}, \alpha, \sigma^2) p(\tilde{u}|\theta) d\tilde{u}.$$

- $\nabla_\theta \log p(y|u, \theta, \alpha, \sigma^2)$, $\nabla_\alpha \log p(y|u, \theta, \alpha, \sigma^2)$ and $\nabla_{\sigma^2} \log p(y|u, \theta, \alpha, \sigma^2)$ are analytically and computationally intractable.

Then, we can't apply the PGD algorithm to solve (7).

Step 1: Projected gradient descent (PGD)

- $p(y|u, \theta, \alpha, \sigma^2)$ is analytically and computationally intractable.

$$p(y|u, \theta, \alpha, \sigma^2) = \int_{\mathbb{R}^d} p(y|\tilde{u}, \alpha, \sigma^2) p(\tilde{u}|\theta) d\tilde{u}.$$

- $\nabla_\theta \log p(y|u, \theta, \alpha, \sigma^2)$, $\nabla_\alpha \log p(y|u, \theta, \alpha, \sigma^2)$ and $\nabla_{\sigma^2} \log p(y|u, \theta, \alpha, \sigma^2)$ are analytically and computationally intractable.

Then, we can't apply the PGD algorithm to solve (7).

Question: How to address this challenge??

Step 1: Fisher's Identity

Using Fisher's Identity, we obtain:

Step 1: Fisher's Identity

Using Fisher's Identity, we obtain:

$$\nabla_\theta \log p(y|\theta, \alpha, \sigma^2) = -\nabla_\theta \log Z(\theta) - \underbrace{\int_{\mathbb{R}^d} g(u) p(u|y, \theta, \alpha, \sigma^2) du}$$

Step 1: Fisher's Identity

Using Fisher's Identity, we obtain:

$$\nabla_\theta \log p(y|\theta, \alpha, \sigma^2) = -\nabla_\theta \log Z(\theta) - \underbrace{\int_{\mathbb{R}^d} g(u) p(u|y, \theta, \alpha, \sigma^2) du}_{\mathbb{E}_{u \sim p(u|y,\theta,\alpha,\sigma^2)}[g(u)]}$$

Step 1: Fisher's Identity

Using Fisher's Identity, we obtain:

$$\nabla_\theta \log p(y|\theta,\alpha,\sigma^2) = -\nabla_\theta \log Z(\theta) - \underbrace{\int_{\mathbb{R}^d} g(u)p(u|y,\theta,\alpha,\sigma^2)du}_{\mathbb{E}_{u\sim p(u|y,\theta,\alpha,\sigma^2)}[g(u)]}$$

$$\nabla_\alpha \log p(y|\theta,\alpha,\sigma^2) = -\underbrace{\int_{\mathbb{R}^d} \nabla_\alpha f^y_{\alpha,\sigma^2}(u)p(u|y,\theta,\alpha,\sigma^2)du}_{\mathbb{E}_{u\sim p(u|y,\theta,\alpha,\sigma^2)}\left[\nabla_\alpha f^y_{\alpha,\sigma^2}(u)\right]}$$

$$\nabla_{\sigma^2} \log p(y|\theta,\alpha,\sigma^2) = -\underbrace{\int_{\mathbb{R}^d} \nabla_{\sigma^2} f^y_{\alpha,\sigma^2}(u)p(u|y,\theta,\alpha,\sigma^2)du}_{\mathbb{E}_{u\sim p(u|y,\theta,\alpha,\sigma^2)}\left[\nabla_{\sigma^2} f^y_{\alpha,\sigma^2}(u)\right]} - \frac{d}{2\sigma^2}$$

Step 1: Gradients approximation

Using the Markov chain $(X_k)_{k\in\mathbb{N}}$ [Durmus et al.2018] generated with MYULA

$$X_{k+1} = (1 - \frac{\gamma}{\lambda})X_k - \gamma\nabla_x f^y_{\alpha,\sigma^2}(X_k) + \frac{\gamma}{\lambda} prox^{\lambda}_{g_\theta}(X_k) + \sqrt{2\gamma}Z_{k+1} \quad (8)$$

where $\lambda > 0, g_\theta = \theta g$ and,

$$prox^{\lambda}_{g_\theta}(u) = \underset{v}{\operatorname{argmin}}\, g_\theta(v) + \frac{1}{2\lambda}||u - v||_2^2 \quad (9)$$

Step 1: Gradients approximation

Using the Markov chain $(X_k)_{k\in\mathbb{N}}$ [Durmus et al.2018] generated with MYULA

$$X_{k+1} = (1 - \frac{\gamma}{\lambda})X_k - \gamma\nabla_x f^y_{\alpha,\sigma^2}(X_k) + \frac{\gamma}{\lambda} prox^{\lambda}_{g_\theta}(X_k) + \sqrt{2\gamma}Z_{k+1} \quad (8)$$

where $\lambda > 0, g_\theta = \theta g$ and,

$$prox^{\lambda}_{g_\theta}(u) = \underset{v}{\operatorname{argmin}}\, g_\theta(v) + \frac{1}{2\lambda}||u - v||_2^2 \quad (9)$$

We obtain the following approximation

- $\mathbb{E}_{u\sim p(u|y,\theta,\alpha,\sigma^2)}\left[g(u)\right] \approx \frac{1}{m}\sum_{k=0}^{m} g(X_k)$
- $\mathbb{E}_{u\sim p(u|y,\theta,\alpha,\sigma^2)}\left[\nabla_\alpha f^y_{\alpha,\sigma^2}(u)\right] \approx \frac{1}{m}\sum_{k=0}^{m} \nabla_\alpha f^y_{\alpha,\sigma^2}(X_k)$
- $\mathbb{E}_{u\sim p(u|y,\theta,\alpha,\sigma^2)}\left[\nabla_{\sigma^2} f^y_{\alpha,\sigma^2}(u)\right] \approx \frac{1}{m}\sum_{k=0}^{m} \nabla_{\sigma^2} f^y_{\alpha,\sigma^2}(X_k)$

Step 1: Gradients approximation

Then, we obtain the following approximation gradients:

- $\nabla_\theta \log p(y|\theta, \alpha, \sigma^2) \approx -\frac{1}{m}\sum_{k=0}^{m} g(X_k) - \frac{d}{\theta}$
- $\nabla_\alpha \log p(y|\theta, \alpha, \sigma^2) \approx -\frac{1}{m}\sum_{k=0}^{m} \nabla_\alpha f^y_{\alpha,\sigma^2}(X_k)$
- $\nabla_{\sigma^2} \log p(y|\theta, \alpha, \sigma^2) \approx -\frac{1}{m}\sum_{k=0}^{m} \nabla_{\sigma^2} f^y_{\alpha,\sigma^2}(X_k) - \frac{d}{2\sigma^2}$

Step 1: Stochastic Approximate Proximal Gradient (SAPG) algorithm

Putting everything together, we obtain the following SAPG algorithm

Algorithm 3.1 SAPG Algorithm

Initialization: $\{\theta_0, \alpha_0, \sigma_0^2, X_0^0\}$, set $\Theta_\theta, \Theta_\alpha,$ and Θ_{σ^2}, define γ, λ and N.
for $n = 0 : N-1$ **do**
  if $n > 0$ **then**
    set $X_0^n = X_{m_n-1}^{n-1}$
  end if
  for $k = 0 : m_n - 1$ **do**
    $X_{k+1}^n = (1 - \frac{\gamma}{\lambda})X_k - \gamma \nabla_x f^y_{\alpha_n, \sigma_n^2}(X_k^n) + \frac{\gamma}{\lambda}\text{prox}^{\lambda}_{\theta_n g}(X_k^n) + \sqrt{2\gamma} Z_{k+1}$
  end for
  $\theta_{n+1} = \Pi_{\Theta_\theta}\left[\theta_n + \delta_{n+1}\sum_{k=1}^{m_n}\left\{\frac{d}{\beta\theta_n} - g(X_k^n)\right\}/m_n\right]$
  $\alpha_{n+1} = \Pi_{\Theta_\alpha}\left[\alpha_n - \delta_{n+1}\sum_{k=1}^{m_n}\nabla_\alpha f^y_{\alpha_n,\sigma_n^2}(X_k^n)/m_n\right]$
  $\sigma^2_{n+1} = \Pi_{\Theta_{\sigma^2}}\left[\sigma_n^2 - \delta_{n+1}\sum_{k=1}^{m_n}\nabla_{\sigma^2} f^y_{\alpha_n,\sigma_n^2}(X_k^n)/m_n\right]$
end for
$\theta_N = \sum_{n=1}^N \omega_n\theta_n/\mathcal{N}$, $\alpha_N = \sum_{n=1}^N \omega_n\alpha_n/\mathcal{N}$ and $\sigma_N^2 = \sum_{n=1}^N \omega_n\sigma_n^2/\mathcal{N}$ where $\mathcal{N} = \sum_{n=1}^N \omega_n$

Figure 2: SAPG algorithm

Step 2: MAP estimation

Having $\hat{\theta}$, $\hat{\alpha}$ and $\hat{\sigma^2}$, we obtain:

- $p(u|y, \hat{\theta}, \hat{\alpha}, \hat{\sigma^2})$ ...

- ... which is then used to recover u by computing the MAP estimation of the following optimisation problem

$$\hat{u}_{MAP} = \underset{\tilde{u} \in \mathbb{R}^d}{\operatorname{argmin}} \left\{ f^y_{\hat{\alpha}, \hat{\sigma^2}}(\tilde{u}) + \hat{\theta} g(\tilde{u}) \right\}.$$

Illustration of the method

Experiment 1

We consider:

- Parametric Gaussian class of convolutional operator:

$$h_{\alpha_h,\alpha_v}(x,y) = \frac{\alpha_h \alpha_v}{2\pi} \exp\left(-\frac{1}{2}\left({\alpha_h}^2 x^2 + {\alpha_v}^2 y^2\right)\right), \ \forall x, y \in \mathbb{R},$$

where $\alpha_h, \alpha_v \in \mathbb{R}^+$ are horizontal and vertical bandwidth inverse.

We consider:

- Parametric Gaussian class of convolutional operator:

$$h_{\alpha_h,\alpha_v}(x,y) = \frac{\alpha_h \alpha_v}{2\pi} \exp\left(-\frac{1}{2}\left({\alpha_h}^2 x^2 + {\alpha_v}^2 y^2\right)\right), \ \forall x, y \in \mathbb{R},$$

  where $\alpha_h, \alpha_v \in \mathbb{R}^+$ are horizontal and vertical bandwidth inverse.

- Properties of h_{α_h,α_v} in the continuous domain:
  1. Probability density function

$$\int_{\mathbb{R}} \int_{\mathbb{R}} h_{\alpha_h,\alpha_v}(x,y) dx dy = 1, \quad \forall \alpha_v, \alpha_h \geq 0,$$

  2. For all $x, y \in \mathbb{R}$,

$$0 \leq h_{\alpha_h,\alpha_v}(x,y) \leq 1.$$

- In discrete domain we have, for all $i \in \mathcal{X}, j \in \mathcal{Y}$

$$h_{\alpha_h,\alpha_v}(i,j) = \frac{\alpha_h \alpha_v}{2\pi} \exp\left(-\frac{1}{2}\left({\alpha_h}^2 i^2 + {\alpha_v}^2 j^2\right)\right)$$

1. $\sum_{i\in\mathcal{X},j\in\mathcal{Y}} h_{\alpha_h,\alpha_v}(i,j) = 1,$ $\times$ Not satisfied
2. For all $i \in \mathcal{X}, j \in \mathcal{Y}$, $0 \leq h_{\alpha_h,\alpha_v}(i,j) \leq 1.$ ✓ Satisfied!!

- In discrete domain we have, for all $i \in \mathcal{X}, j \in \mathcal{Y}$

$$h_{\alpha_h,\alpha_v}(i,j) = \frac{\alpha_h \alpha_v}{2\pi} \exp\left(-\frac{1}{2}\left({\alpha_h}^2 i^2 + {\alpha_v}^2 j^2\right)\right)$$

1. $\sum_{i\in\mathcal{X},j\in\mathcal{Y}} h_{\alpha_h,\alpha_v}(i,j) = 1,$ $\times$ Not satisfied
2. For all $i \in \mathcal{X}, j \in \mathcal{Y}$, $0 \leq h_{\alpha_h,\alpha_v}(i,j) \leq 1.$ ✓ Satisfied!!

Correction of h_{α_h,α_v},

$$\tilde{h}_{\alpha_h,\alpha_v}(i,j) = \frac{\alpha_h \alpha_v}{2\pi} \exp\left(-\frac{1}{2}\left({\alpha_h}^2 i^2 + {\alpha_v}^2 j^2\right)\right) / C(\alpha_h, \alpha_v)$$

where $C(\alpha_h, \alpha_v) = \sum_{i\in\mathcal{X},j\in\mathcal{Y}} h_{\alpha_h,\alpha_v}(i,j)$

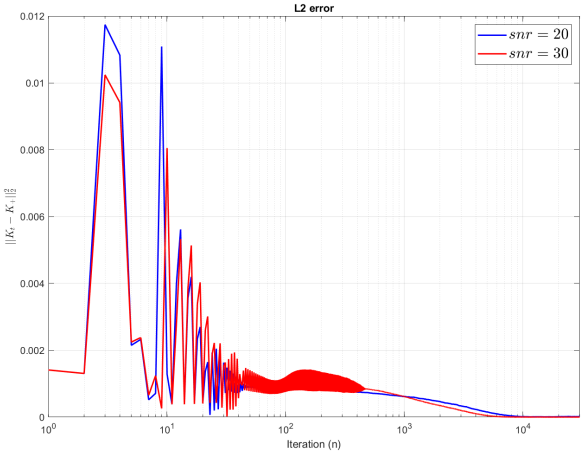
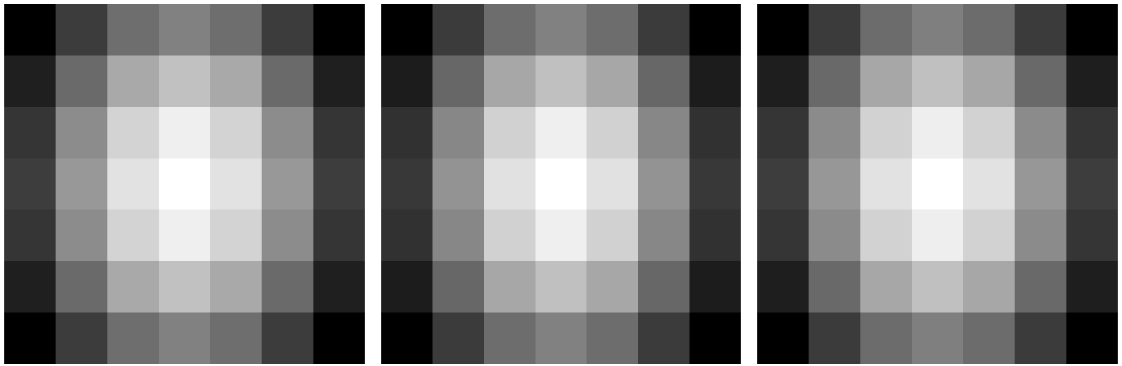
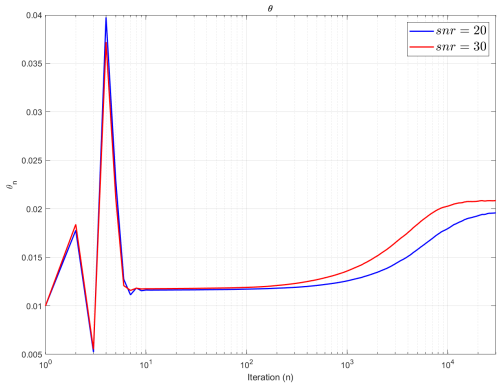


Figure 3: ℓ_2 error between the estimated PSF and the true PSF.

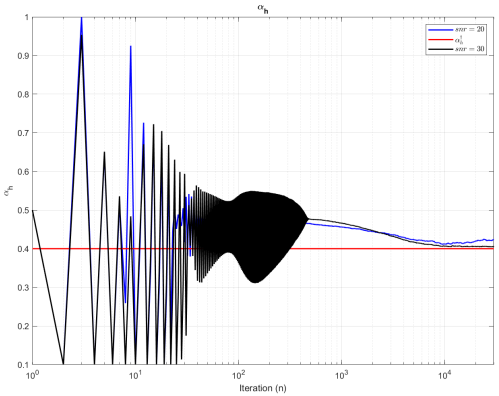


(a) True kernel (b) $h\hat{\alpha}$ SNR = 20dB (c) $h\hat{\alpha}$, SNR = 30dB

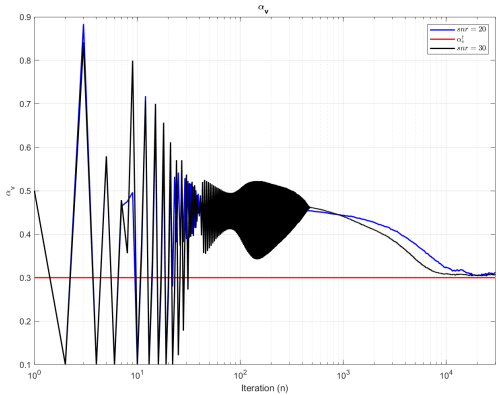
Gaussian kernel



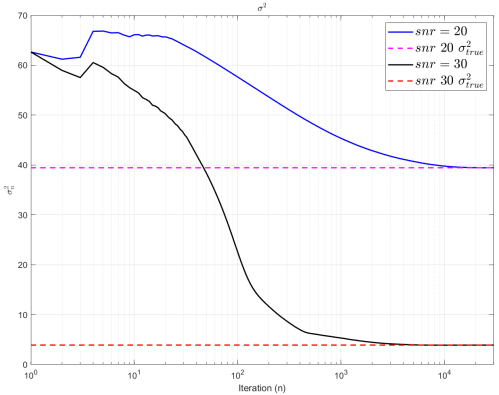
(a) θ



(b) α_h



(c) α_v



(d) σ^2

Comparison with state-of-the-art methods

Kernel	SNR	Para.	AAB	ALB	JF-FOB	Ours
Gaussian	$20dB$	$\alpha_h = 0.4$	$0.72 \pm 2.4 \times 10^{-2}$	$0.36 \pm 2.0 \times 10^{-2}$	$0.60 \pm 2.0 \times 10^{-2}$	$0.39 \pm 0.3 \times 10^{-2}$
		$\alpha_v = 0.3$	$0.67 \pm 1.4 \times 10^{-2}$	$0.25 \pm 3.0 \times 10^{-2}$	$0.26 \pm 4.0 \times 10^{-3}$	$0.29 \pm 2.0 \times 10^{-3}$
	$30dB$	$\alpha_h = 0.4$	$0.71 \pm 4.3 \times 10^{-2}$	$0.46 \pm 0.3 \times 10^{-2}$	$0.39 \pm 1.4 \times 10^{-4}$	$0.40 \pm 2.7 \times 10^{-4}$
		$\alpha_v = 0.3$	$0.61 \pm 3.1 \times 10^{-2}$	$0.48 \pm 5.0 \times 10^{-2}$	$0.27 \pm 2.3 \times 10^{-4}$	$0.31 \pm 3.4 \times 10^{-4}$

Figure 6: Mean and variance of the estimated parameters from 10 test images. AAB [Almeida and Almeida2009], ALB [Levin et al.2011], and JF-FOB [Orieux et al.2010].

Comparison with state-of-the-art methods

- ℓ_1-norm between h and $h(\hat{\alpha})$

Kernel	SNR	AAB	ALB	JF-FOB	Ours
Gaussian	$20dB$	$1.5 \times 10^{-1} \pm 2.1 \times 10^{-3}$	$3.3e{-2} \pm 5.1 \times 10^{-4}$	$8.8 \times 10^{-2} \pm 1.3 \times 10^{-3}$	$\mathbf{1.0 \times 10^{-2} \pm 1.2 \times 10^{-4}}$
	$30dB$	$1.4 \times 10^{-1} \pm 5.2 \times 10^{-3}$	$4.9 \times 10^{-2} \pm 1.6 \times 10^{-3}$	$5.0 \times 10^{-3} \pm 4.0 \times 10^{-6}$	$\mathbf{4.5 \times 10^{-3} \pm 8.8 \times 10^{-6}}$

Figure 7: Mean and variance of the estimated parameters from 10 test images

Comparison with state-of-the-art methods

- ℓ_1-norm between h and $h(\hat{\alpha})$

Kernel	SNR	AAB	ALB	JF-FOB	Ours
Gaussian	$20dB$	$1.5 \times 10^{-1} \pm 2.1 \times 10^{-3}$	$3.3e{-2} \pm 5.1 \times 10^{-4}$	$8.8 \times 10^{-2} \pm 1.3 \times 10^{-3}$	$\mathbf{1.0 \times 10^{-2} \pm 1.2 \times 10^{-4}}$
	$30dB$	$1.4 \times 10^{-1} \pm 5.2 \times 10^{-3}$	$4.9 \times 10^{-2} \pm 1.6 \times 10^{-3}$	$5.0 \times 10^{-3} \pm 4.0 \times 10^{-6}$	$\mathbf{4.5 \times 10^{-3} \pm 8.8 \times 10^{-6}}$

Figure 7: Mean and variance of the estimated parameters from 10 test images

- Time complexity

	AAB	ALB	JF-FOB	Ours
Gaussian	2.5	205	26.18	167.8

Figure 8: Average time complexity.

Given $\hat{\theta}, \hat{\alpha_h}, \hat{\alpha_v}$, and $\hat{\sigma^2}$ MAP estimate ([Afonso et al.2010])

(a) Original (b) Blurred (c) $\hat{u}$ with h (d) ours (e) JF-FOB

Table 1: Summary metrics for the estimated image to the $20dB$ setup

	observed y	Non-blind	Ours	[Orieux et al.2010]
MSE	45.6558	26.4087	**26.4466**	29.27
PSNR	14.8381	19.5931	**19.5806**	18.76
α_h	-	0.40	**0.42**	0.61
α_v	-	0.30	**0.30**	0.54

Given $\hat{\theta}, \hat{\alpha_h}, \hat{\alpha_v}$, and $\hat{\sigma^2}$ MAP estimate ([Afonso et al.2010])

(a) Original (b) Blurred (c) $\hat{u}$ with h (d) Ours (e) JF-FOB

Table 2: Summary metrics for the estimated image to the $30dB$ setup

	observed y	Non-blind	Ours	[Orieux et al.2010]
MSE	45.27	21.05	**21.05**	27.46
PSNR	14.91	21.56	**21.56**	20.57
α_h	-	0.40	**0.40**	0.38
α_v	-	0.30	**0.30**	0.26

Conclusion

To conclude:

- We proposed an empirical Bayesian method and SAPG algorithm to perform inference in semi-blind image deconvolution problems.

Conclusion

To conclude:

- We proposed an empirical Bayesian method and SAPG algorithm to perform inference in semi-blind image deconvolution problems.
- The proposed SAPG algorithm can be deployed to any semi-blind inverse problems that is convex w.r.t. the unknown image

Perspectives

Future works:

- Extend this work to blind deconvolution problems.

$$\hat{u}, \hat{h} \in \underset{(u,h)\in\mathbb{R}^d\times\mathbb{S}}{\operatorname{argmax}} \; p(u|y,h)p(u|\theta)p(h|\kappa)$$

where,

$$\mathbb{S} = \left\{ h \in \mathbb{R}^{p\times p}; \quad 0 \leq h_{ij} \leq 1 \text{ and } \sum_{i,j} h_{ij} = 1 \right\}$$

Perspectives

Future works:

- Extend this work to blind deconvolution problems.

$$\hat{u}, \hat{h} \in \underset{(u,h)\in\mathbb{R}^d\times\mathbb{S}}{\operatorname{argmax}} \; p(u|y,h)p(u|\theta)p(h|\kappa)$$

where,

$$\mathbb{S} = \left\{ h \in \mathbb{R}^{p\times p}; \quad 0 \leq h_{ij} \leq 1 \text{ and } \sum_{i,j} h_{ij} = 1 \right\}$$

- Improve the method in terms of complexity time and accuracy by using a data-driven prior.

References

Afonso, M. V., Bioucas-Dias, J. M., and Figueiredo, M. A. (2010).
Fast image recovery using variable splitting and constrained optimization.
IEEE transactions on image processing, 19(9):2345–2356.

Almeida, M. S. and Almeida, L. B. (2009).
Blind and semi-blind deblurring of natural images.
IEEE Transactions on Image Processing, 19(1):36–52.

Durmus, A., Moulines, E., and Pereyra, M. (2018).
Efficient bayseian computation by proximal markov chain monte carlo: when langevin meets moreau.
SIAM Journal on Imaging Sciences, 11(1):473–506.

Levin, A., Weiss, Y., Durand, F., and Freeman, W. T. (2011).
Efficient marginal likelihood optimization in blind deconvolution.
In *CVPR 2011*, pages 2657–2664. IEEE.

Orieux, F., Giovannelli, J.-F., and Rodet, T. (2010).
Bayesian estimation of regularization and point spread function parameters for wiener–hunt deconvolution.
JOSA A, 27(7):1593–1607.

Questions ?