SURE-based Blind Gaussian Deconvolution

Feng Xue and Thierry Blu
Department of Electronic Engineering, The Chinese University of Hong Kong (CUHK)

香港中文大學 The Chinese University of Hong Kong

Summary

- **Problem**: blind deconvolution without the knowledge of the Point Spread Function;
- **Basic procedure**: PSF estimation + non-blind deconvolution with estimated PSF;
- **Our scope**: Gaussian PSF with unknown variance s_0^2 (to be estimated);
- **Originality**: novel objective functional — blur SURE, a modified version of SURE (Stein's unbiased risk estimate);
- **Potential**: possibly extend SURE-based framework to other types of PSF with known parametric form.

Problem statement

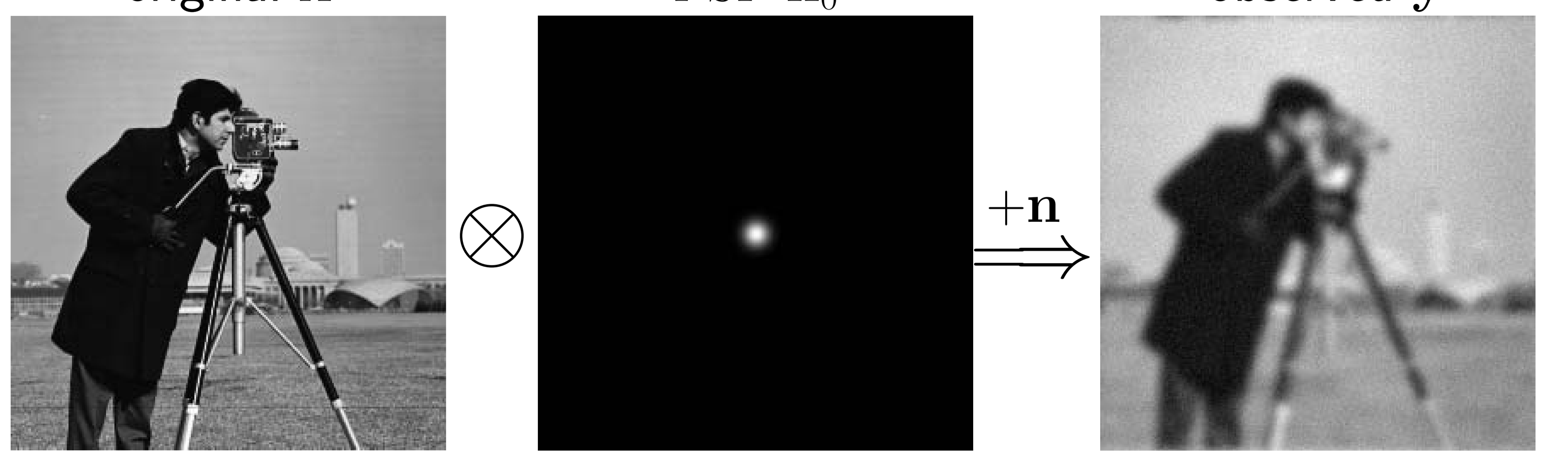
Linear observation model

$$\mathbf{y} = \mathbf{H}_0\mathbf{x} + \mathbf{n}$$

where

- $\mathbf{H}_0$ — the latent true convolution matrix associated with true PSF $\mathbf{h}_0$
- Gaussian noise $\mathbf{n} \sim \mathscr{N}(0, \sigma^2\mathbf{I})$

original $\mathbf{x}$ PSF $\mathbf{h}_0$ observed $\mathbf{y}$



Problem: $\mathbf{x} = ?$ and $\mathbf{h}_0 = ?$, knowing $\mathbf{y}$ only.

Solution — separate estimation of PSF, and then signal:

Step 1 — PSF estimation; **Step 2** — deconvolution⋆.

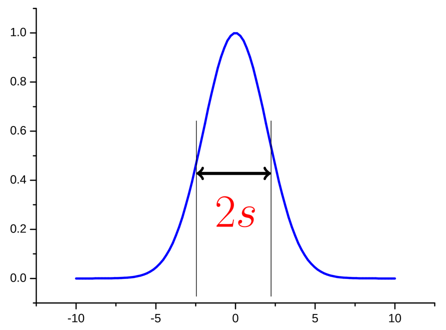
⋆ We use our recently proposed SURE-LET approach to perform (non-blind) deconvolution [1,2].

Gaussian kernel

- Parametric form with standard deviation s

$$\mathbf{h}(i,j;s) = C \cdot \exp\left(-\frac{i^2+j^2}{2s^2}\right)$$

s — blur size, width of the Gaussian shape;
C — normalization coefficient, s.t. $\sum_{i,j}\mathbf{h}(i,j) = 1$.



- $\mathbf{h}_0$ — latent true Gaussian kernel with unknown width s_0
- **Question**: how to estimate s_0, from observed $\mathbf{y}$?

[1]. F. Xue, F. Luisier, and T. Blu, SURE-LET image deconvolution using multiple Wiener filters, *ICIP 2012*.
[2]. F. Xue, F. Luisier, and T. Blu, Multi-Wiener SURE-LET Deconvolution, *submitted to IEEE TIP*.

Blur SURE as a new criterion

- **blur MSE** (mean squared error) is defined as (with unknown $\mathbf{H}_0\mathbf{x}$):

$$\text{blur MSE} = \frac{1}{N}\mathscr{E}\left\{\left\|\mathbf{HF}(\mathbf{y}) - \mathbf{H}_0\mathbf{x}\right\|^2\right\}$$

- **blur SURE** — unbiased estimate of the blur MSE:

$$\epsilon = \frac{1}{N}\left\|\mathbf{HF}(\mathbf{y}) - \mathbf{y}\right\|^2 + \frac{2\sigma^2}{N}\mathrm{div}_{\mathbf{y}}\left(\mathbf{HF}(\mathbf{y})\right) - \sigma^2$$

Remarks:

– the blur SURE depends on the observed data only (NOT on $\mathbf{H}_0$ and $\mathbf{x}$);
– divergence operator: $\mathrm{div}_{\mathbf{y}}\mathbf{u} = \sum_{n=1}^{N}\frac{\partial u_n}{\partial y_n}$ for $\forall \mathbf{u} \in \mathbb{R}^N$;
– Minimizing the blur-SURE yields results that are very close to minimizing the blur-MSE.

Blur-SURE minimization for Wiener processing

Theorem: Consider the approximate Wiener filtering:

$$\mathbf{F}(\mathbf{y}) = \underbrace{(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^{\mathrm{T}}}_{\mathbf{W}_{\mathbf{H},\lambda}}\mathbf{y}$$

Then, the minimization of the blur MSE over both $\mathbf{H}$ and λ:

$$\min_{\mathbf{H},\lambda}\overbrace{\frac{1}{N}\left\|\mathbf{HW}_{\mathbf{H},\lambda}\mathbf{y} - \mathbf{H}_0\mathbf{x}\right\|^2}^{\text{blur MSE}}$$

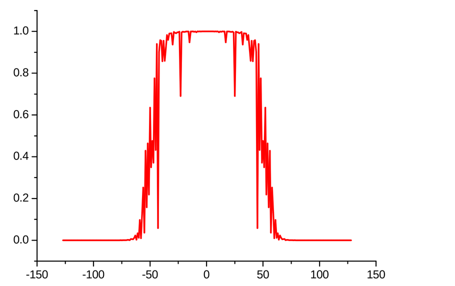
yields $\mathbf{H} \approx \mathbf{H}_0$.

Explanation (Fourier representation)

■ Consider the exact Wiener processing with known $H_0(\omega)$:

$$W(\omega) = \frac{H_0^*(\omega)}{|H_0(\omega)|^2 + \sigma^2/S(\omega)}$$

where $S(\omega)$ is the power spectrum density of image $\mathbf{x}$.
Then, $U_0(\omega) = H_0(\omega)W(\omega)$ behaves like a band indicator.

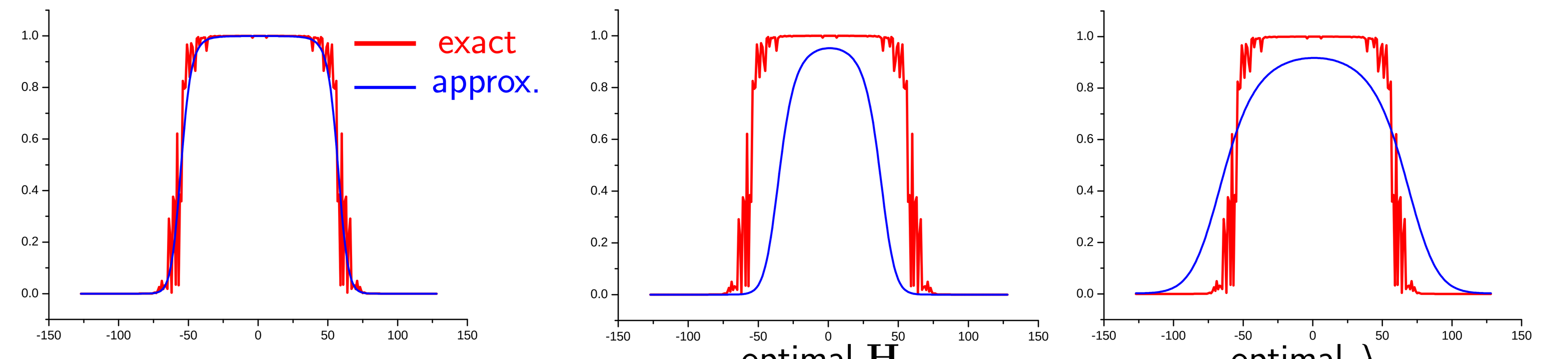


A typical example for $U_0(\omega)$

■ The blur-SURE minimization results in another band indicator $\mathbf{U} = \mathbf{HW}_{\mathbf{H},\lambda}$, which is as close as possible to $\mathbf{U}_0$:

$$\underbrace{\frac{|H_0(\omega)|^2}{|H_0(\omega)|^2 + \sigma^2/S(\omega)}}_{\text{exact } U_0(\omega)} \approx \underbrace{\frac{|H(\omega)|^2}{|H(\omega)|^2 + \lambda}}_{\text{approx. } U(\omega)} \quad \text{for } \forall\omega$$

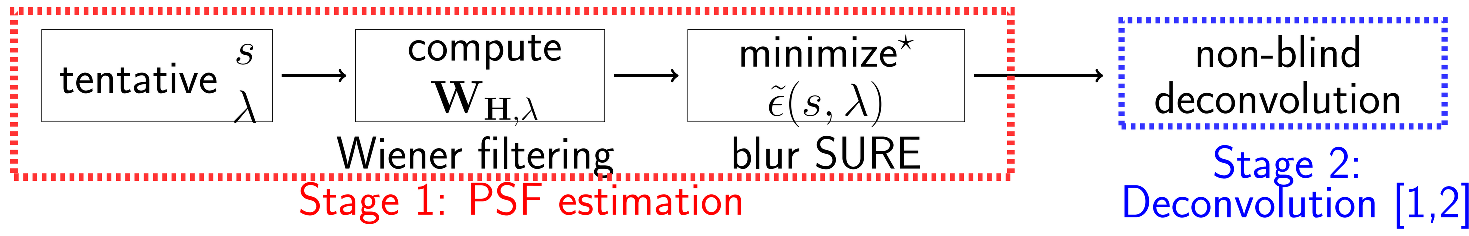
Approximation of the band indicator $U_0(\omega)$



optimal $\mathbf{H}$ and λ | optimal $\mathbf{H}$, non-optimal λ | optimal λ, non-optimal $\mathbf{H}$

Results and discussions

SURE-based framework to estimate s_0 and λ



⋆ One possibility is to use **alternating minimizations** between s and λ.

Estimation of s_0, followed by deconvolution

Table 1: Blind deconvolution (*Cameraman*)

BSNR (in dB)	40	30	20	10	40	30	20	10	40	30	20	10
true s_0	$s_0 = 1.0$				$s_0 = 2.0$				$s_0 = 3.0$			
estimated s_0	1.12	1.19	1.24	1.33	2.15	2.18	2.25	2.48	3.28	3.34	3.37	3.52
PSNR difference•	0.26	0.18	0.12	0.09	0.11	0.07	0.07	0.10	0.13	0.11	0.08	0.10

• PSNR difference after deconvolution with oracle.

Note that the PSNR loss due to the inexactness of the estimation is kept within 0.2dB.

Comparisons with the state-of-the-art in blind deconvolution

Table 2: SNR improvement (in dB) of deconvolution performance for $s_0^2 = 9$

Method	SAR1 [3]	SAR2 [3]	TV1 [4]	TV2 [4]	SURE
BSNR	40dB				
Cameraman	1.03	1.01	1.82	1.73	**3.15**
Lena	1.35	1.43	2.53	2.59	**4.54**
BSNR	20dB				
Cameraman	1.16	-8.83	1.70	-40.89	**2.15**
Lena	1.62	-11.32	2.62	-32.50	**3.13**

A visual example

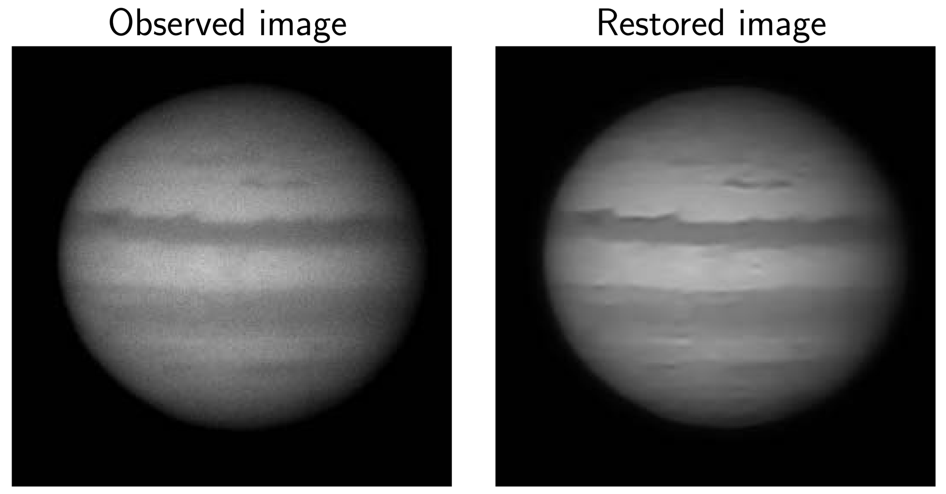
blurred with $s_0 = 1.50$, BSNR = 20dB, PSNR = 23.23dB | blind deconvolution with estimated $s_0 = 1.79$, PSNR = 25.65dB | non-blind deconvolution with known $s_0 = 1.50$, PSNR = 25.75dB

Real data

Restoration of *Jupiter*:

- The estimated noise std is $\sigma = 4.68$ by using MAD (median absolute deviation);
- Estimated $s_0 = 2.41$

Observed image | Restored image



[3]. R. Molina, J. Mateors, and A. Katsaggelos, *IEEE TIP*, vol.15, no.12, pp.3715–3727, 2006.
[4]. S. Babacan, R. Molina, and A. Katsaggelos, *IEEE TIP*, vol.18, no.1, pp.12–26, 2009