Image Restoration

Median filtering

Bilateral filtering

Deblurring

Lecturer: Sang Hwa Lee

Median Filter

Median Filtering (1)

- Median Filtering

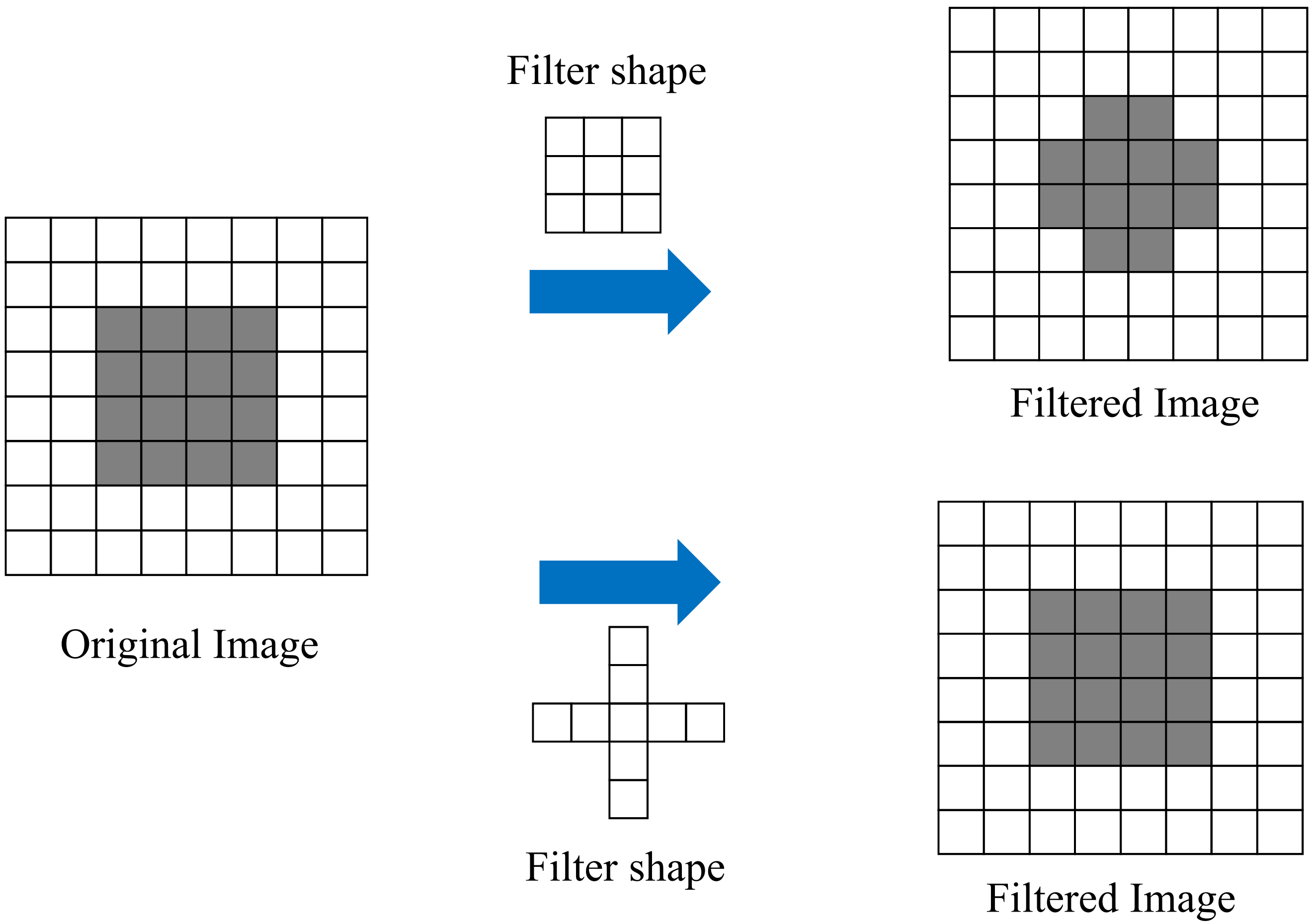
  $-v(m,n) = median \; \{y(m-k, n-l), (k,l) \in W\}$

  –filter length should be odd number

- median filter preserve discontinuities in a step function
- smooth a few pixels whose values differ significantly from the surrounding, without affecting the other pixels

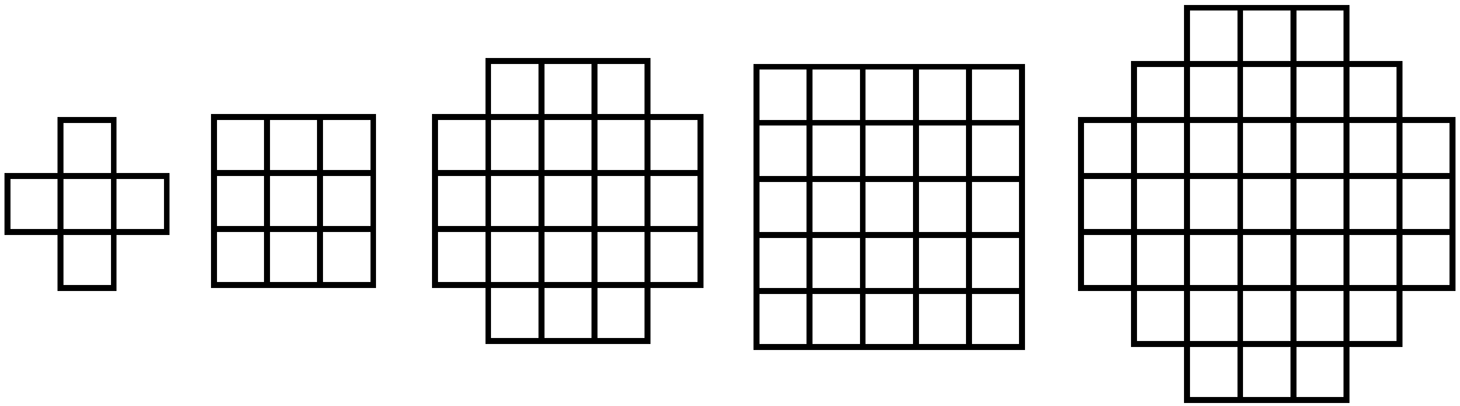
Median Filtering (2)

- 2-D median filtering
  - Comparing the shapes of median filters



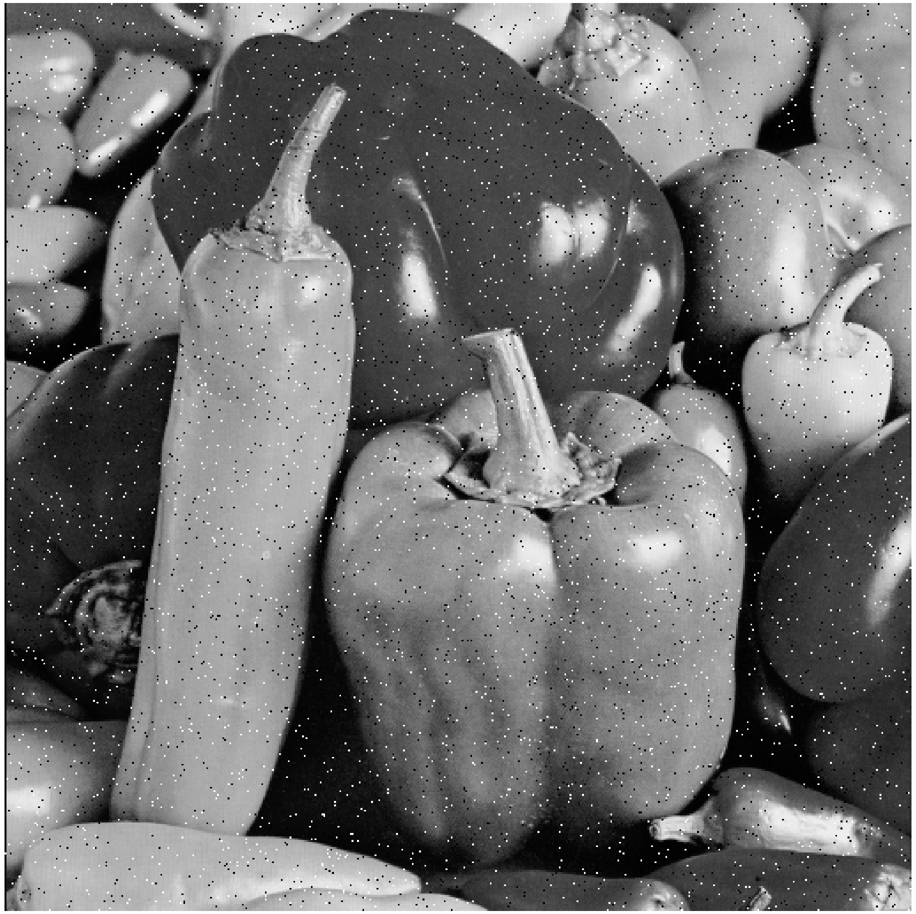
Median Filtering (3)

Various window sizes and shapes



a b c d e

Median Filtering Example (1)



Original

7by7 Median filtered image

Most proper to edge-preserving impulsive noise reduction

Median Filtering Example (2)

(a) Gaussian noise add
mean = 0 variance = 0.005

(b)3x3 median filter

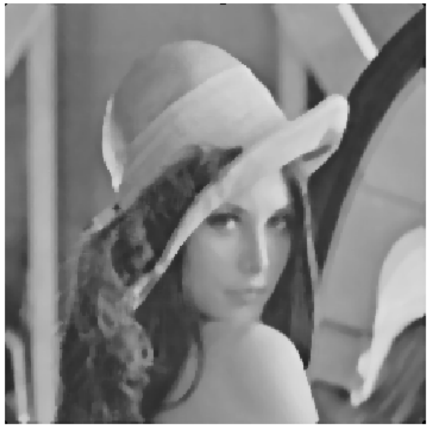
(c)5x5 median filter

(d)9x9 median filter

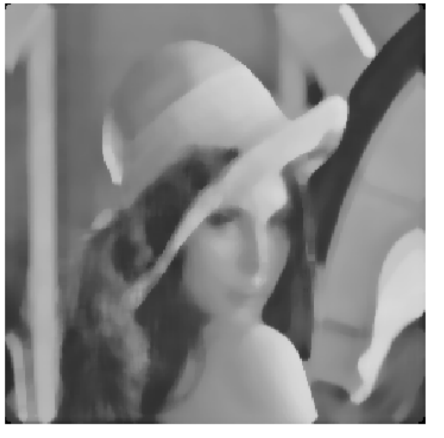
Median Filtering Example (3)

(a)impulsive noise add density = 0.02

(b) 3x3 median filter

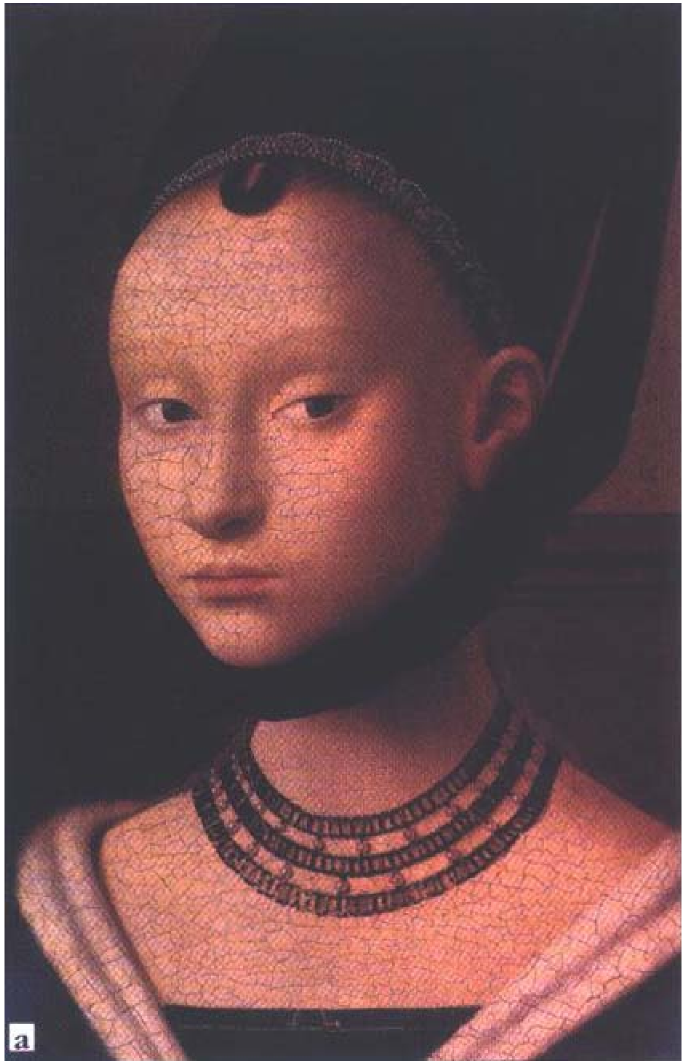


(c)5x5 median filter

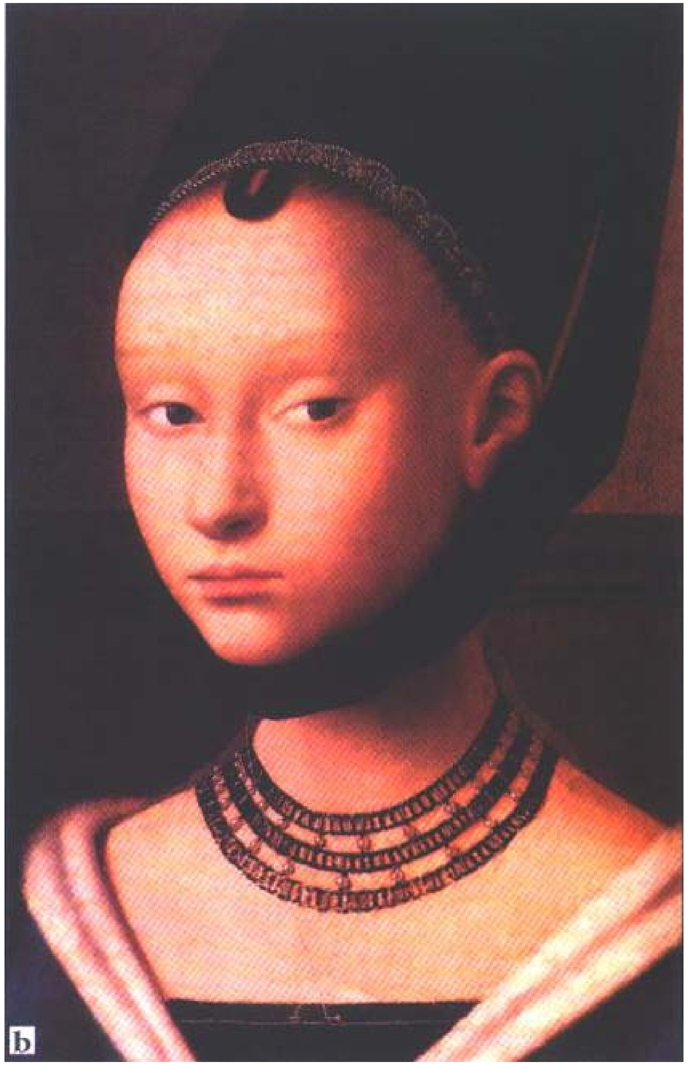


(d)9x9 median filter

Median Filtering Example (4)



Original image



Median filtering and color compensation

Bilateral Filter (Denoise Filter)

Noise reduction - Bilateral Filter

- Bilateral filter => Domain + Range
  - Two parameters
- Functions based on the parameters
  - Noise reduction
  - Smoothing
  - Segmentation

Bilateral Filter (1)

- Domain filter

$$\mathbf{h}(\mathbf{x}) = k_d^{-1}(\mathbf{x}) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{f}(\xi) c(\xi, \mathbf{x})\, d\xi$$

$$k_d(\mathbf{x}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, \mathbf{x})\, d\xi \; .$$

$$c(\xi, \mathbf{x}) = e^{-\frac{1}{2}\left(\frac{d(\xi, \mathbf{x})}{\sigma_d}\right)^2}$$

$$d(\xi, \mathbf{x}) = d(\xi - \mathbf{x}) = \|\xi - \mathbf{x}\|$$

Bilateral Filter (2)

- Range filter

$$\mathbf{h}(\mathbf{x}) = k_r^{-1}(\mathbf{x}) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{f}(\xi) s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x}))\, d\xi$$

$$k_r(\mathbf{x}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x}))\, d\xi$$

$$s(\xi, \mathbf{x}) = e^{-\frac{1}{2}\left(\frac{\delta(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x}))}{\sigma_r}\right)^2}$$

$$\delta(\phi, \mathbf{f}) = \delta(\phi - \mathbf{f}) = \|\phi - \mathbf{f}\|$$

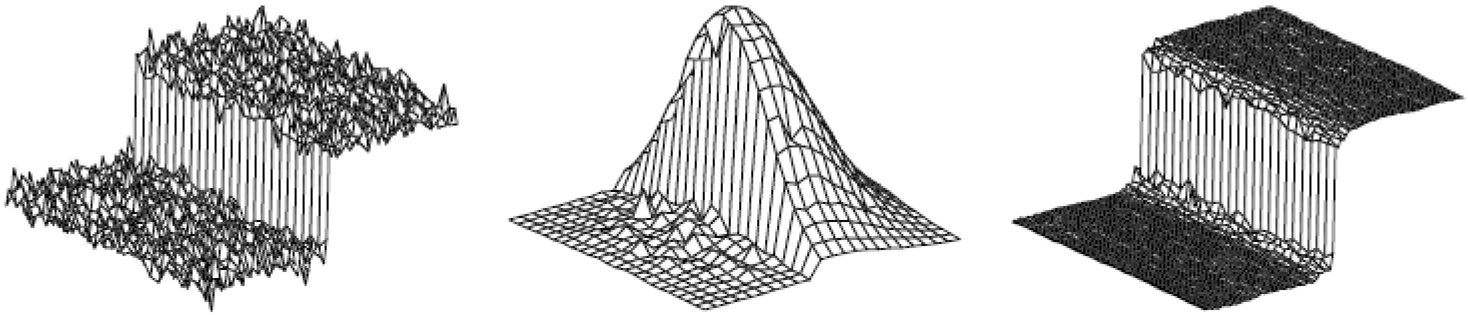
Bilateral Filter (3)

- Combining Domain and Range filters

$$\mathbf{h}(\mathbf{x}) = k^{-1}(\mathbf{x}) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{f}(\xi) c(\xi, \mathbf{x}) s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x}))\, d\xi$$

with the normalization

$$k(\mathbf{x}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, \mathbf{x}) s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x}))\, d\xi \ .$$



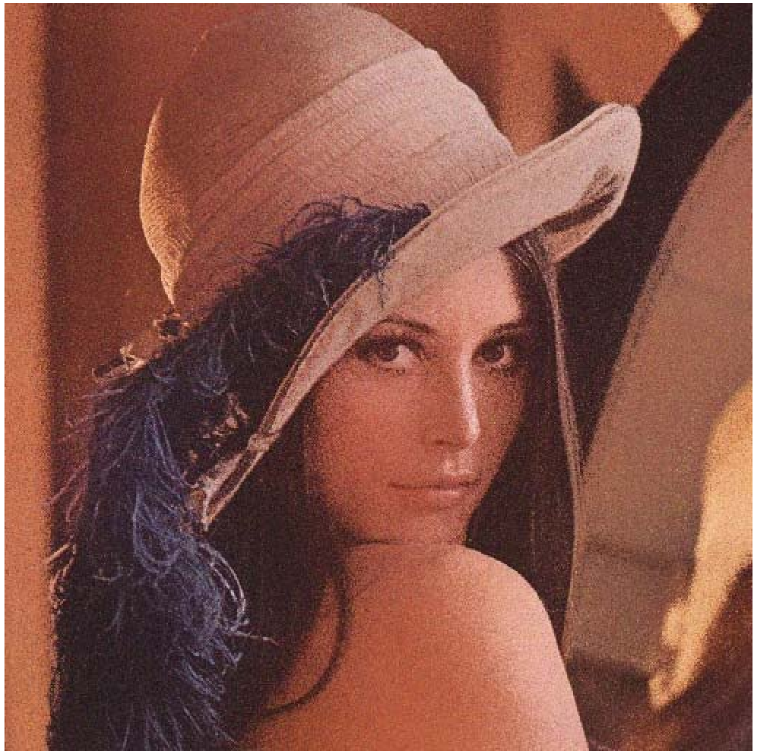
(a) (b) (c)

(a) Image with Gaussian noise, (b) bilateral weight at the boundary, (c) filtered image

Results (1)

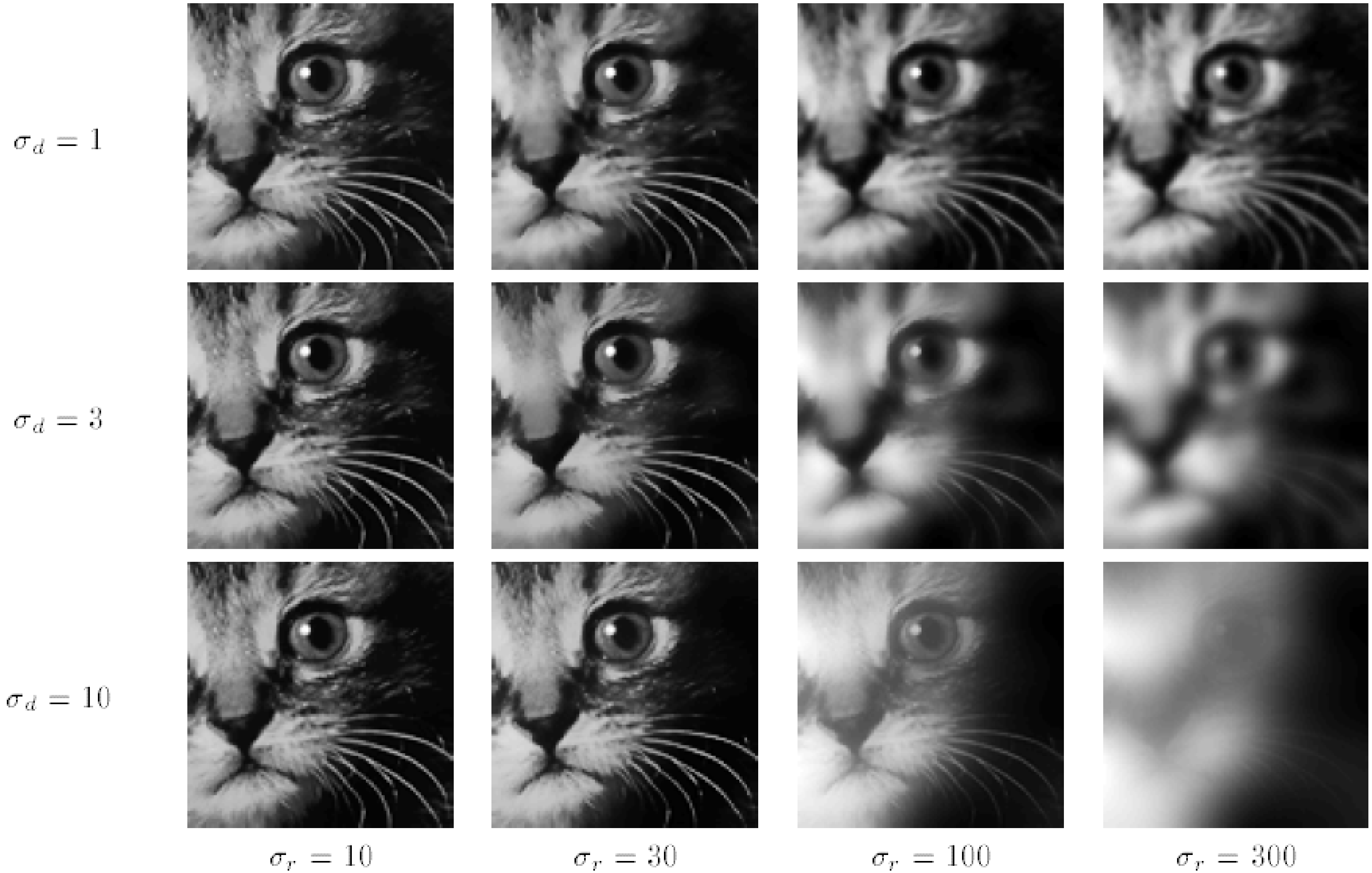
- σ_s : 30, σ_r : 70 in photoshop

PSNR : 18.42



PSNR :24.45

Results (6)


$\sigma_d = 1$
$\sigma_d = 3$
$\sigma_d = 10$
$\sigma_r = 10$
$\sigma_r = 30$
$\sigma_r = 100$
$\sigma_r = 300$

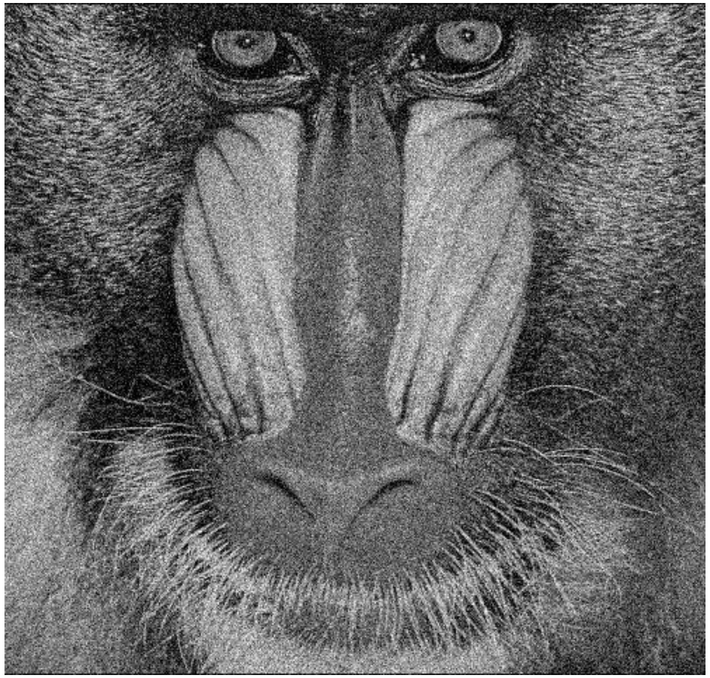
Adaptive Bilateral Filter (1)

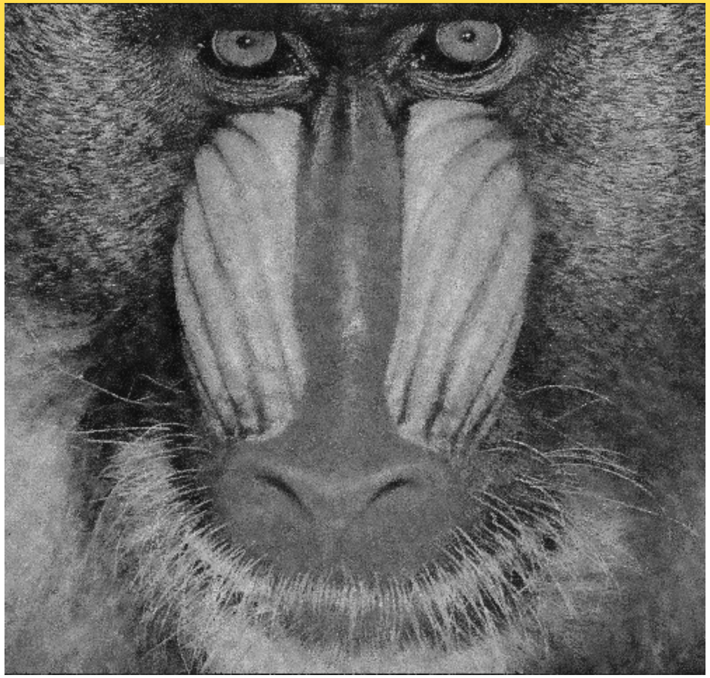
- Considering local variance of edges
  - Horizontal variance, Vertical variance
- Edge magnitude
  - The larger edge magnitudes are, the more is preserved edges
  - Strongly reduced when small edge maginitudes
- Edge orientation
  - Considering edge direction such that the pixels are averaged along the edges

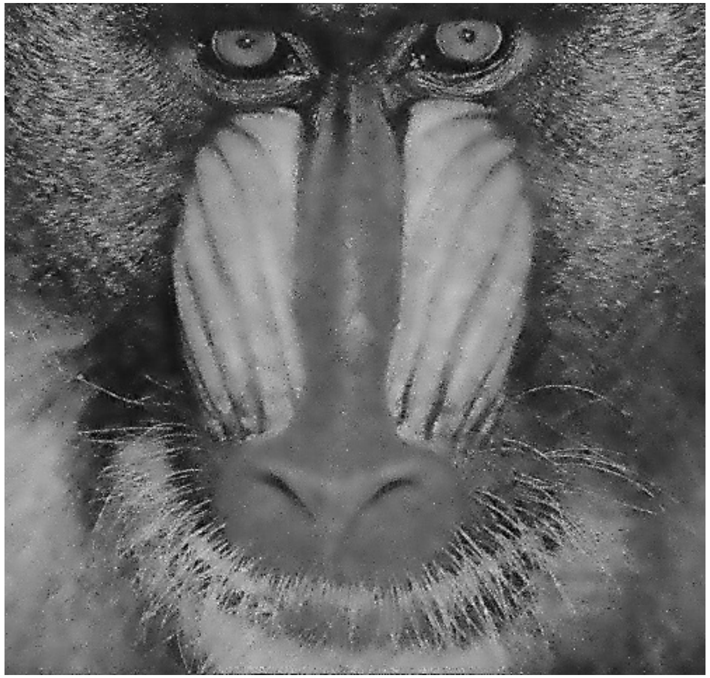
Comparison (1)

	SNR			SSIM		
	Bilateral filter		Adaptive	Bilateral filter		Adaptive
Lena	50.4877		51.4352	0.8914		0.8968
barbara	42.0069		47.5426	0.8815		0.9148
baboon	31.1805		34.0745	0.7666		0.7707
peppers	49.821		52.0161	0.8887		0.9184

Comparison (2)

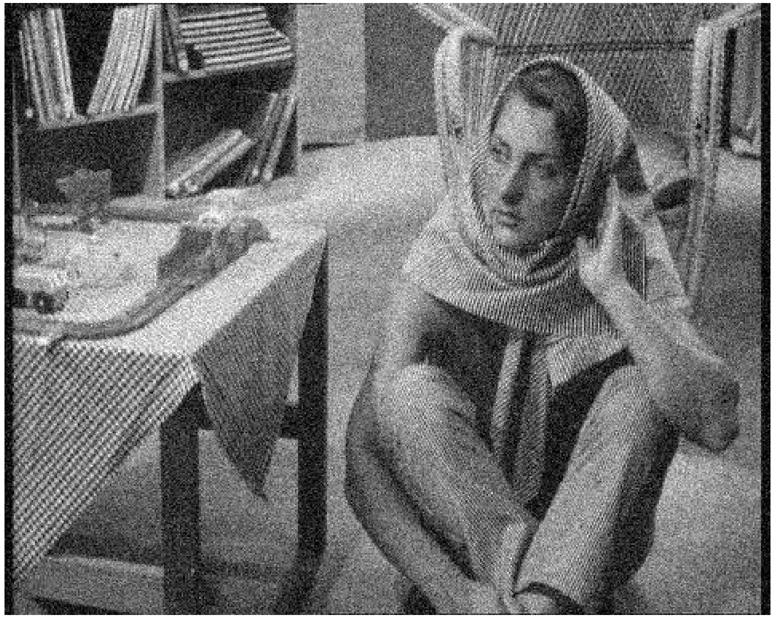


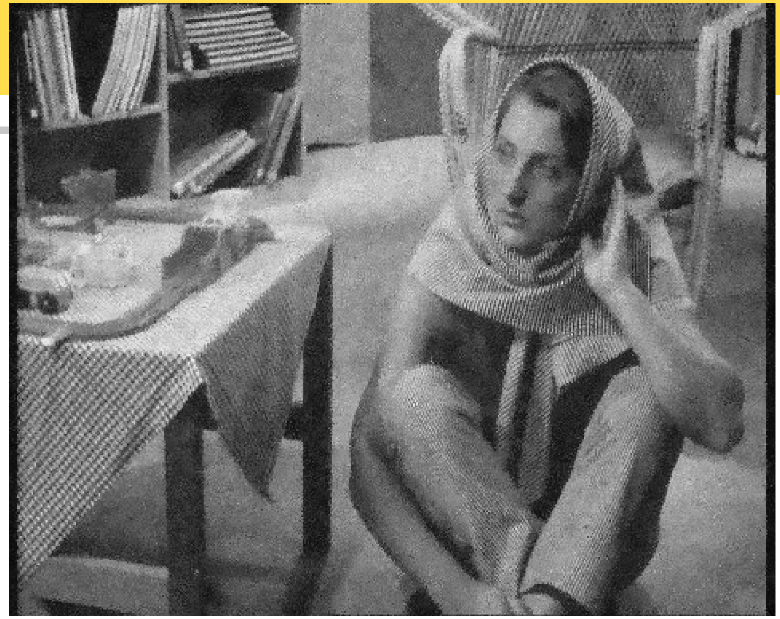




Noisy	**Bilateral**
	adaptive

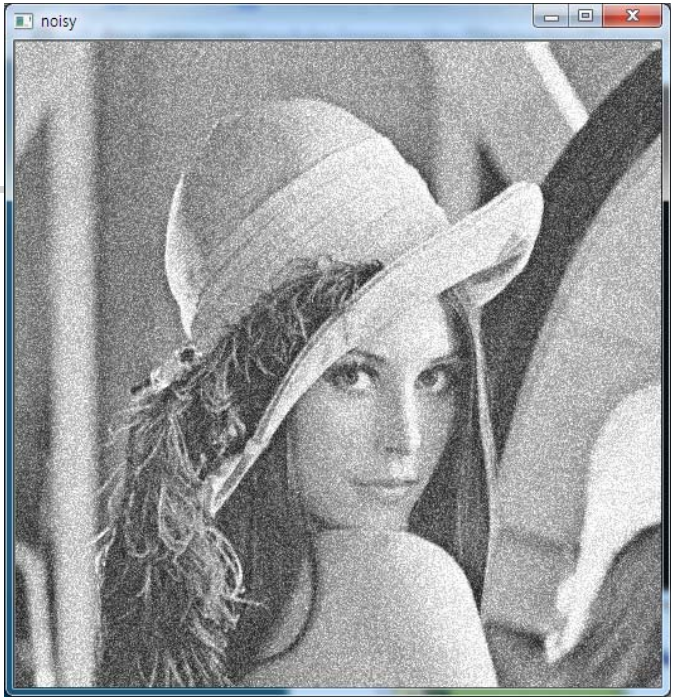
Comparison (3)



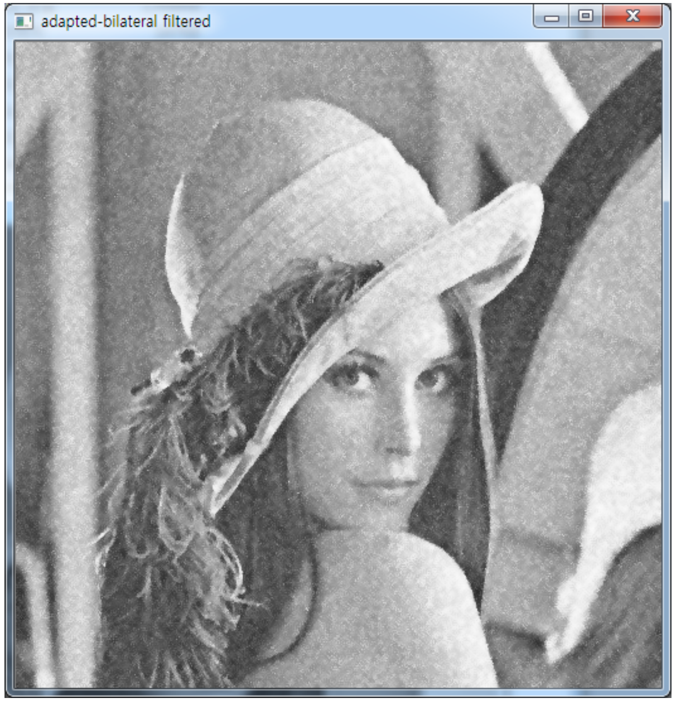


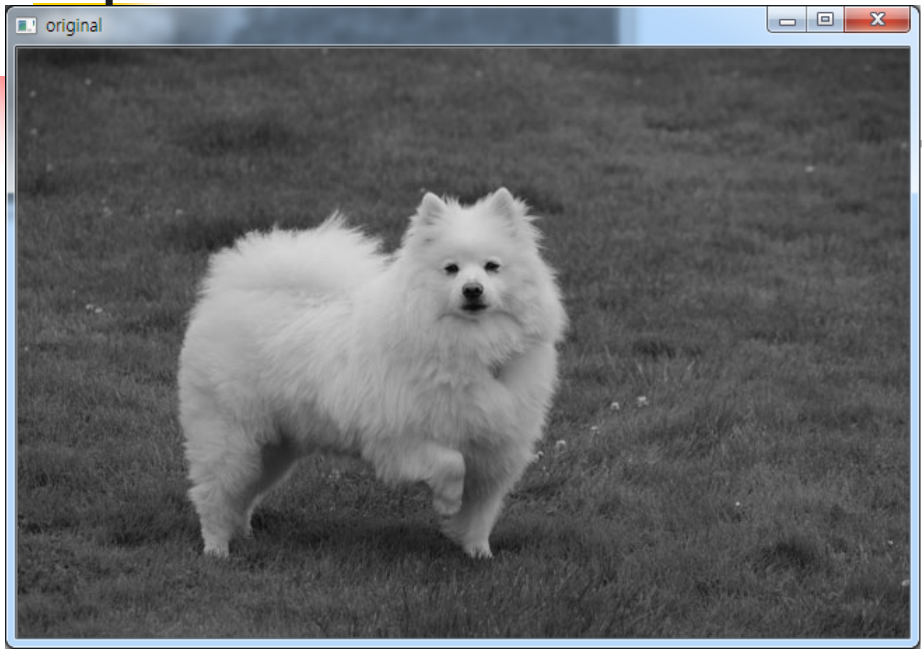
Noisy	**Bilateral**
	adaptive

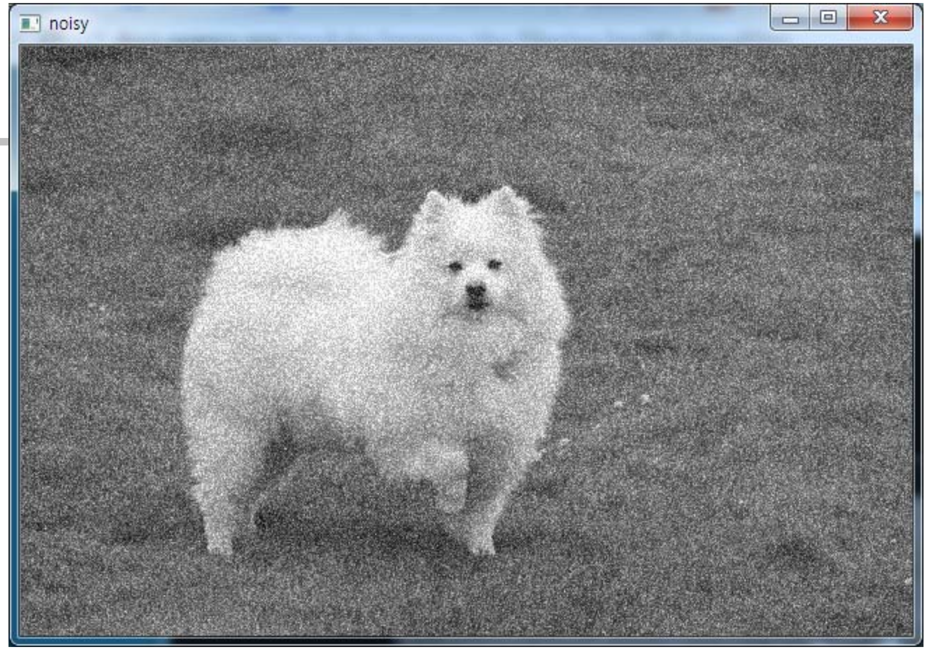

original

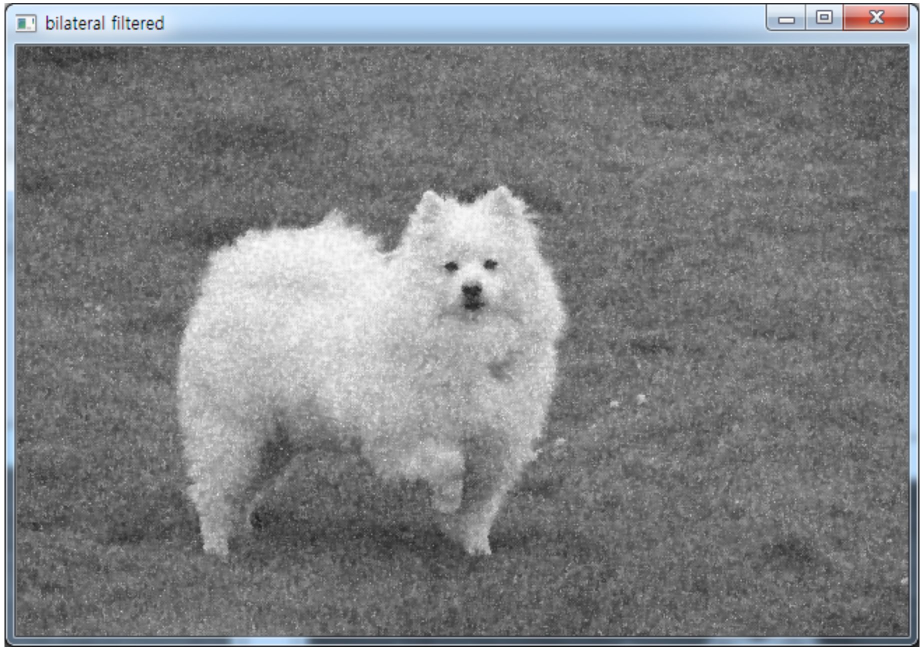

noisy


bilateral filtered


adapted-bilateral filtered


original


noisy


bilateral filtered

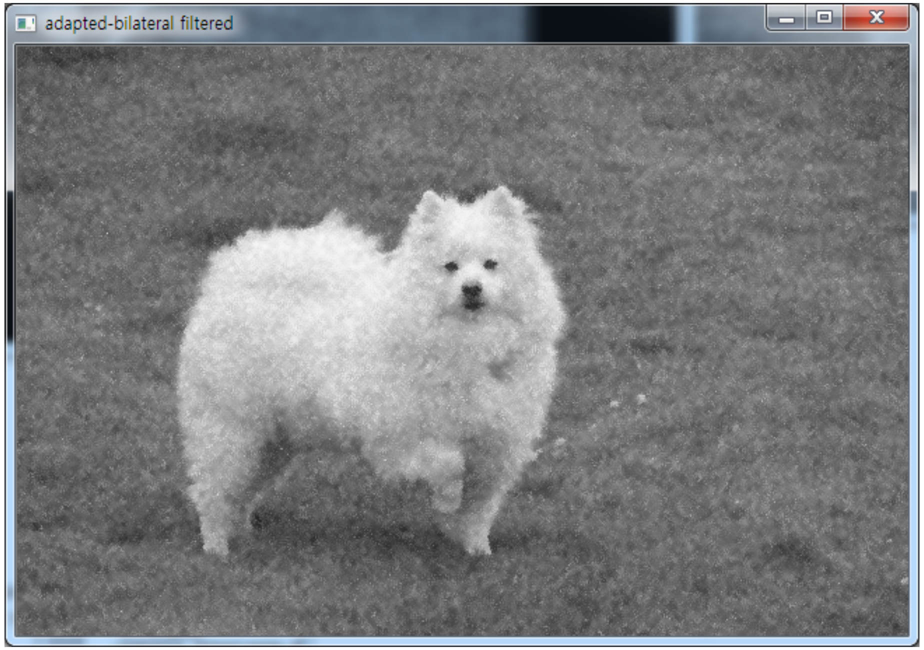
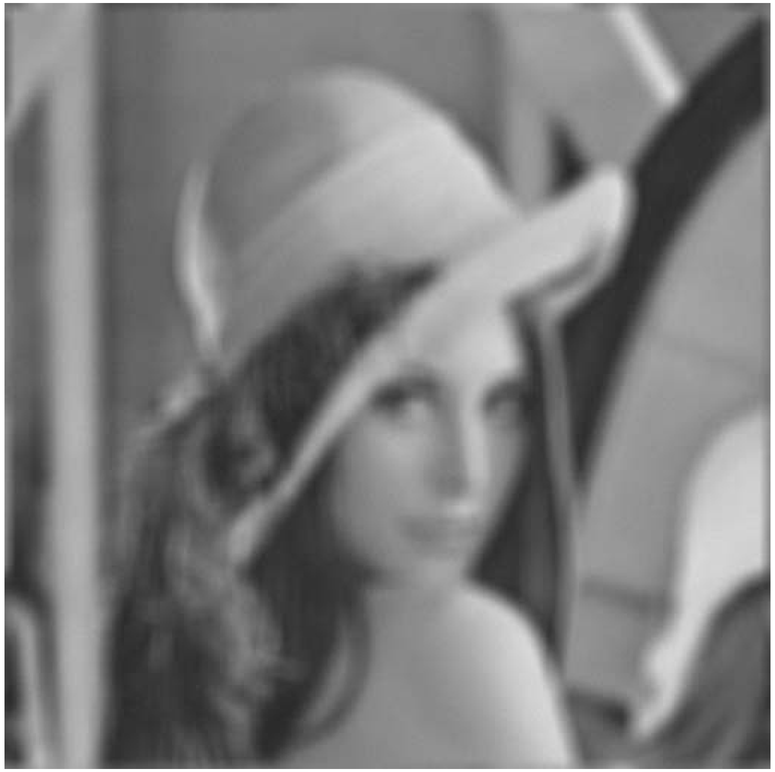

adapted-bilateral filtered

Image Restoration/Deblurring

Image Deblurring

- remove or minimize the degradations in image
  - determine the original image, given the observed image and knowledge about the degradation (**H**)



Degraded (blur + additive noise)

Restored (Iterative least squares)

Image Observation Model

⇒ Estimate or recover f(x,y) from g(x,y)

⇒ Deconvolution/Deblurring problem

$$g(x,y) = \iint h(x,y;\alpha,\beta) f(\alpha,\beta)\, d\alpha\, d\beta + n(x,y)$$

$$\text{or} \quad \mathbf{g} = \mathbf{Hf} + \mathbf{n} \quad ; \quad \text{discrete model}$$

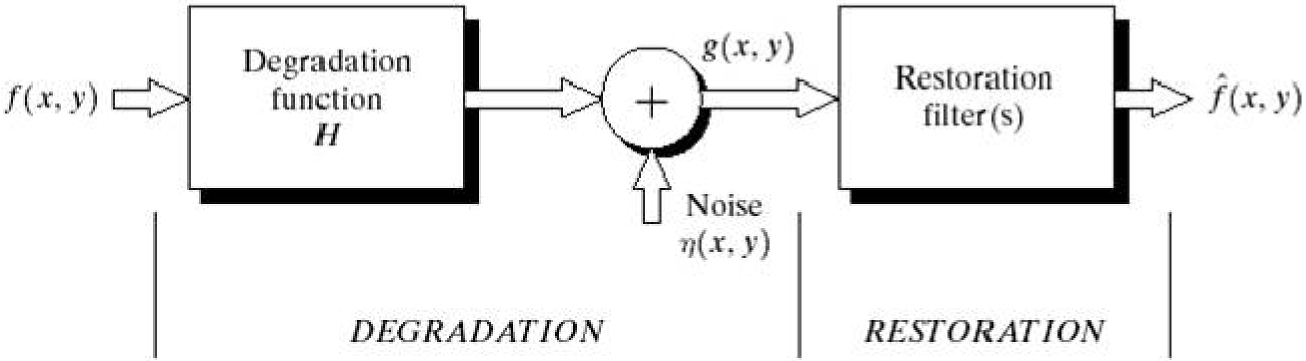
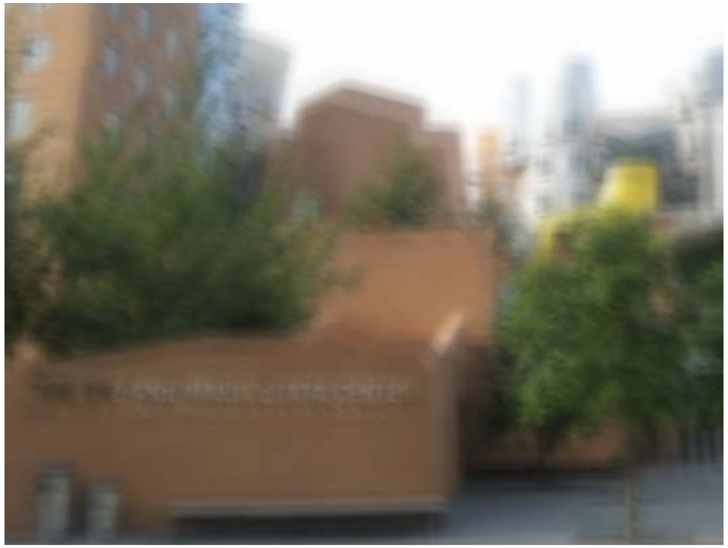


Image Formation Process



=

⊗

Blur kernel

Blurry image

Input to algorithm

Sharp image

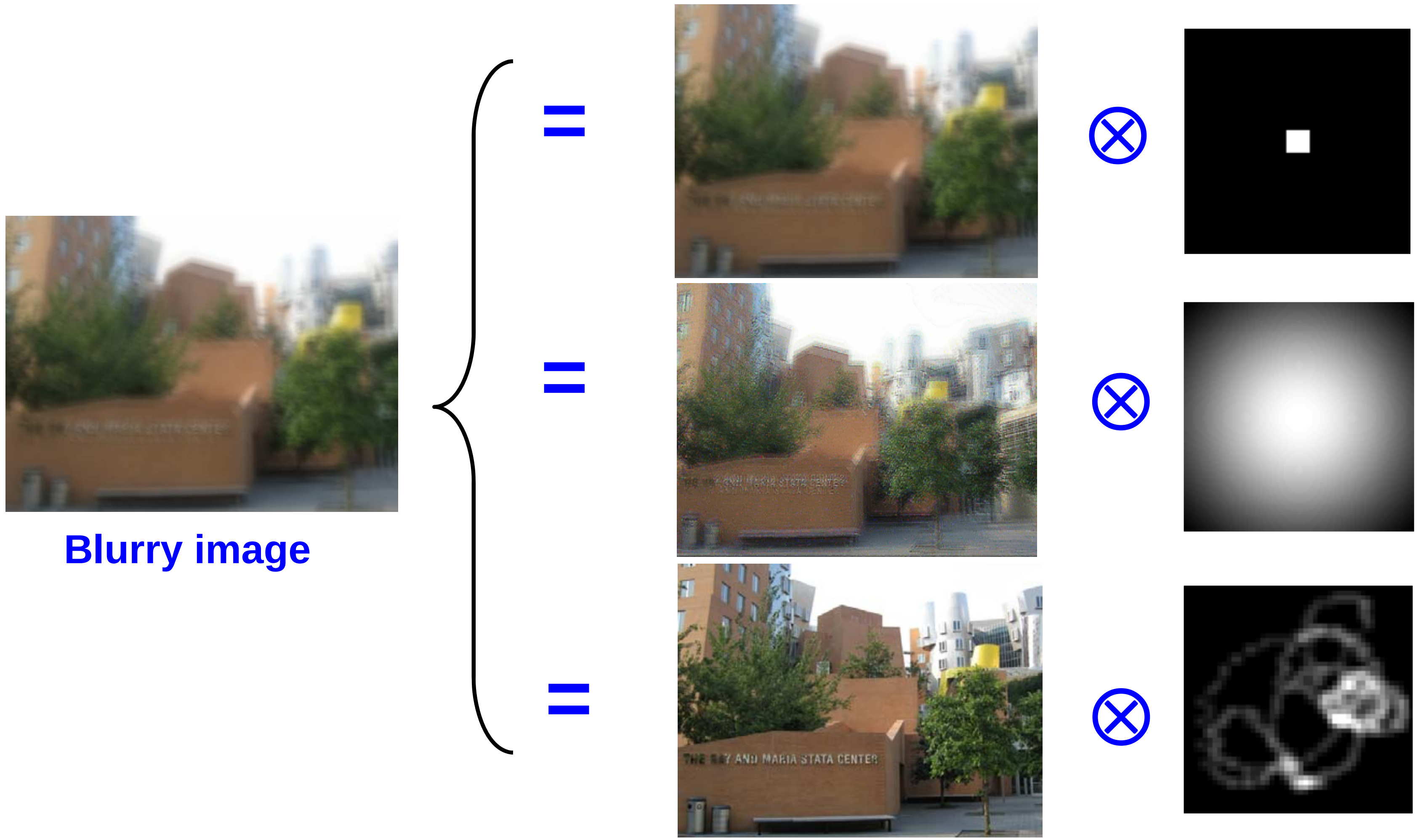
Desired output

Convolution operator

Blur model is an approximation!

Ill-posed Problem

- Not one-to-one mapping
  - Possible to have multiple solutions



Related Terminology

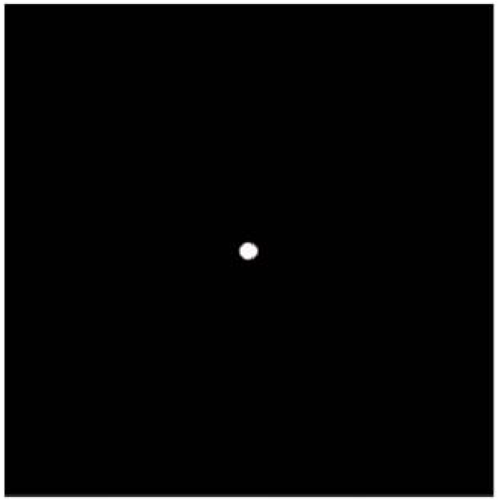
- Image Deblurring
- Image Deconvolution
- Methodology
  - Inverse filtering
  - Minimum mean square error: Wiener filtering
  - Constrained Least Squares (Regularization)
    - Regularized (high pass operator)
    - Bayesian (Prior model)

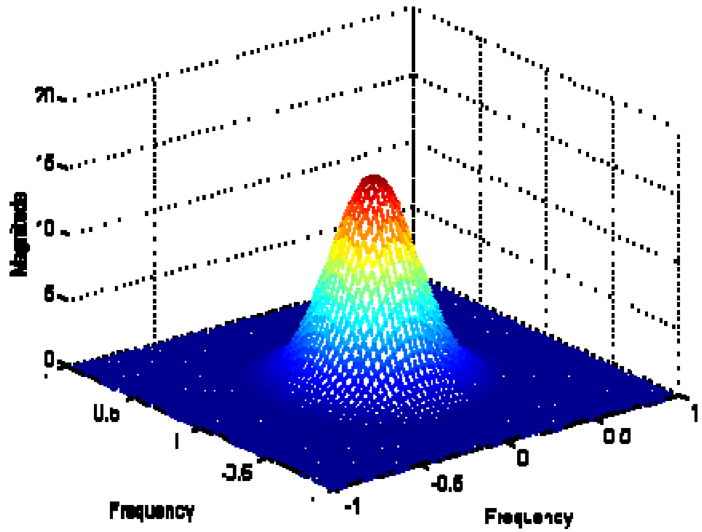
Degradation Functions (1)

- Blurring + noise
- Point spread function (blurring)
  - Focusing error: Gaussian blur
  - Motion blur
  - Atmosphere turbulence
    - spatially variant index of refraction
- film grain noise
  - silver grains are randomly distributed over the film

Degradation Functions (2)

- Point spread Function
  - Blurring models
  - NxN uniform/Gaussian matrix
- PSF를 구하는 것이 가장 중요함.

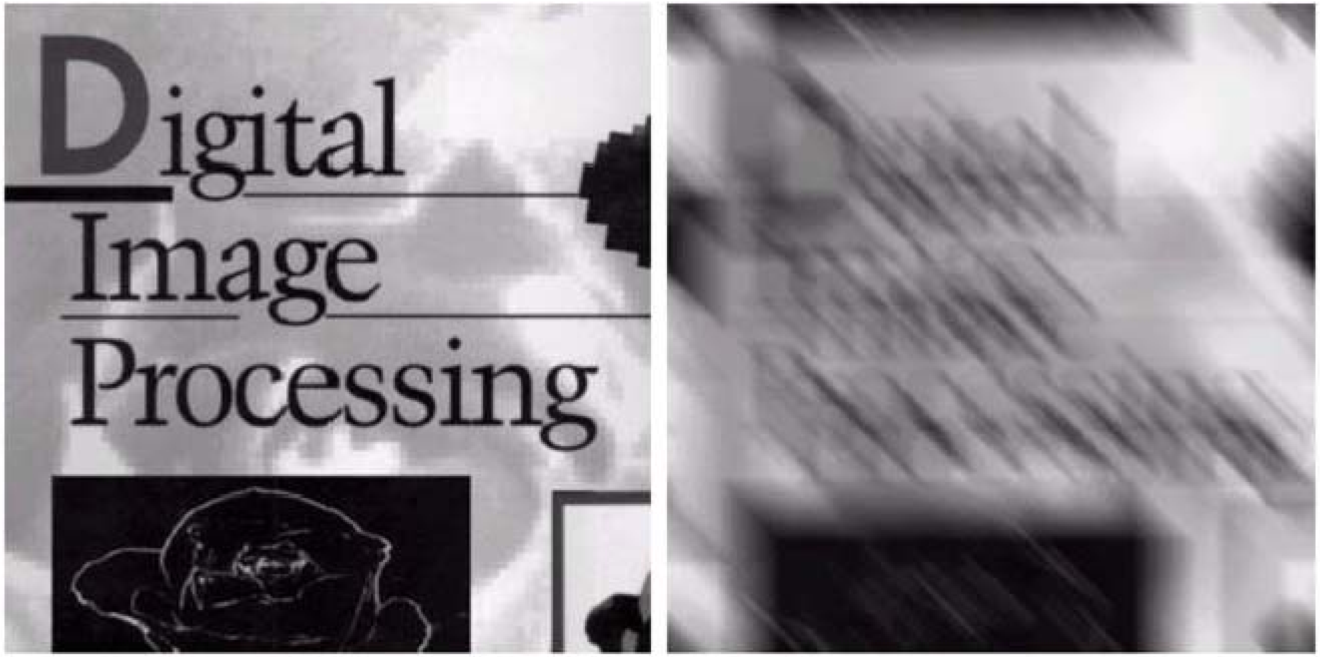




$$G(x,y)=\frac{1}{2\pi\sigma_x\sigma_y}\exp\left\{-\left(\frac{x^2}{2\sigma_x^2}+\frac{y^2}{2\sigma_y^2}\right)\right\}$$

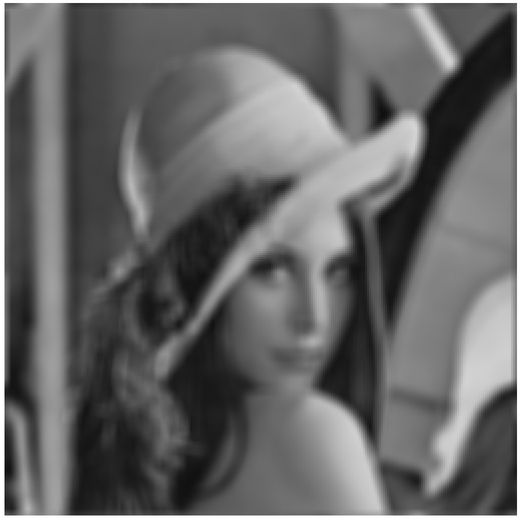
Degradation Functions (3)

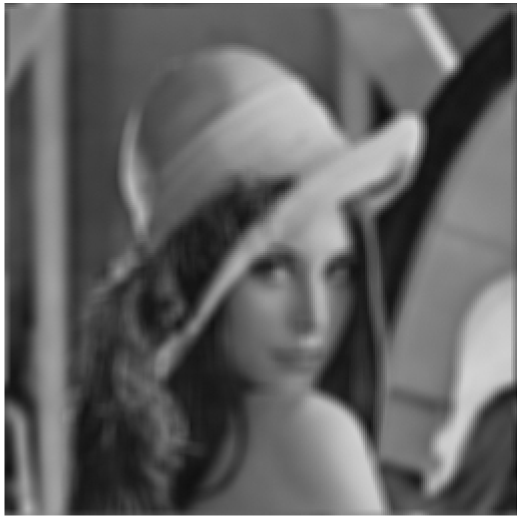
- Motion Blurring
  - 1-directional blurring parallel to object/camera motion



Degradation Functions (4)


original image


7x7 blurred image


40dB Noise image


20dB Noise image


10dB Noise image

Degradation Functions (5)

Atmospheric turbulance: $$H(u,v) = e^{-k\left(u^2+v^2\right)}$$

k: a constant that depends on the nature of the turbulence

a b
c d

FIGURE 5.25 Illustration of the atmospheric turbulence model. (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. (Original image courtesy of NASA.)

Inverse filtering (1)

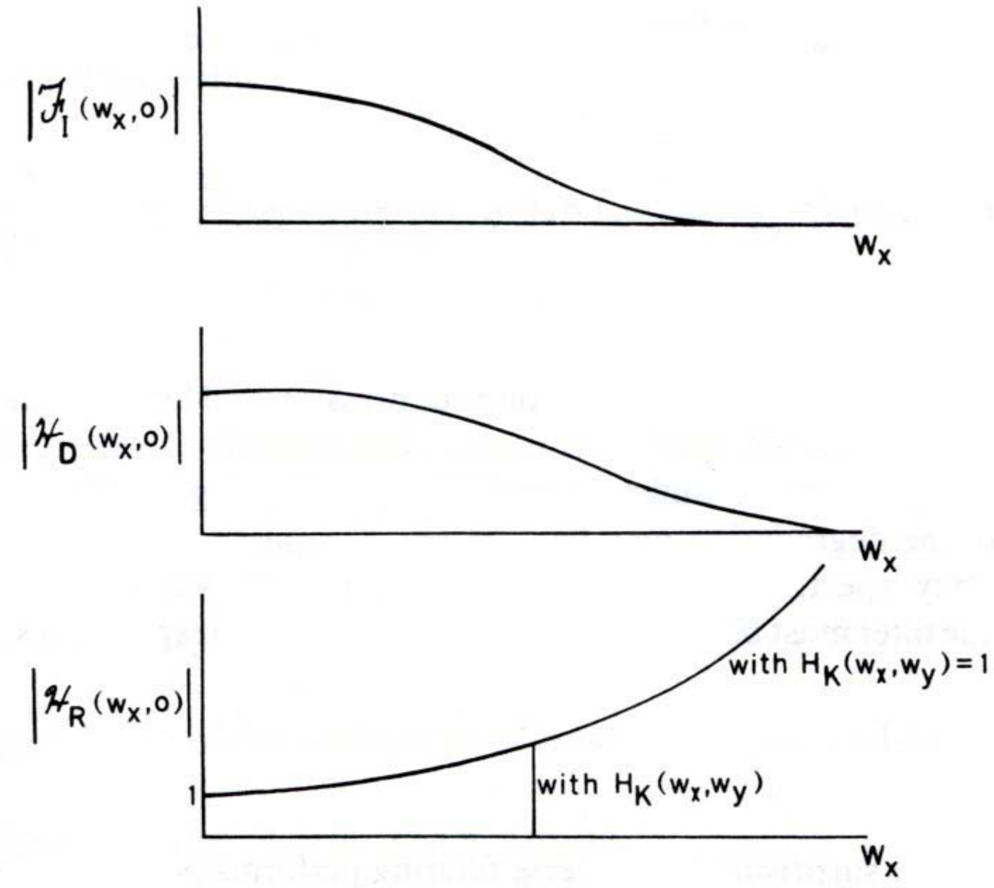
- Reciprocal filter in the frequency domain

$$H_R(u,v)=\frac{1}{H_D(u,v)}$$

$$\hat{F}_I(u,v)=F_I(u,v)+\frac{N(u,v)}{H_D(u,v)}$$

$$H_R(u,v)=\frac{H_k(u,v)}{H_D(u,v)}$$

$$|H_k(u,v)|=\begin{cases}1 & \forall |u,v|\le \omega_c\\ 0 & \text{otherwise}\end{cases}$$



Inverse filtering (2)

Inverse Filter- Pseudo_Inverse Filtering

(a) (b) (c)

(a) Original image of 512x512 pixels.
(b) Image blurred by a Gaussian-shaped point spread function.
(c) Result of inverse filtering.

Wiener filtering (1)

- Minimum mean squared error

$$\mathbf{g} = \mathbf{Hf} + \mathbf{n}$$

$$e = \mathbf{E}[(\mathbf{f} - \tilde{\mathbf{f}})^{\mathbf{t}}(\mathbf{f} - \tilde{\mathbf{f}})] \text{ is minimized}$$

by orthogonality theorem $E\left[\left(\mathbf{f} - \hat{\mathbf{f}}\right)\mathbf{g}^t\right] = 0$

$$\mathbf{G} = \mathbf{R}_f\mathbf{H}^t\left[\mathbf{H}\mathbf{R}_f\mathbf{H}^t + \mathbf{R}_n\right]^{-1} :$$

where $\mathbf{R}_f = E\left[\mathbf{f}\mathbf{f}^t\right]$, $\mathbf{R}_n = E\left[\mathbf{n}\mathbf{n}^t\right]$

Wiener filtering (2)

$$S_{F_I F_O}(u,v) = \mathrm{H}_D^*(u,v) S_{F_I}(u,v)$$

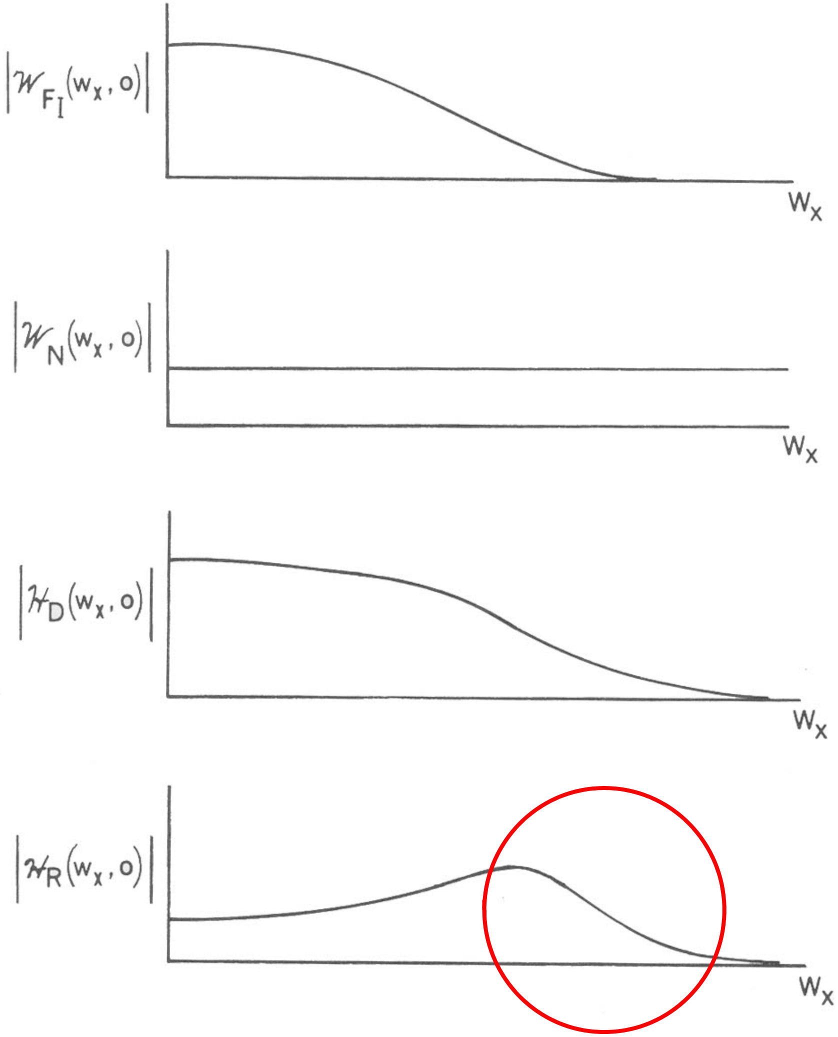
$$S_{F_O F_O}(u,v) = \left|\mathrm{H}_D^*(u,v)\right|^2 S_{F_I}(u,v) + S_N(u,v)$$

$$\therefore \mathrm{H}_R(u,v) = \frac{H_D^*(u,v)}{\left|H_D(u,v)\right|^2 + S_N(u,v)/S_I(u,v)} : \quad \text{continuous Wiener - filter}$$

⇨ 1. This filter is non - causal

2. a priori statistical knowledge of the input and noise should be known

Wiener filtering (3)



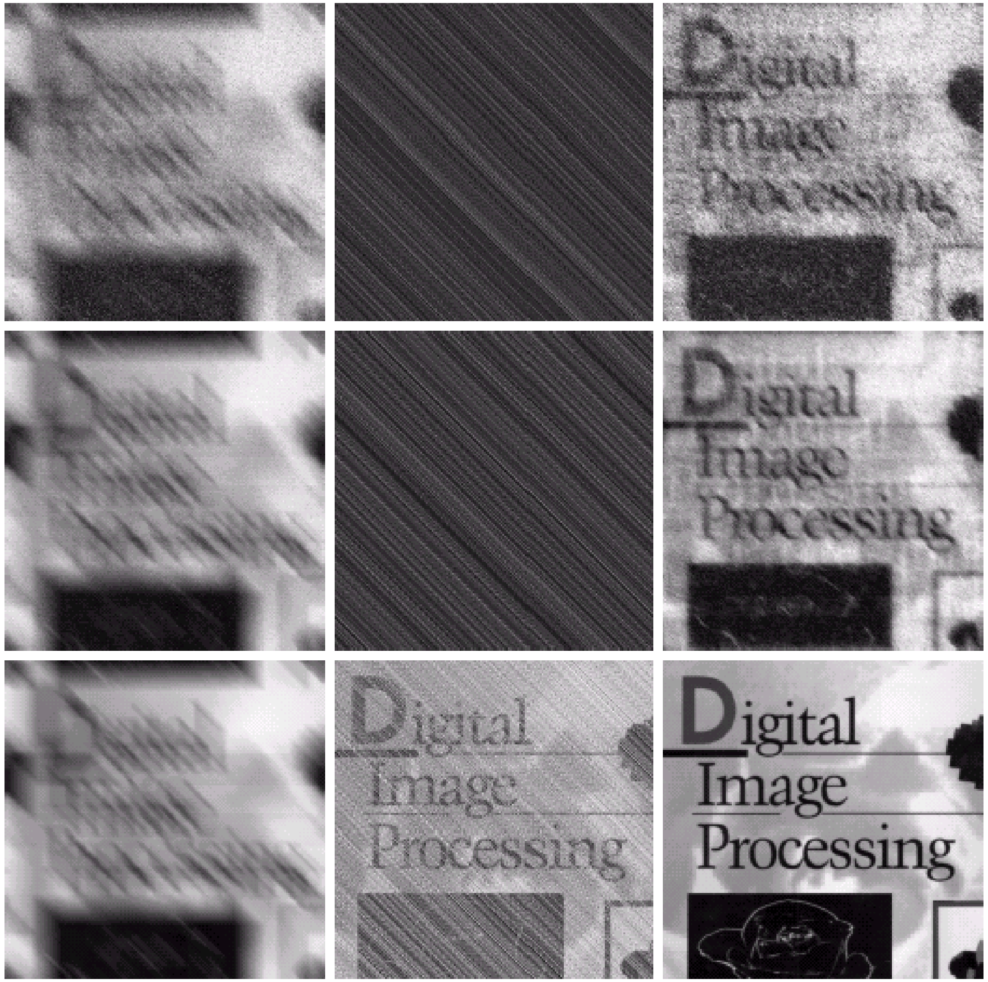
Typical spectra of Wiener filtering image restoration system: high freq-band artifacts

Wiener filtering (4)

(*a*) Observation (*b*) Restoration

FIGURE 12.5-3. Wiener image restoration.

Wiener filtering (5)



a b c
d e f
g h i

FIGURE 5.29 (a) Image corrupted by motion blur and additive noise. (b) Result of inverse filtering. (c) Result of Wiener filtering. (d)–(f) Same sequence, but with noise variance one order of magnitude less. (g)–(i) Same sequence, but noise variance reduced by five orders of magnitude from (a). Note in (h) how the deblurred image is quite visible through a "curtain" of noise.

Least Squares Method

- Observation: $\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}$
- Solution: $\tilde{\mathbf{f}}$
- least square solution

$$(\mathbf{g} - \mathbf{H}\tilde{\mathbf{f}})^{\mathbf{t}}(\mathbf{g} - \mathbf{H}\tilde{\mathbf{f}}) = \mathbf{n}^{\mathbf{t}}\mathbf{n} \quad \text{is minimized}$$

- MMSE solution (Wiener filter)

$$e = \mathbf{E}[(\mathbf{f} - \tilde{\mathbf{f}})^{\mathbf{t}}(\mathbf{f} - \tilde{\mathbf{f}})] \quad \text{is minimized}$$

Pseudo-inverse Restoration (1)

$$\mathbf{W}(\tilde{\mathbf{f}}) = \| \mathbf{g} - \mathbf{H}\tilde{\mathbf{f}} \|^2 = (\mathbf{g} - \mathbf{H}\tilde{\mathbf{f}})^T (\mathbf{g} - \mathbf{H}\tilde{\mathbf{f}})$$

$$\frac{\mathbf{W}(\tilde{\mathbf{f}})}{\partial \tilde{\mathbf{f}}} = -2\mathbf{H}^T(\mathbf{g} - \mathbf{H}\tilde{\mathbf{f}}) = 0 \quad \Rightarrow \quad \mathbf{H}^T\mathbf{g} = \mathbf{H}^T\mathbf{H}\tilde{\mathbf{f}}$$

$$\therefore \tilde{\mathbf{f}} = \mathbf{H}^{-}\mathbf{g} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{g} : \ \text{pseudo - inverse}$$

$$\text{Consider} \quad \underset{(P\times 1)}{\mathbf{p}} = \underset{(P\times Q)}{\mathbf{T}}\underset{(Q\times 1)}{\mathbf{f}}$$

1) If $\mathbf{T}$ is a square matrix $\Rightarrow \mathbf{f} = \mathbf{T}^{-1}\mathbf{g}$

2) $P > Q$ (over - determined case) and $\mathbf{T}$ is of rank P

$$\mathbf{T}^{-} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T \quad \text{and} \quad \mathbf{T}^{-}\mathbf{T} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{T} = \mathbf{I}$$

3) $P < Q$ (under - determined case); restoration model

$$\mathbf{T}^{-} = \mathbf{T}^T(\mathbf{T}^T\mathbf{T})^{-1} \quad \text{but} \quad \mathbf{T}^{-}\mathbf{T} \neq \mathbf{I}$$

Pseudo-inverse Restoration (2)

Ringing artifact

FIGURE 12.3-6. Pseudoinverse image restoration for moderate and high degrees of horizontal blur.

Constrained Least Squares Method (1)

- Minimize $(\mathbf{g}-\mathbf{H}\tilde{\mathbf{f}})^t(\mathbf{g}-\mathbf{H}\tilde{\mathbf{f}})$

  subject to $\mathbf{E}[\tilde{\mathbf{f}}^T\tilde{\mathbf{f}}]=e$: finite energy constraint

  or subject to $\mathbf{E}[\tilde{\mathbf{f}}^T\tilde{\mathbf{f}}]\le \mathrm{e}$ and $\tilde{\mathbf{f}}\ge 0$

  $\Rightarrow$ constrained restoration

  $\Rightarrow$ Lagrangian multiplier

Constrained Least Squares Method (2)

$$\mathbf{g} = \mathbf{Hf} + \mathbf{n} \quad \Rightarrow \quad \text{minimize } e = \| \mathbf{g} - \mathbf{H}\tilde{\mathbf{f}} \|^2$$

1. energy conservation constraint, given by

$$\tilde{\mathbf{f}}^T \tilde{\mathbf{f}} = \text{const.} = c$$

$$\min J(\tilde{\mathbf{f}}, \lambda) = \| \mathbf{g} - \mathbf{H}\tilde{\mathbf{f}} \|^2 + \lambda(\| \tilde{\mathbf{f}} \|^2 - c)$$

$$\frac{\partial J(\tilde{\mathbf{f}},\lambda)}{\partial \tilde{\mathbf{f}}} = \frac{\partial}{\partial \tilde{\mathbf{f}}}\left[(\mathbf{g} - \mathbf{H}\tilde{\mathbf{f}})^T (\mathbf{g} - \mathbf{H}\tilde{\mathbf{f}}) + \lambda(\tilde{\mathbf{f}}^T \tilde{\mathbf{f}} - c)\right]$$

$$= \frac{\partial}{\partial \tilde{\mathbf{f}}}\left[\mathbf{g}^T \mathbf{g} - \tilde{\mathbf{f}}^T \mathbf{H}^T \mathbf{g} - \mathbf{g}^T \mathbf{H}\tilde{\mathbf{f}} + \tilde{\mathbf{f}}^T \mathbf{H}^T \mathbf{H}\tilde{\mathbf{f}} + \lambda(\tilde{\mathbf{f}}^T \tilde{\mathbf{f}} - c)\right]$$

$$= -2\mathbf{H}^T \mathbf{g} + 2\mathbf{H}^T \mathbf{H}\tilde{\mathbf{f}} + 2\lambda\tilde{\mathbf{f}} = 0$$

$$\Rightarrow \tilde{\mathbf{f}} = (\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I})^{-1} \mathbf{H}^T \mathbf{g}$$

Constrained Least Squares Method (3)

Regularized constrained least squares method

$$\text{minimize} \| \mathbf{Q}\tilde{\mathbf{f}} \|^2, \text{subject to} \| \mathbf{g} - \mathbf{H}\tilde{\mathbf{f}} \|^2 = \| \mathbf{n} \|^2$$

$$J(\tilde{\mathbf{f}}, \lambda) = \| \mathbf{Q}\tilde{\mathbf{f}} \|^2 + \lambda \left(\| \mathbf{g} - \mathbf{H}\tilde{\mathbf{f}} \|^2 - \| \mathbf{n} \|^2 \right)$$

$$\tilde{\mathbf{f}} = \left[\mathbf{H}^T\mathbf{H} + \frac{1}{\lambda}\mathbf{Q}^T\mathbf{Q} \right]^{-1} \mathbf{H}^T\mathbf{g} : \text{ constrained least square filter}$$

1) $Q = I$

$$\tilde{\mathbf{f}} = \left[\mathbf{H}^T\mathbf{H} + \frac{1}{\lambda} I \right]^{-1} \mathbf{H}^T\mathbf{g} \text{ ; energy conservation constrained filter}$$

Constrained Least Squares Method (4)

- Result

Degraded (15x15 blur + additive noise)

Restored (Iterative least squares)

Comparison (1)

Levin, et al. “Image and Depth from a Conventional Camera with a Coded Aperture,” SIGGRAPH 2007.

Uniform blurring (17x17)

Restored result

Comparison (2)

Levin, et al. "Image and Depth from a Conventional Camera with a Coded Aperture," SIGGRAPH 2007.

Uniform blurring (37x37)

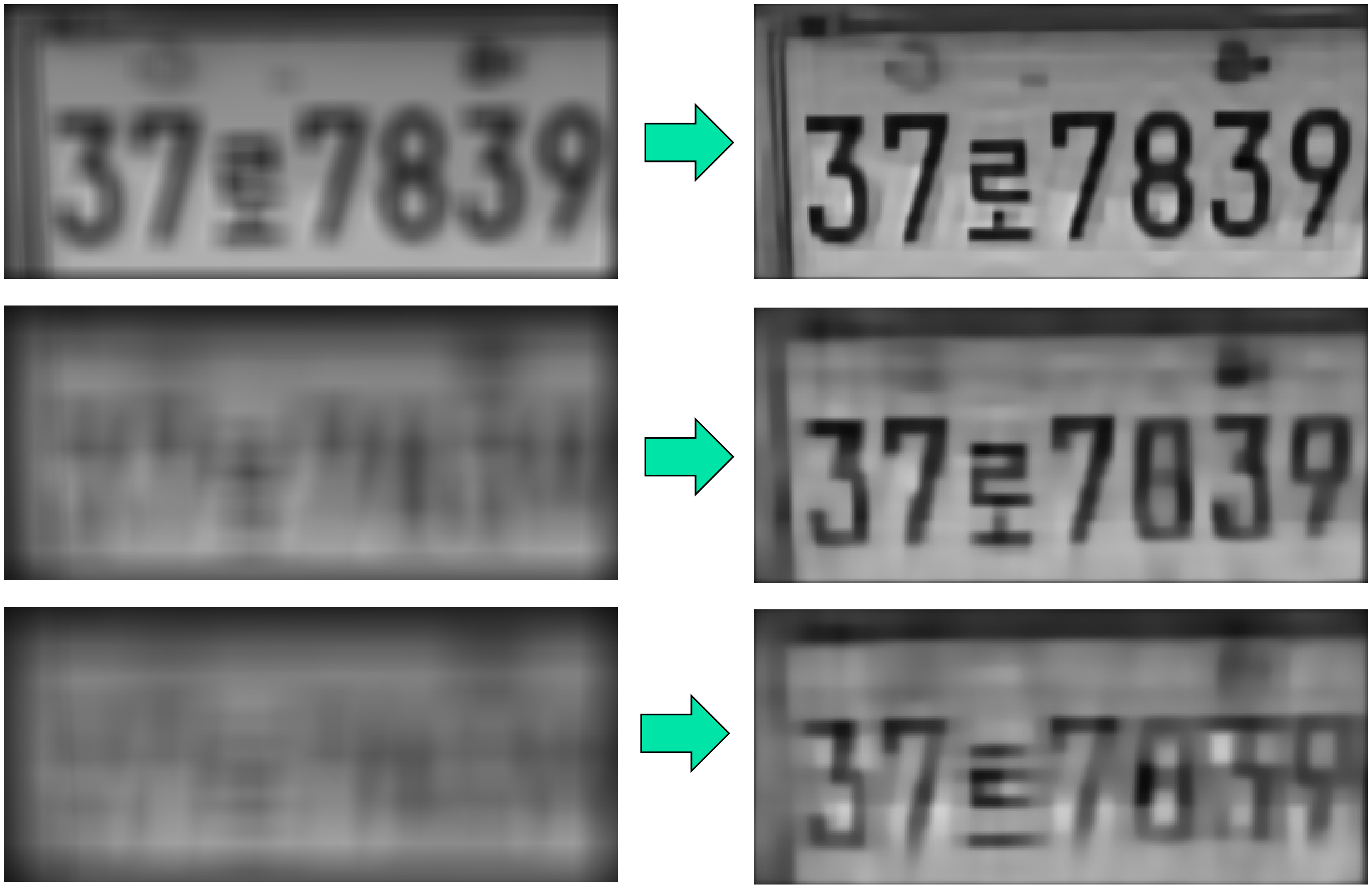
Restored result

Comparison (3)

블러 크기 17

37

47

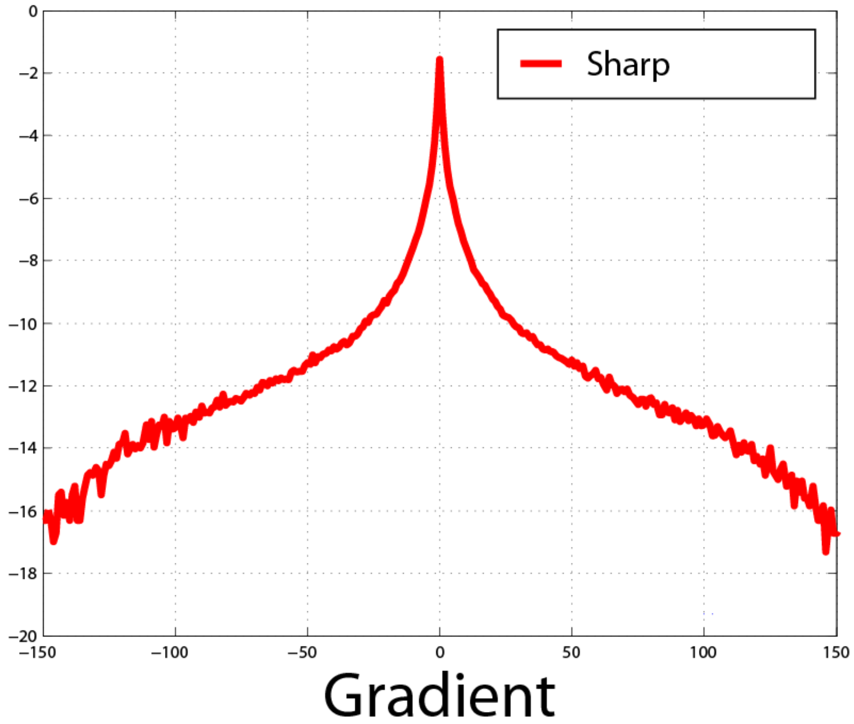


블러 영상 디블러링 복원 결과

Bayesian Deblurring

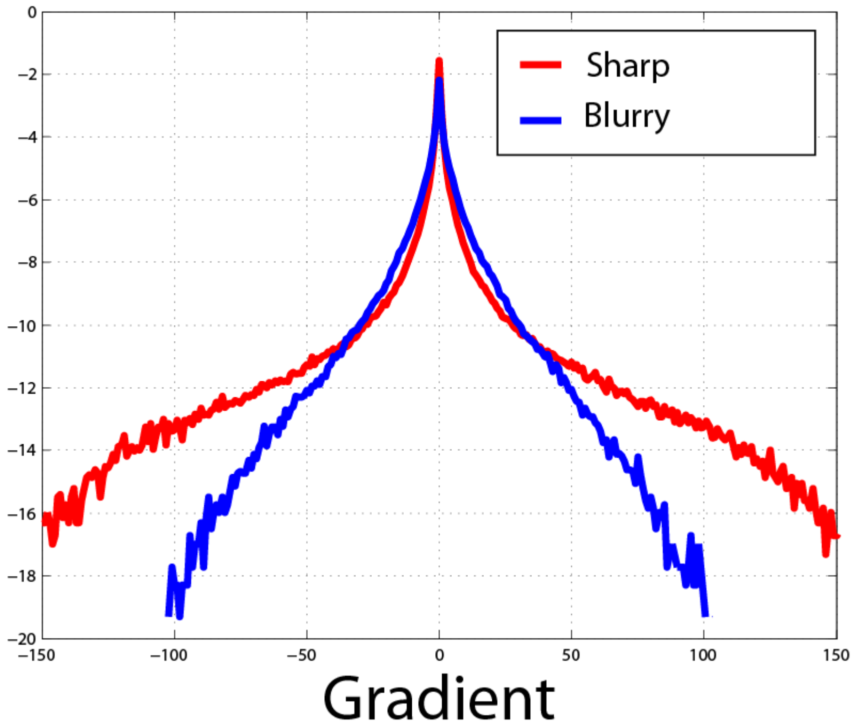
Natural image statistics (prior model)

Histogram of image gradients



Blury images have different statistics

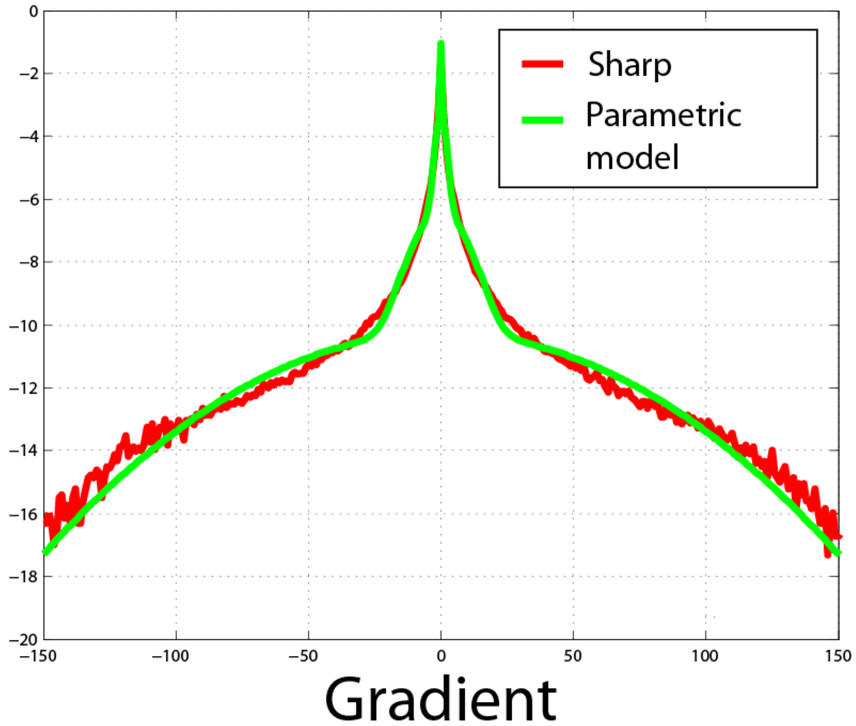
Histogram of image gradients



Parametric model of sharp image

- Parametric distribution of gradients for sharp images

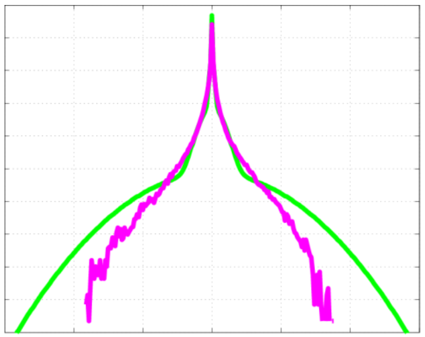
Histogram of image gradients



Three sources of information

1. Reconstruction constraint:

2. Image prior:



Distribution of gradients

3. Blur prior:

Positive
&
Sparse

Three sources of information

y = observed image   b = blur kernel   x = sharp image

$$p(b, x|y) = k \; p(y|b, x) \; p(x) \; p(b)$$

Posterior | 1. Likelihood (Reconstruction constraint) | 2. Image prior | 3. Blur prior

1. Likelihood $p(y|b,x)$

y = observed image b = blur x = sharp image

Reconstruction constraint:

$$p(y|b,x) = \prod_i N(y_i | x_i \otimes b, \sigma^2)$$

$$\propto \prod_i e^{-\frac{(x_i \otimes b - y_i)^2}{2\sigma^2}}$$

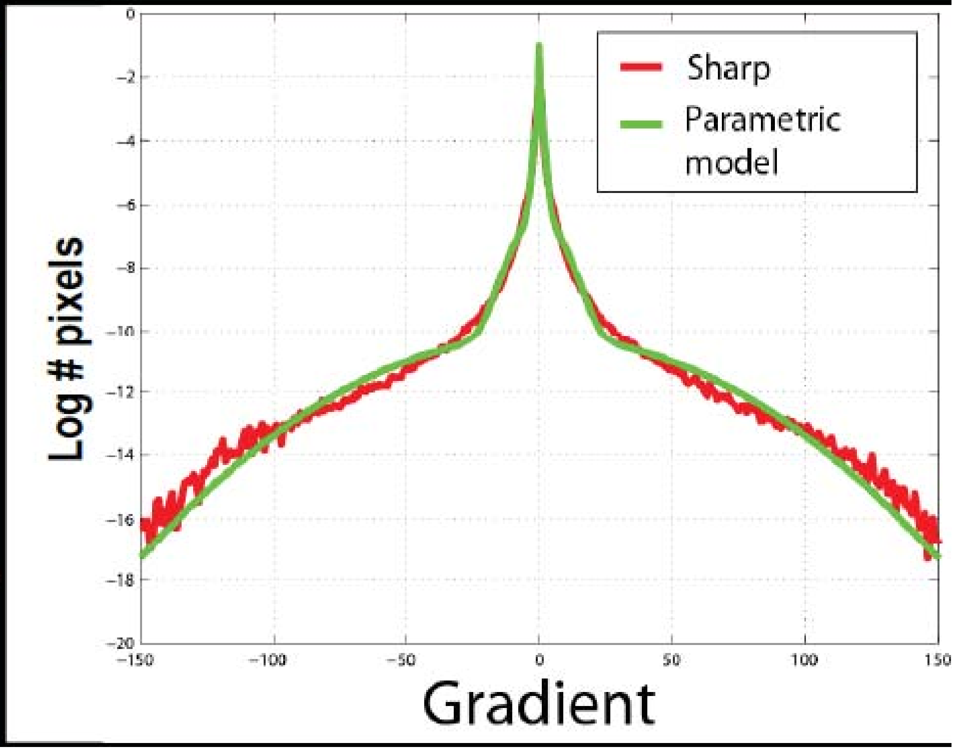
i - pixel index

2. Image prior $p(x)$

y = observed image b = blur x = sharp image

$$p(x) = \prod_i \sum_{c=1}^{C} \pi_c \mathcal{N}(f(x_i)|0, s_c^2)$$

Mixture of Gaussians fit to empirical distribution of image gradients



i - pixel index

c - mixture component index

f - derivative filter

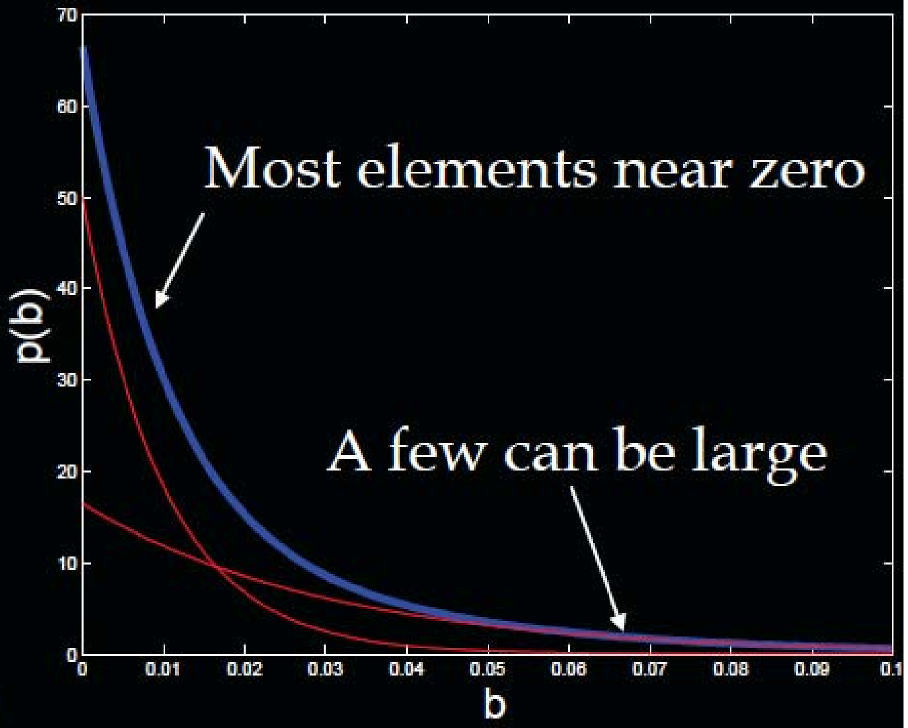
3. Blur prior $p(b)$

y = observed image   b = blur   x = sharp image

$$p(b) = \prod_j \sum_{d=1}^{D} \pi_d \, \mathcal{E}(b_j | \lambda_d)$$

Mixture of Exponentials

- Positive & sparse
- No connectivity constraint



j - blur kernel element

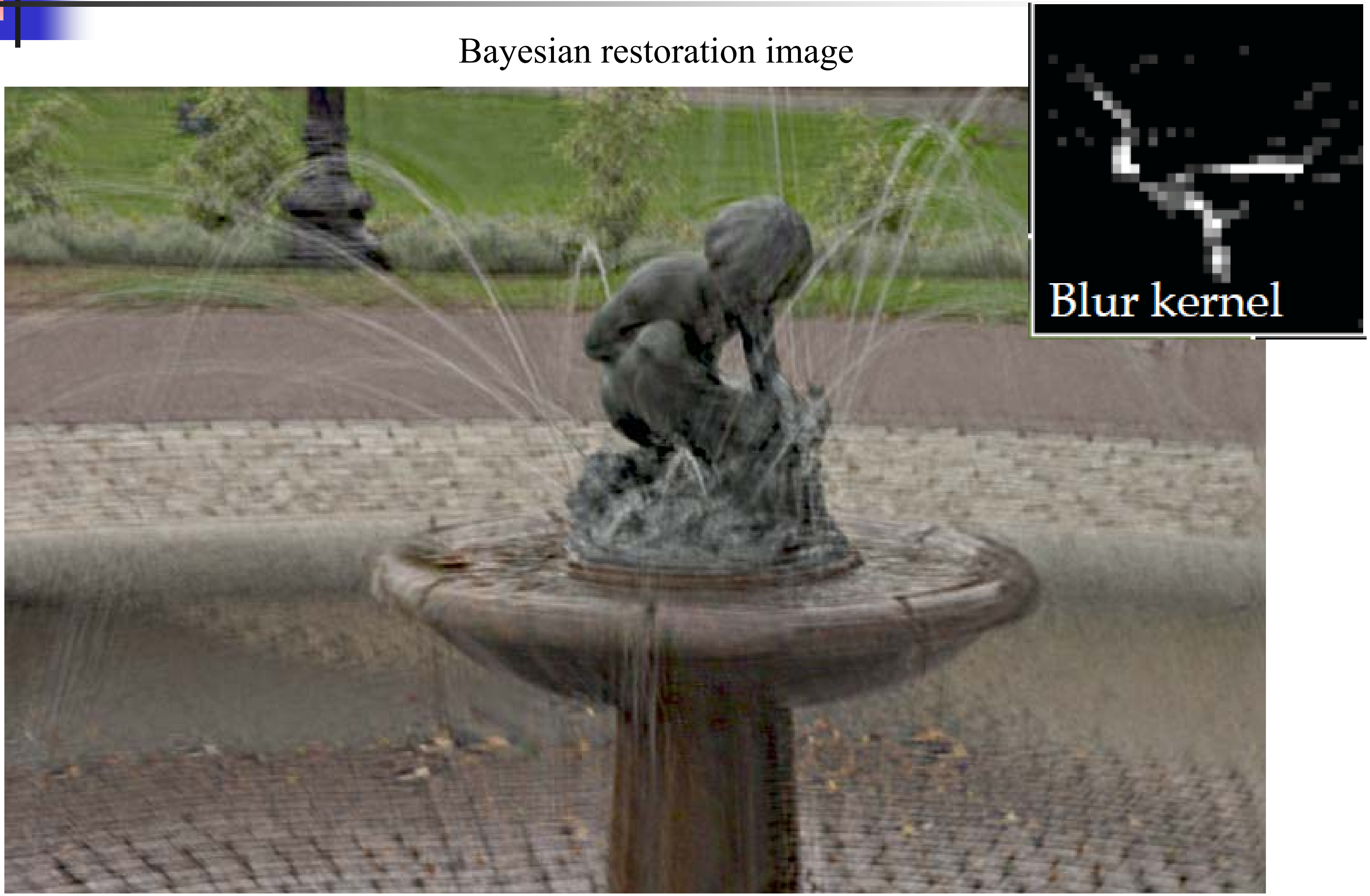
d - mixture component index

Results of Bayesian Method (1)

Original blurred image

Results of Bayesian Method (2)

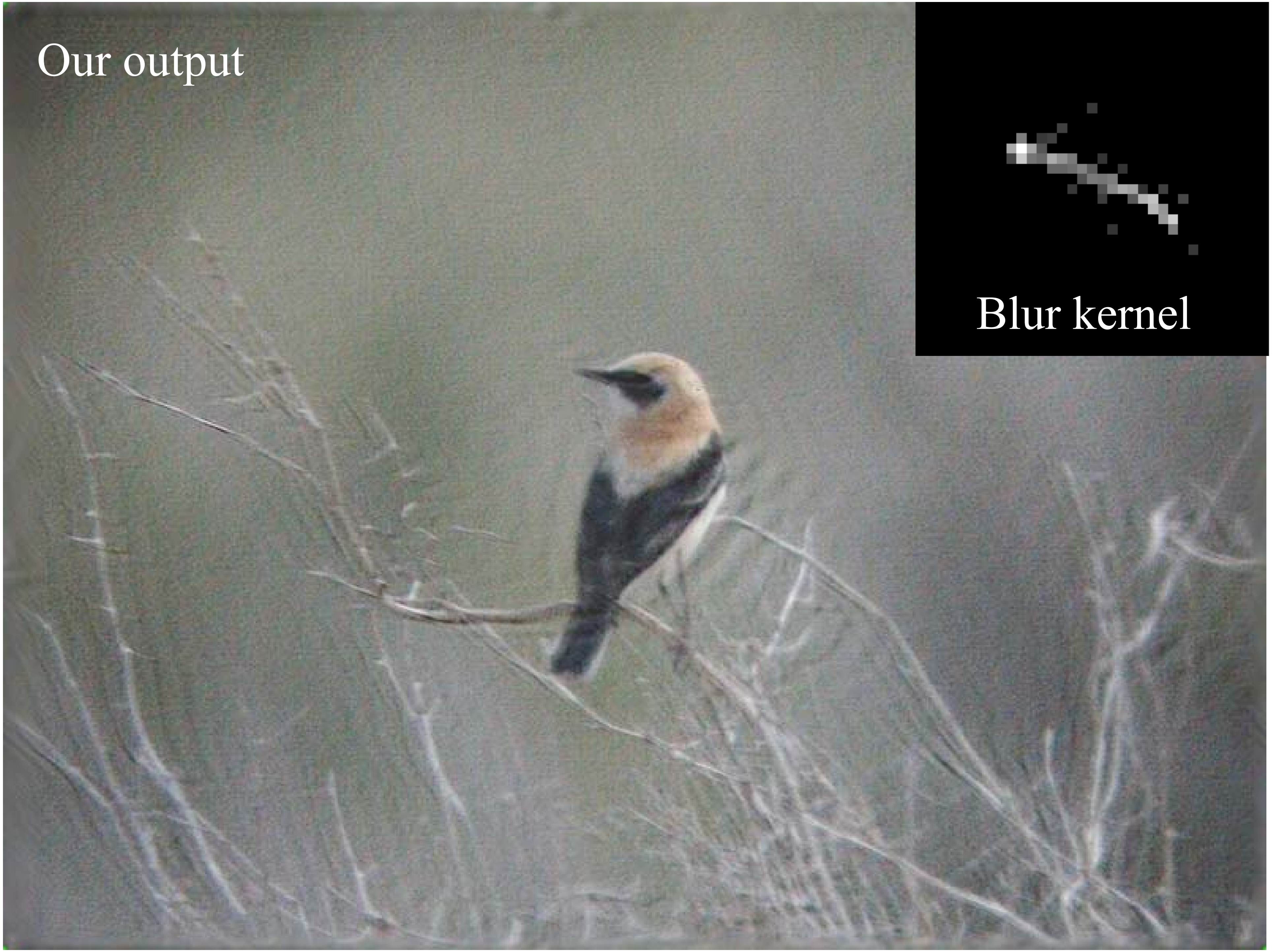
Bayesian restoration image



Results of Bayesian Method (3)

Matlab restoration (deconvolution) image

Original photograph


Our output
Blur kernel

Results of Bayesian Method (4)

Original

Unsharp mask

Bayesian