Medical Image Analysis and Processing

Image Noise Filtering – Point Esimation

Emad Fatemizadeh

Distance/online Course: Session 08

Date: 09 March 2021, 19th Esfand 1399

Main Category

- Point Estimation Methods
- Transform Domain Methods
- Total Variation Approach
- Diffusion Anisotropic Filtering
- Machine Learning

Point Estimation Methods

- Noisy Image Model:

$$g(X) = f(X) + n(X)$$

- g: Noisy Observation
- f: Clean Image
- n: additive noise:
  - Zero mean and known variance (σ^2)
  - $i.i.d$ (independent identical distribution)
  - Uncorrelated
- X: Pixel Position, $X = (x_i, yi)$

Point Estimation Methods

- Problem Formulation:

$$g(X) = f(X) + n(X)$$

$$\hat{f}(X) = \underset{f(X)}{\operatorname{argmin}} \left\{ \int_{\Omega} \big(g(Y) - f(X)\big)^2 K\big(X, Y; g(X), g(Y)\big) dY \right\}$$

- Ω: Image Domain ($\mathbb{R}^2$)
- K: Symmetric, Positive, and Monomodal Kernel
- X, Y: Pixel Position ($\mathbb{R}^2$),

Point Estimation Methods

› Problem Formulation (Continuous Case):

$$\frac{\partial}{\partial f}\left\{\int_{\Omega}\big(g(Y)-f(X)\big)^2 K\big(X,Y;g(X),g(Y)\big)dY\right\}=0$$

$$\Rightarrow -2\int_{\Omega}\big(g(Y)-f(X)\big)K\big(X,Y;g(X),g(Y)\big)dY=0$$

$$\therefore\ \hat{f}(X)=\frac{\int_{\Omega} g(Y)K\big(X,Y;g(X),g(Y)\big)dY}{\int_{\Omega} K\big(X,Y;g(X),g(Y)\big)dY}$$

Point Estimation Methods

› Problem Formulation (Discrete Case):

$$\hat{f}(X) = \frac{\sum_{Y \in \Omega} g(Y) K\big(X, Y; g(X), g(Y)\big)}{\sum_{Y \in \Omega} K\big(X, Y; g(X), g(Y)\big)}$$

Classical Regression Filtering

› Formulation (Gaussian):

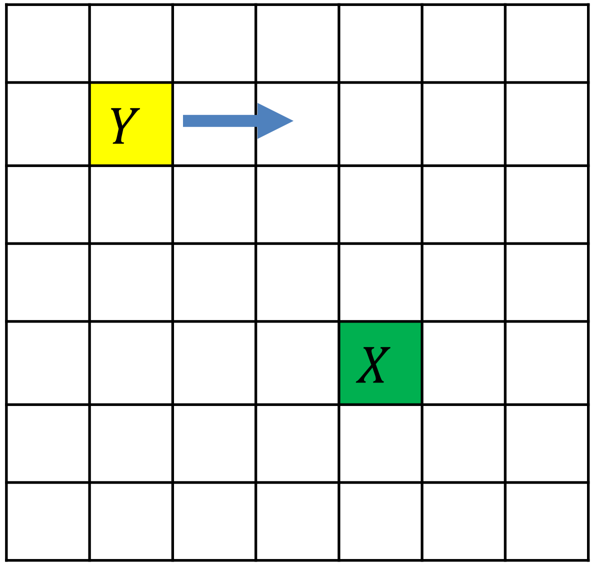
$$K\big(X,Y;g(X),g(Y)\big)=K(X,Y)=\exp\left(-\frac{\|X-Y\|_2^2}{2h_X^2}\right)=G_{h_X}(\|X-Y\|)$$

$$\hat{f}(X)=\frac{\sum_{Y\in\Omega}g(Y)\exp\left(-\frac{\|X-Y\|_2^2}{2h_X^2}\right)}{\sum_{Y\in\Omega}\exp\left(-\frac{\|X-Y\|_2^2}{2h_X^2}\right)}=\frac{\sum_{Y\in\Omega}g(Y)G_{h_X}(\|X-Y\|)}{\sum_{Y\in\Omega}G_{h_X}(\|X-Y\|)}$$

Classical Regression Filtering

› Local Weighted Means (Averaging/Smoothing)!

$$\boxed{\hat{f}(X) = \frac{\sum_{Y\in\Omega} g(Y) G_{h_X}(\|X-Y\|)}{\sum_{Y\in\Omega} G_{h_X}(\|X-Y\|)}}$$



› Linear Filtering + Normalization

Classical Regression Filtering

- Example:

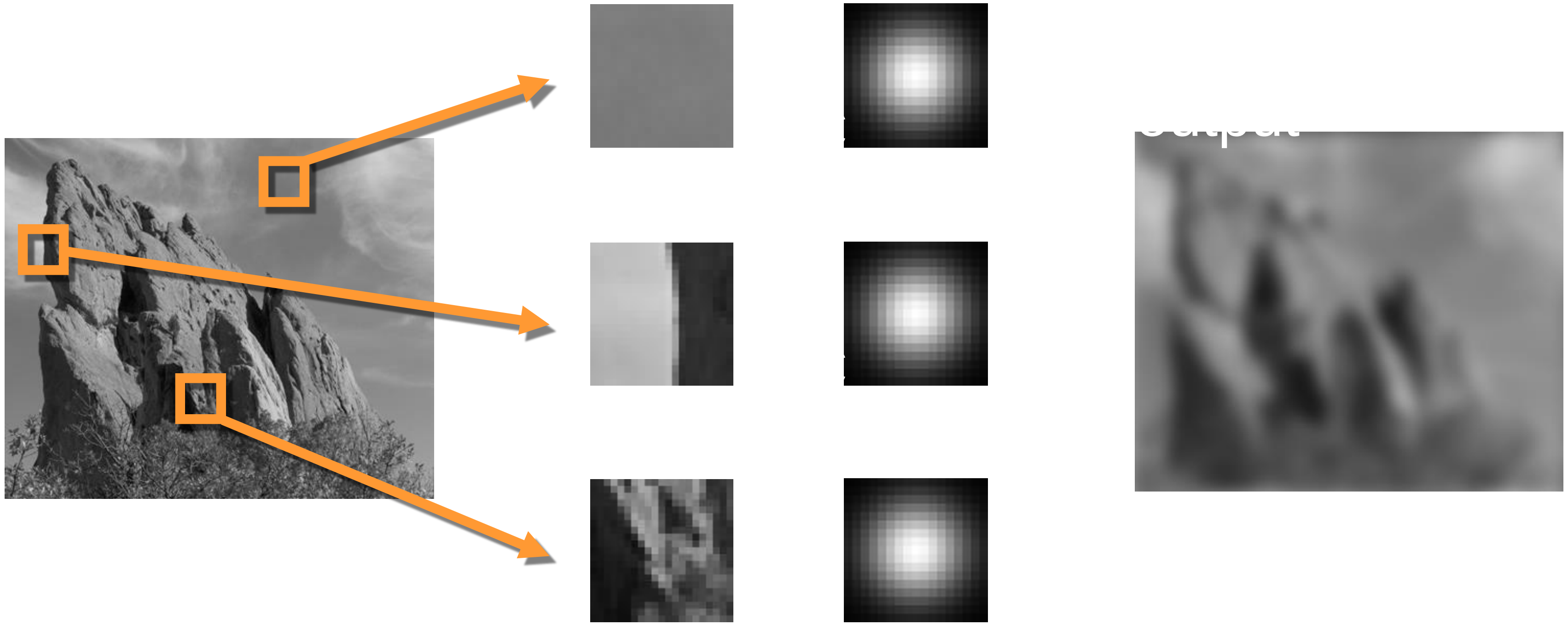
$$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

- Local Mean → Edge Blurring!

Classical Regression Filtering

› Blurring Comes from Blind Local Mean:

› Same Gaussian kernel everywhere.



Bilateral Filtering

- Formulation:
- Idea: Edge Preserving via two spatial and gray level weight function

$$K\big(X, Y; g(X), g(Y)\big) = \exp\left(-\frac{\|X - Y\|_2^2}{2h_X^2}\right) \exp\left(-\frac{\|g(X) - g(Y)\|_2^2}{2h_g^2}\right)$$

$$K\big(X, Y; g(X), g(Y)\big) = G_{h_X}(\|X - Y\|) G_{h_g}(\|g(X) - g(Y)\|)$$

Bilateral Filtering

› Formulation:

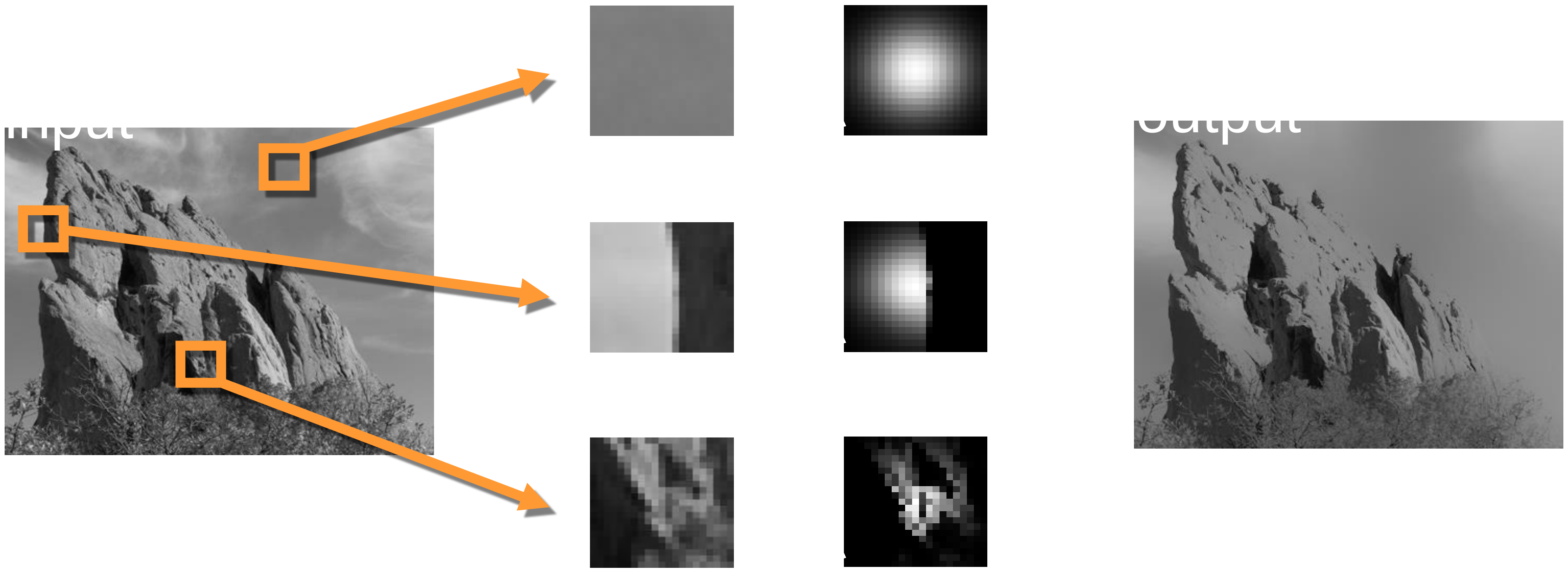
$$\hat{f}(X) = \frac{\sum_{Y \in \Omega} g(Y) G_{h_X}(\|X - Y\|) G_{h_g}(\|g(X) - g(Y)\|)}{\sum_{Y \in \Omega} G_{h_X}(\|X - Y\|) G_{h_g}(\|g(X) - g(Y)\|)}$$

› Nonlinear and high computation cost (why)?

Bilateral Filtering

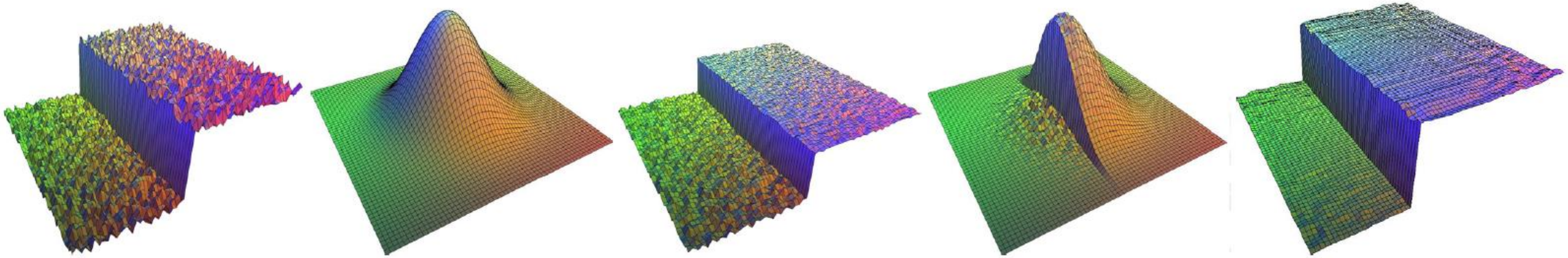
› Gray Level Aware Local Mean:

› Image Content dependent kernel shape



Bilateral Filtering

› Edge Preserving



input

spatial kernel f

influence g in the intensity domain for the central pixel

weight $f \times g$ for the central pixel

output

Bilateral Filtering

› Numerical Example:

$$\begin{bmatrix} 10 & 20 & 25 & 30 \\ 5 & 45 & 35 & 45 \\ 105 & \mathbf{25} & 25 & 43 \\ 35 & 35 & 15 & 45 \end{bmatrix}, \quad G_X = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \quad G_g = 2^{-\frac{|g(X)-g(Y)|}{10}} = \begin{bmatrix} 0.25 & 0.25 & 0.5 \\ 0.004 & 1 & 1 \\ 0.5 & 0.5 & 0.5 \end{bmatrix}$$

$$G_X \odot G_g = \begin{bmatrix} 0.25 & 0.5 & 0.5 \\ 0.008 & 4 & 2 \\ 0.5 & 1 & 0.5 \end{bmatrix} \Rightarrow f_{LM} = \left[\frac{610}{16}\right] = 38, \quad f_{BL} = \left[\frac{252.0703}{9.2578}\right] = 27$$

Bilateral Filtering Parameters

- $K\big(X, Y; g(X), g(Y)\big) = \exp\left(-\frac{\|X-Y\|_2^2}{2h_X^2}\right)\exp\left(-\frac{\|g(X)-g(Y)\|_2^2}{2h_g^2}\right)$
- h_X: Control the smoothness of details:
  - Approximately 0.02 of diagonal image size
- h_g: Control edge preserving capability
  - Proportional to edge amplitude (mean or median of image gradient)
  - Proportional to noise level ($h_g = 1.95\sigma_n$)

Bilateral Filtering - Variation

› Formulation for “Salt & Pepper” noise:

$$\hat{f}(X) = \frac{\sum_{Y\in\Omega} g(Y)G_{h_X}(\|X-Y\|)G_{h_g}(\|g_{Med}(X)-g_{Med}(Y)\|)}{\sum_{Y\in\Omega} G_{h_X}(\|X-Y\|)G_{h_g}(\|g_{Med}(X)-g_{Med}(Y)\|)}$$

› g_{Med}: Median-Filtered version of images!

› Iterative Bilateral Filtering:

$$\hat{f}_{n+1}(X) = BF\left(\hat{f}_n(X)\right)$$

Bilateral Filtering - Variation

- Trilateral Filter
- Symmetric Bilateral Filter
- Cross and Joint Bilateral Filter
- Dual Bilateral Filter
- Several efforts to speedup
- …

Non Local Mean (NLM) Filtering

- Idea:
  - No Local Mean (NLM): $(h_X \to 0)$
  - Average (smooth) with weighs determines by Region Similarity

$$K\big(g(X), g(Y)\big) = \exp\left(-\frac{dist\big(S_g(X), S_g(Y)\big)}{2h_V^2}\right)$$

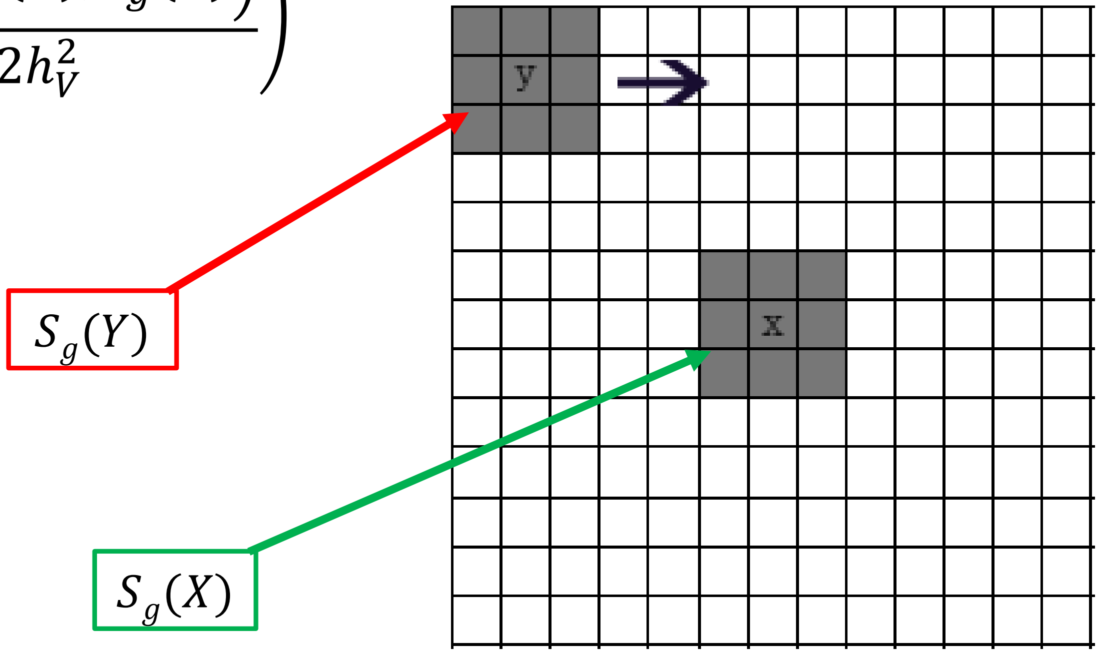
  - $S_g(X)$: Region Properties/Contents/Features/... based on a windows around X.

Non Local Mean (NLM) Filtering

› Implementation:

– One fixed window, $S(X)$, and one float window, $S(Y)$

$$K\big(g(X), g(Y)\big) = \exp\left(-\frac{dist\left(S_g(X), S_g(Y)\right)}{2h_V^2}\right)$$



Non Local Mean (NLM) Filtering

- Naïve Implementation:

$$dist\left(S_g(X), S_g(Y)\right) = \sum_{S_g} G_p \odot \left(S_g(X) - S_g(Y)\right).^2$$

- Sum of weighted (element-wise) square distance.
- G_p: Penalizes pixels far from the center of the neighborhood
- If $X = Y, K(g(X), g(X)) = 1$ (over weight effect, due to normalization):

$$K\left(g(X), g(X)\right) = \max_{X \neq Y}\left\{K\left(g(X), g(Y)\right)\right\}$$

Non Local Mean (NLM) Filtering

- For $G_p = 1$ and a more precis distance estimation (using clean signal, f):

$$dist\left(S_f(X), S_f(Y)\right) = \left\|S_f(X) - S_f(Y)\right\|_2^2$$

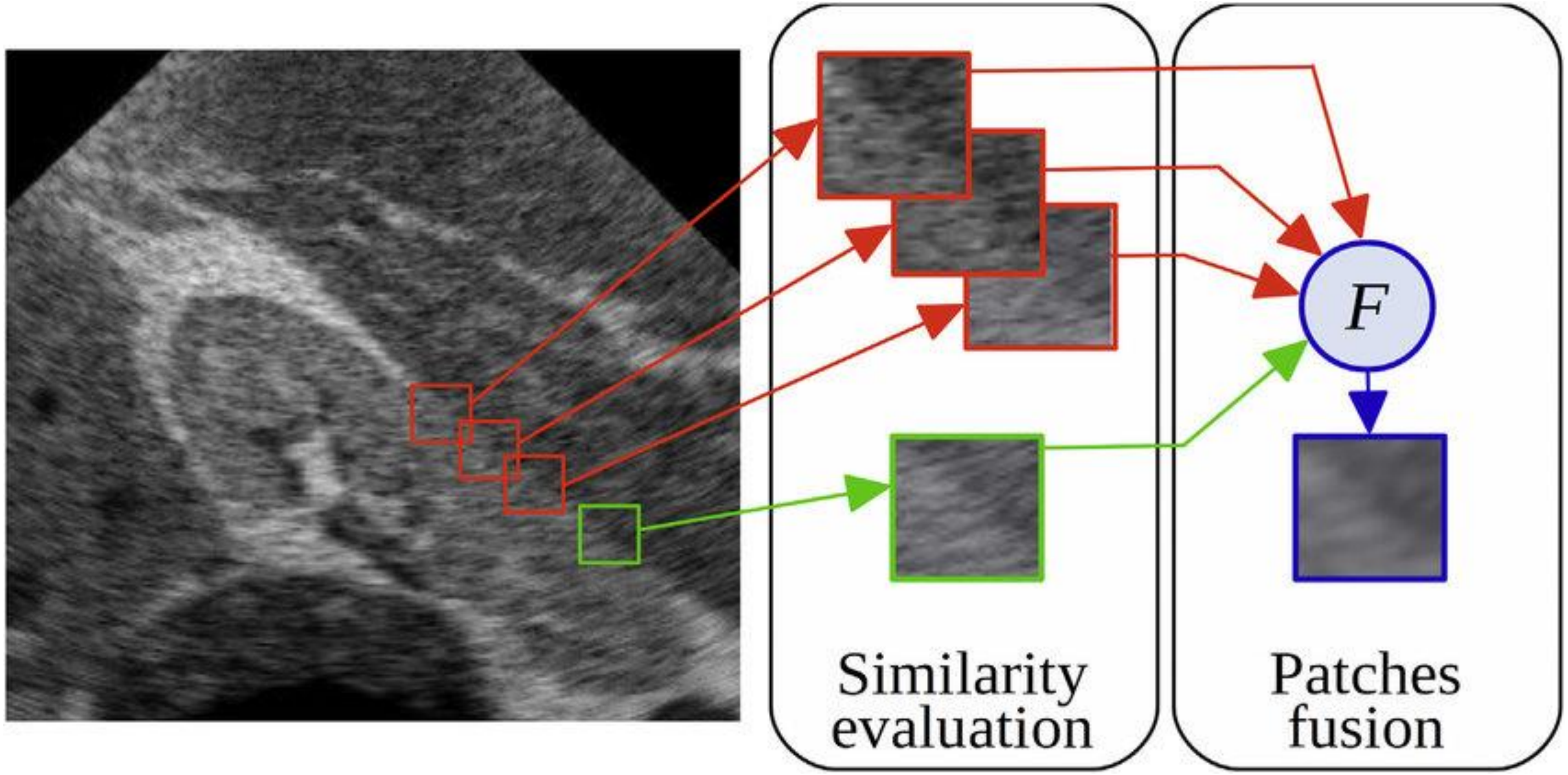
- How to estimate?

$$g = f + n \Rightarrow E\left\{\left\|S_g(X) - S_g(Y)\right\|_2^2\right\} = E\left\{\left\|S_f(X) - S_f(Y)\right\|_2^2\right\} + 2\sigma^2$$

$$dist\left(S_f(X), S_f(Y)\right) = \max\left\{\left\|S_g(X) - S_g(Y)\right\|_2^2 - 2\sigma^2, 0\right\}$$

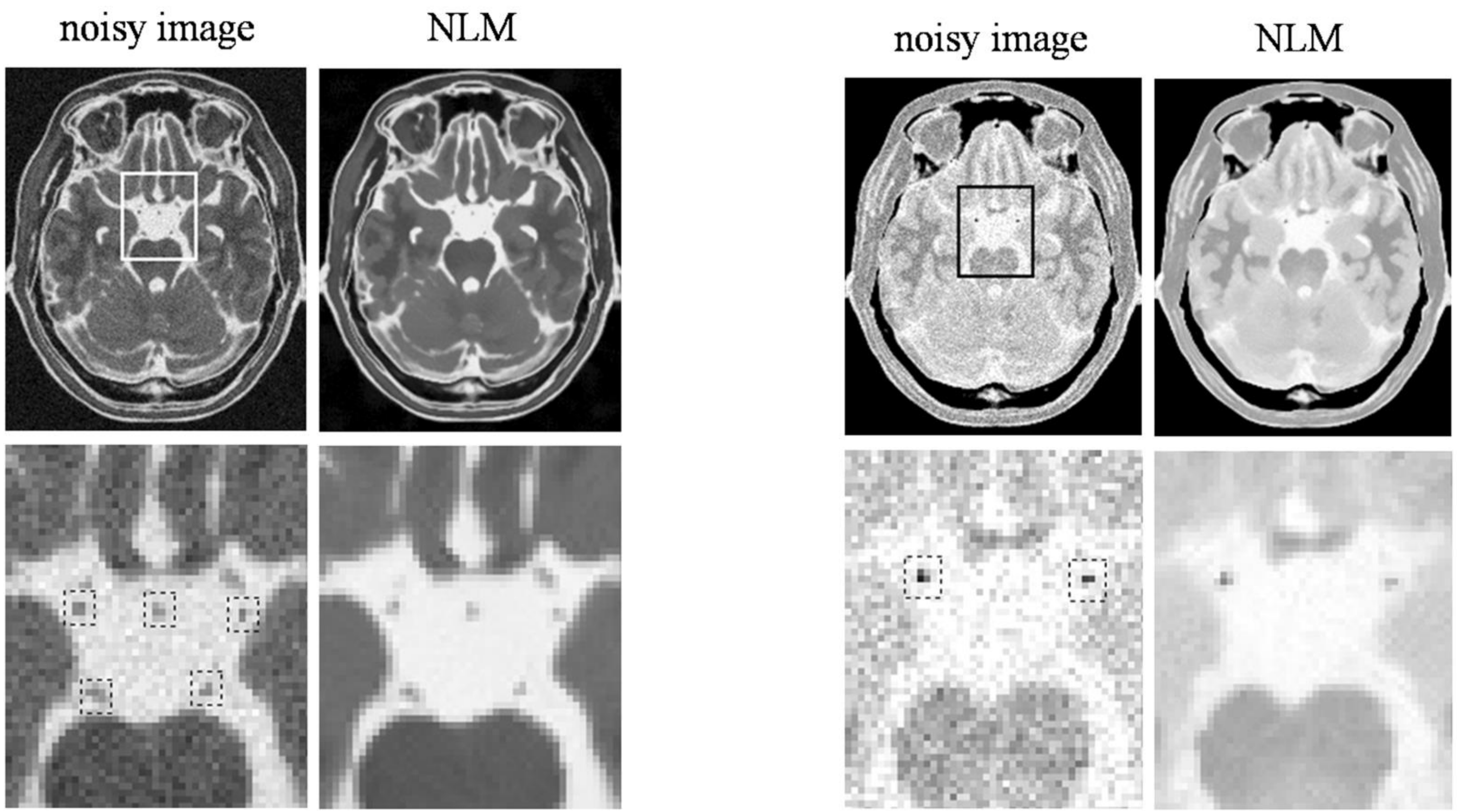
Non Local Mean (NLM) Filtering

› Similarity idea:



Non Local Mean (NLM) Filtering

› Example:



The End

- AnY QuEsTiOn?