Image Denoising Using Two-stage Non-local Means

Enming Luo

advisor: Truong Nguyen

March 15, 2012

Outline

Introduction

A noisy image is modelled as

$$\boldsymbol{Z} = \boldsymbol{U} + \boldsymbol{V}$$

- $\boldsymbol{U}$: clean image
- $\boldsymbol{V}$: noise
- $\boldsymbol{Z}$: noisy image

  Z_i: ith pixel

  $\boldsymbol{Z}_i$: patch centered at the ith pixel

Most denoising papers assume:

$$\boldsymbol{V} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$$

Denoising process:

$$\boldsymbol{Z} \to \hat{\boldsymbol{Z}} \approx \boldsymbol{U}$$

Introduction-short review

Local methods: small search window

- Bilateral filter
- Directional filters

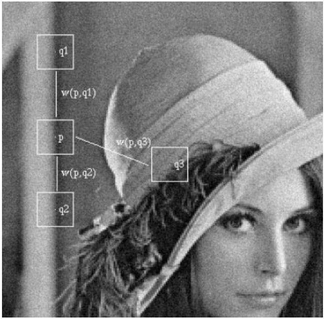
Non-local methods: large search window

- NLMeans (Non-local Means)-2005
- BM3D (Block Matching and 3D Filtering)-2007
- LPG-PCA (Local Pixel Grouping-Principal Component Analysis)-2010

NLMeans filtering

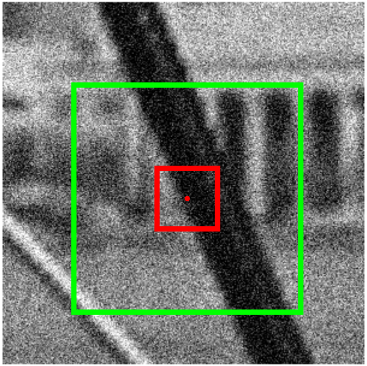
Weighted averaging:

$$\hat{Z}_i = \sum_{j \in N_i} W_{ij} Z_j$$



- NLMeans: find enough similar pixels and assign them large weights.
  - enough: search window
  - similar: similarity measure
  - weight: weight calculation

NLMeans - search window



- there is no reason that similar pixels should be close
- not restricted to a local neighbourhood

NLMeans - similarity measure

- similarity decreases in distance.
- An accurate way: squared Euclidean distance $(U_i - U_j)^2$.
- We only have the noisy data.
  - $(Z_i - Z_j)^2 \approx (U_i - U_j)^2$ (not accurate)
  - $E[(Z_i - Z_j)^2] = E[(U_i - U_j + V_i - V_j)^2] = (U_i - U_j)^2 + 2\sigma^2$
  - $(Z_i - Z_j)^2 - 2\sigma^2 \approx (U_i - U_j)^2$
- We could use patch-based distance instead of pixel-based distance.
  - Similar patches most likely have similar center pixels.
  - Use $\sum_{k=1}^{d} (\boldsymbol{Z}_i(k) - \boldsymbol{Z}_j(k))^2 - 2d\sigma^2 \approx \sum_{k=1}^{d} (\boldsymbol{U}_i(k) - \boldsymbol{U}_j(k))^2$
  - d is the number of pixels in the patch

NLMeans - weight calculation

- weight is an increasing function of similarity, or decreasing function of distance.

- Exponential kernel $e^{-\frac{x^2}{h^2}}$ is used to compute the weight.
  - x is the Euclidean distance
  - h is the decay parameter

- Specifically in NLMeans

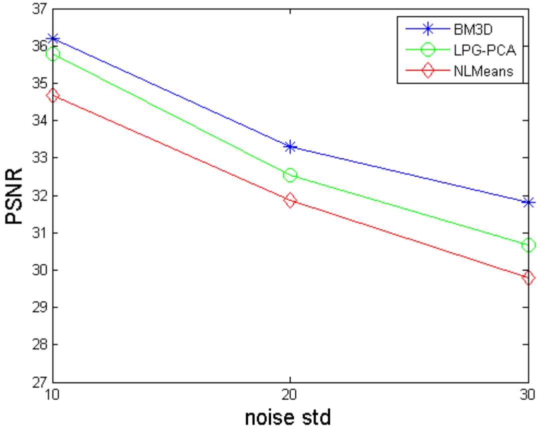
$$W_{ij} = \exp\left(-\frac{\sum_{k=1}^{d}(\boldsymbol{Z}_i(k) - \boldsymbol{Z}_j(k))^2 - 2d\sigma^2}{d\sigma^2 T^2}\right)$$

  - $\sum_{k=1}^{d}(\boldsymbol{Z}_i(k) - \boldsymbol{Z}_j(k))^2 - 2d\sigma^2$ is the approximation of squared Euclidean distance
  - $d\sigma^2$ is for normalization, T is the decay parameter

NLMeans - performance

NLMeans:

- Good: remove noises but keep details (edges, textures).



Iterative NLMeans

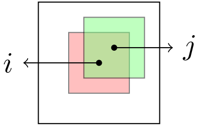
Can we apply NLMeans again to the denoised image?

- NLMeans assume IID noises:

$$Z_i : \text{IID noise,}$$
$$\hat{Z}_i = \sum_{j \in N_i} W_{ij} Z_j : \text{non-IID noise}$$

- Non-identical variance: $Var(\hat{Z}_i) = (\sum_{j \in N_i} W_{ij}^2)\sigma^2$

- $\hat{Z}_i$ and $\hat{Z}_j$ become dependent.



- Answer: No (not directly)

Modification of NLMeans

- NLMeans weight(patch size 1x1):

$$W_{ij} = \exp\left(-\frac{(Z_i - Z_j)^2 - 2\sigma^2}{\sigma^2 T^2}\right)$$

- $E[(Z_i - Z_j)^2] - Var(Z_i - Z_j) = E[Z_i - Z_j]^2$
- $(Z_i - Z_j)^2 - Var(Z_i - Z_j) \approx E[Z_i - Z_j]^2 = (U_i - U_j)^2$
- Replace $2\sigma^2$ with $Var(Z_i - Z_j)$:

$$W_{ij} = \exp\left(-\frac{(Z_i - Z_j)^2 - Var(Z_i - Z_j)}{Var(Z_i - Z_j)\,\frac{T^2}{2}}\right)$$

$$= \exp\left(-\frac{\frac{(Z_i - Z_j)^2}{Var(Z_i - Z_j)} - 1}{\frac{1}{2}T^2}\right)$$

Generalized NLMeans

- Extending to a patch, we would get the new weight:

$$W_{ij}^{G} = \exp\left(-\frac{\sum_{k=1}^{d}\left[\frac{(\boldsymbol{Z}_i(k)-\boldsymbol{Z}_j(k))^2}{Var(\boldsymbol{Z}_i(k)-\boldsymbol{Z}_j(k))} - 1\right]}{\frac{1}{2}dT^2} \right)$$

- Generalized:
  - noises in $\boldsymbol{Z}_i(k)$ and $\boldsymbol{Z}_j(k)$ not necessarily independent or identical.
  - For IID noise, $W_{ij}^{G} \equiv W_{ij}$

Iterative denoising

Back to original problem:

Can we apply NLMeans again on the denoised image?

- NLMeans has:

$$\hat{Z}_i = \sum_{j \in N_i} W_{ij} Z_j$$

- To further denoise $\hat{Z}$:

$$\tilde{Z}_i = \sum_{j \in N_i} W_{ij} \hat{Z}_j \qquad \times$$

$$\tilde{Z}_i = \sum_{j \in N_i} W_{ij}^{G} \hat{Z}_j$$

Variance calculation

$$W_{ij}^{G} = \exp\left(-\frac{\sum_{k=1}^{d} \left[\frac{(\hat{\boldsymbol{Z}}_i(k) - \hat{\boldsymbol{Z}}_j(k))^2}{Var(\hat{\boldsymbol{Z}}_i(k) - \hat{\boldsymbol{Z}}_j(k))} - 1 \right]}{\frac{1}{2} d T^2} \right)$$

- New problem: how to compute

$$Var(\hat{\boldsymbol{Z}}_i(k) - \hat{\boldsymbol{Z}}_j(k))\,?$$

- For better explanation, again assume the patch is 1x1 and

$$\hat{\boldsymbol{Z}}_i - \hat{\boldsymbol{Z}}_j = \hat{Z}_i - \hat{Z}_j$$

Variance calculation

- $\hat{Z}_i$ and $\hat{Z}_j$ are from NLMeans:

$$\hat{Z}_i = \sum_{\ell \in N_i} W_{i\ell} Z_\ell = \sum_{\ell \in N_i} W_{i\ell}(U_\ell + V_\ell)$$

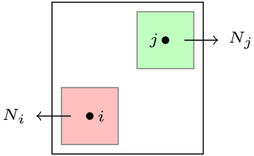
$$\hat{Z}_j = \sum_{\ell \in N_j} W_{j\ell} Z_\ell = \sum_{\ell \in N_i} W_{j\ell}(U_\ell + V_\ell)$$

- Thus we have

$$Var(\hat{Z}_i - \hat{Z}_j) = Var\Big(\sum_{\ell \in N_i} W_{i\ell} V_\ell - \sum_{\ell \in N_j} W_{j\ell} V_\ell \Big)$$

Variance calculation - no overlapping

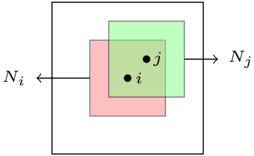
$$Var(\hat{Z}_i - \hat{Z}_j) = Var\Big(\textstyle\sum_{\ell \in N_i} W_{i\ell} V_\ell - \sum_{\ell \in N_j} W_{j\ell} V_\ell \Big)$$



$$Var(\hat{Z}_i - \hat{Z}_j) = \Big(\sum_{\ell \in N_i} W_{i\ell}^2 + \sum_{\ell \in N_j} W_{j\ell}^2 \Big) \sigma^2$$

Variance calculation - overlapped

$$Var(\hat{Z}_i - \hat{Z}_j) = Var\Big(\sum_{\ell \in N_i} W_{i\ell} V_\ell - \sum_{\ell \in N_j} W_{j\ell} V_\ell \Big)$$

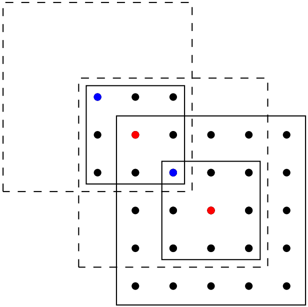


$$\sum_{\ell \in N_i} W_{i\ell} V_\ell - \sum_{\ell \in N_j} W_{j\ell} V_\ell = \sum_{\ell \in N_i \setminus N_j} W_{i\ell} V_\ell - \sum_{\ell \in N_j \setminus N_i} W_{j\ell} V_\ell + \sum_{\ell \in N_i \cap N_j} (W_{i\ell} - W_{j\ell}) V_\ell$$

$$Var(\hat{Z}_i - \hat{Z}_j) = \Big(\sum_{\ell \in N_i} W_{i\ell}^2 + \sum_{\ell \in N_j} W_{j\ell}^2 - 2 \sum_{\ell \in N_i \cap N_j} W_{i\ell} W_{j\ell} \Big) \sigma^2$$

Algorithm

- center:to be denoised, window size:5x5
- weight depends on the patch distance.
- pixel-wise variance is required.
- dependence due to overlapping



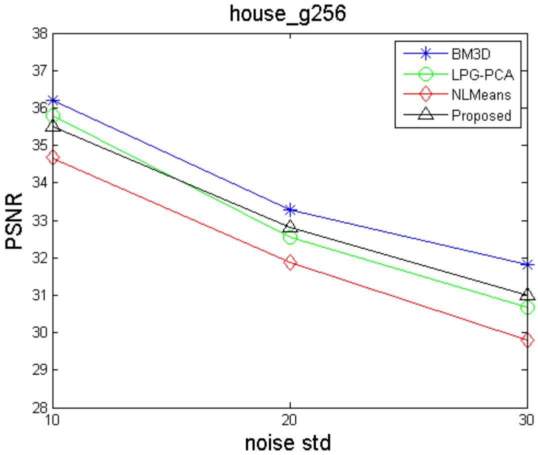
Simulation setup

- Search window size
  - small: can't provide enough similar pixels
  - large: bring more bias
- Patch size
  - small: not robust to noise
  - large: can't find enough similar pixels
- Decay parameter T
  - It controls trade-off between bias and variance
- These parameters are chosen empirically in NLMeans and we did the same way.
  - for example, if the noise variance is 10^2, search window size is 21x21, patch size is 3x3 and T is 0.4

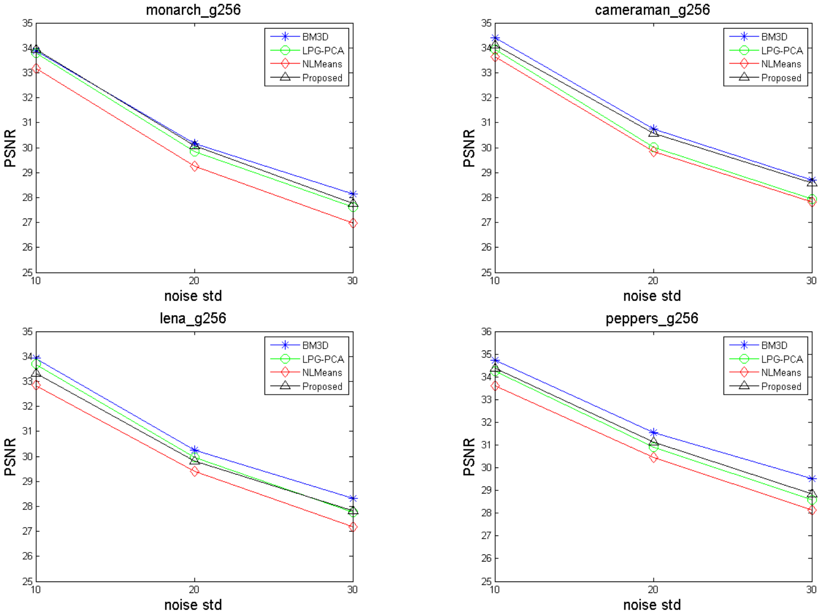
Simulation setup

- Methods for comparison
  - NLMeans
  - BM3D
  - LPG-PCA
- Test images
  - Standard test images like House, Cameraman, Monarch, Peppers.
- Noise added
  - Noises with std being 10,20 and 30 are pre-generated and then added in order to have a fair comparison between different methods.
- Comparison metric
  - PSNR
  - SSIM

Simulation results - PSNR

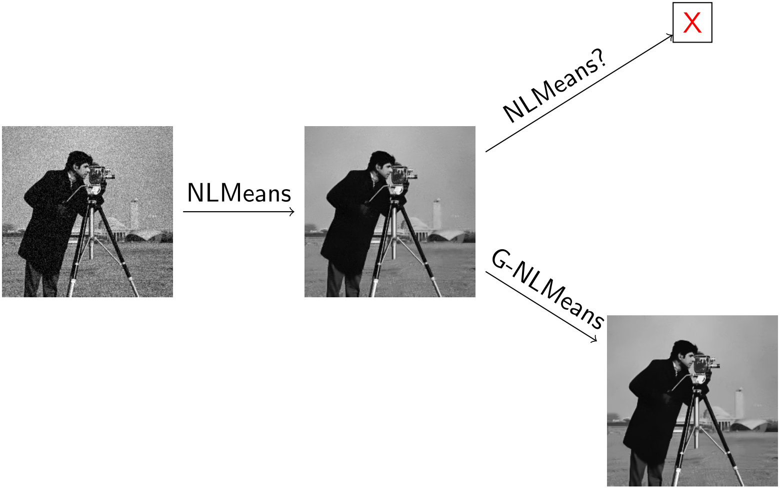


Simulation results - PSNR



Visual Quality

Conclusion



Future work

- Improve weight calculation
  - Different kernel functions, different distances.
  - Adaptive search window size, patch size and decay parameter.
- Extend to denoising real images
  - Estimate the noise variance for each pixel.
- Extend to stereo or multi-view denoising
  - More views will help the denoising.

Reference

- T., M., "Bilateral fitering for gray and color images," 1998
- B., C., M., "A review of image denoising algorithms, with a new one," 2005
- Z.,D.,Z.,S.,"Two-stage image denoising by principal component analysis with local pixel grouping," 2010
- D., F., K., E.,"Image denoising by sparse 3d transform-domain collaborative filtering," 2007
- B., C., M., "Self-similarity-based image denoising," 2011

Thank you!