# Adaptive Boosting for Image Denoising:
## Beyond Low-rank Representation and Sparse Coding

Zixiang Xiong

Texas A&M University
http://lena.tamu.edu

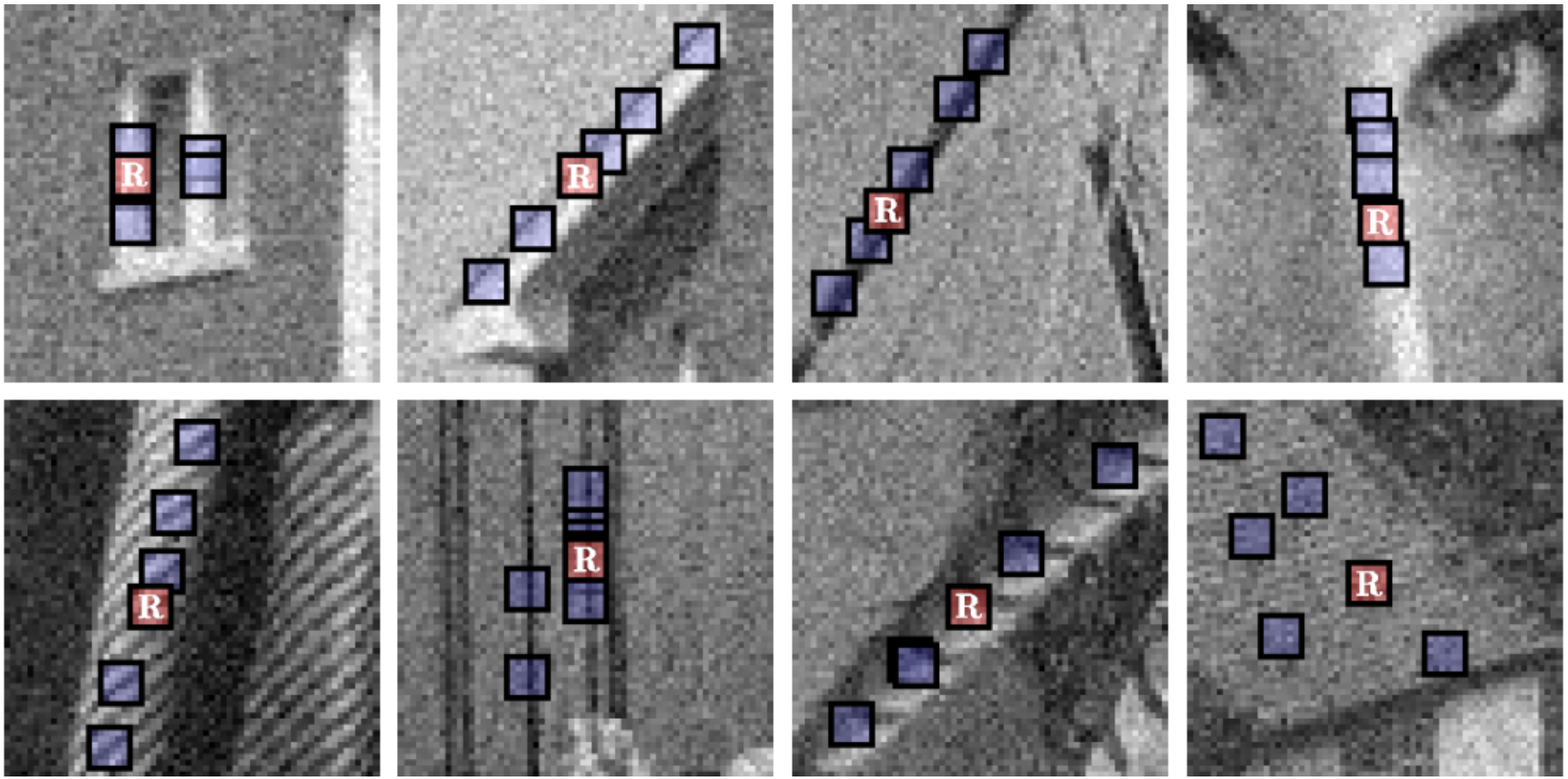
# Denoising: An old topic with wide apps

- **Recovering $\mathbf{x}$ from $\mathbf{y}=\mathbf{x}+\mathbf{n}$** has been studied from several angles
  - IP, statistics, CV & ML, coding, and optimizations
- Classic Wiener filtering technique (1949) first applied to image denoising in 1980
- Wavelet theory found wide applications in image processing in the 1990s
  - Compression (with critically sampled wavelet transform)
  - Denoising via wavelet shrinkage
    - Overcomplete representation is advantageous!
- Breakthroughs
  - Non-local mean (NLM) in 2005
  - Block-matching + 3D Wiener filtering (BM3D) in 2006

# NLM and BM3D denoising

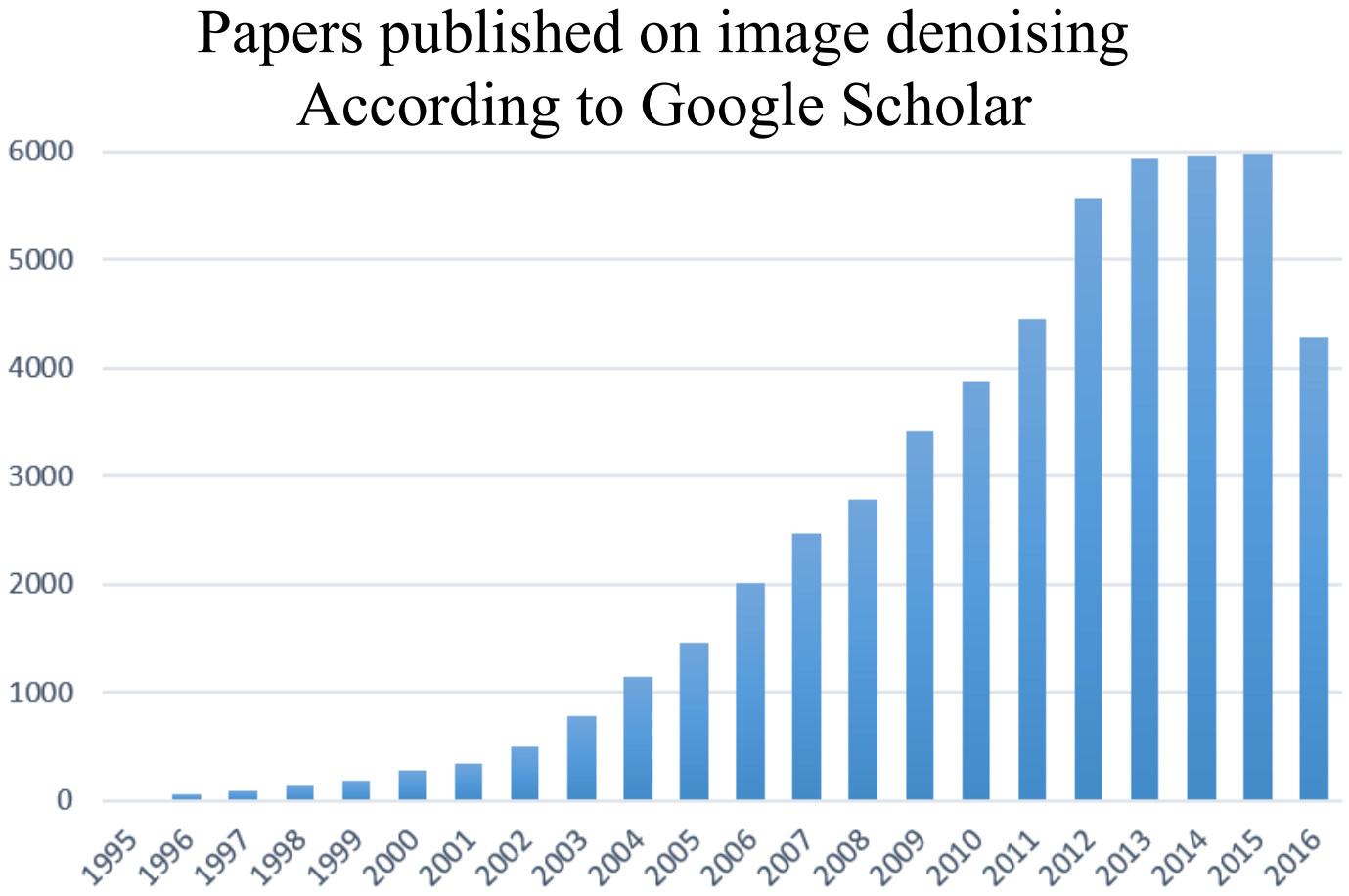
## built upon powerful non-local patch-based image models

- To exploit non-local self-similarity



# Image denoising research

- Explosive growth since NLM (2005) & BM3D (2006)



# Leading image denoising methods

Are built upon powerful patch-based image models

- NLM (2005): Self-similarity within natural images
- K-SVD (2006): Sparse representation modeling of image patches
- BM3D (2006): Combines a sparsity prior and non local self-similarity
- EPLL (2009): Expected path log likelihood
- NCSR (2011): Non-local centralized sparse representation
- SAIST (2013): Spatially adaptive iterative singular-value thresholding
- WNNM (2014): Weighted non-local nuclear mean
- …

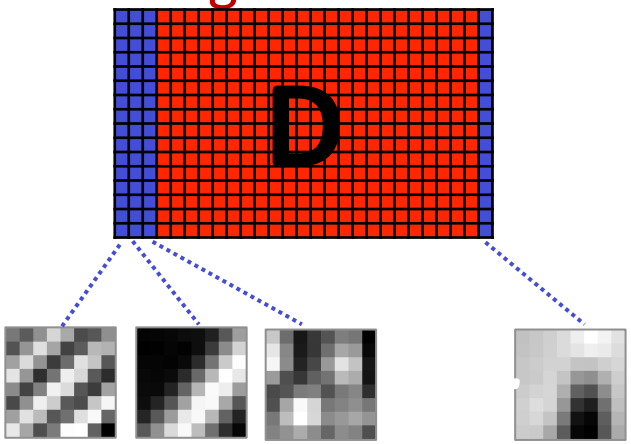
# Leading image denoising methods

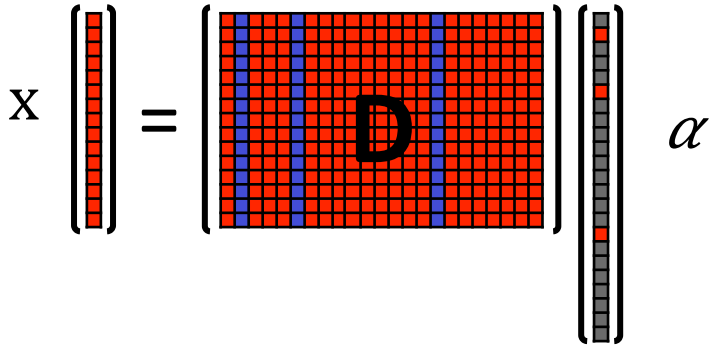
Two main components: low-rank representation & sparse coding

- Signals/images are decomposed into a low-rank representation $x = D\alpha$

  where $D$ is a fat matrix with low-rank and $\alpha$ is sparse, meaning that it contains mostly zeros

- The computation of $\alpha$ from $x$ (or its noisy version) is called sparse coding



# Theoretical advance on low-rank matrix approx.

Nuclear norm minimization (NNM) [Candès & Recht'09]

- The nuclear norm of a matrix $\boldsymbol{X} \in R^{m \times n}$ is $\|\boldsymbol{X}\|_* = \sum_i |\sigma_i(\boldsymbol{X})|$
  - Nuclear norm is the tightest convex relaxation of the rank penalty of a matrix
- Let $\boldsymbol{Y} \in R^{m \times n}$ be the given data matrix, the NNM problem aims to

$$\min_{\boldsymbol{X}} \{ \|\boldsymbol{Y} - \boldsymbol{X}\|_F^2 + \lambda \|\boldsymbol{X}\|_* \}$$

  where λ is a positive regularization parameter
- The closed-form solution (Cai & Candès & Shen'10)

$$\boldsymbol{X}^* = \boldsymbol{U} \boldsymbol{S}_\lambda(\boldsymbol{\Sigma}) \boldsymbol{V}^T$$

  with $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ being the SVD of $\boldsymbol{Y}$ and $\boldsymbol{S}_\lambda(\boldsymbol{\Sigma}) = \max(\boldsymbol{0}, \boldsymbol{\Sigma} - \lambda/2)$
  - This is just soft thresholding in the SVD domain!

# Theoretical advance on low-rank matrix approx.

Weighted nuclear norm minimization (WNNM) [Zhang et al'14]

- Extends NNM to allow non-uniform thresholding of $\sigma_i(\boldsymbol{X})$ to exploit *a priori* image info
- The weighted nuclear norm of a matrix $\boldsymbol{X} \in R^{m \times n}$ is $\|\boldsymbol{X}\|_{\boldsymbol{w},*} = \sum_i |\omega_i \sigma_i(\boldsymbol{X})|$,

  where $\boldsymbol{w} = [\omega_1, \omega_2, \ldots, \omega_m]$ is the weighting vector of nonnegative thresholds
- The WNNM problem aims at

$$\arg\min_{\boldsymbol{X}} \{ \|\boldsymbol{Y} - \boldsymbol{X}\|_F^2 + \lambda \|\boldsymbol{X}\|_{\boldsymbol{w},*} \}$$

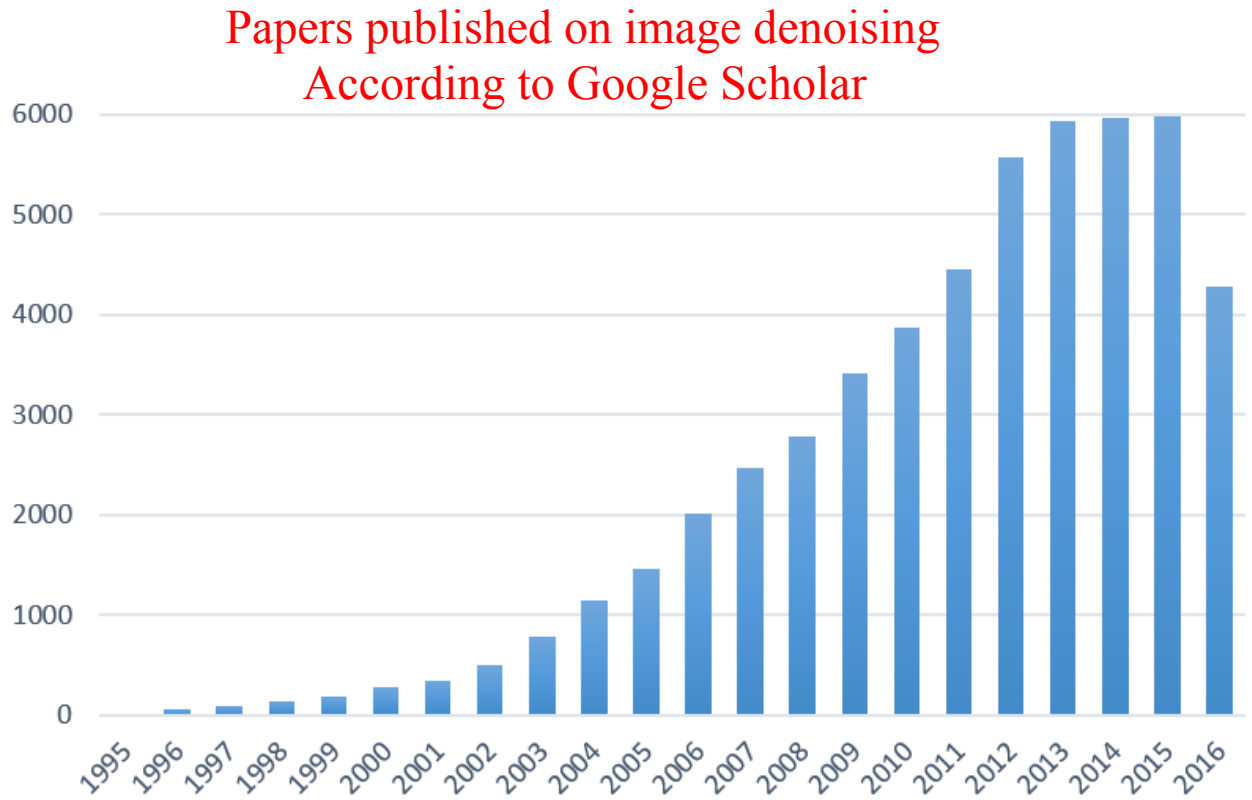
- When the weights/thresholds satisfy $0 \le \omega_1 \le \omega_2 \le \cdots \le \omega_m$ *(Zhang et al'14)*,

$$\boldsymbol{X}^* = \boldsymbol{U}\boldsymbol{S}_{\boldsymbol{W}}(\boldsymbol{\Sigma})\boldsymbol{V}^T \quad \text{and} \quad \boldsymbol{S}_{\boldsymbol{W}}(\boldsymbol{\Sigma}) = \max(\boldsymbol{0}, \Sigma_{ii} - \omega_i)$$

- Threshold $\omega_i$ is chosen empirically as [Vetterli et al'00]

# What is next?

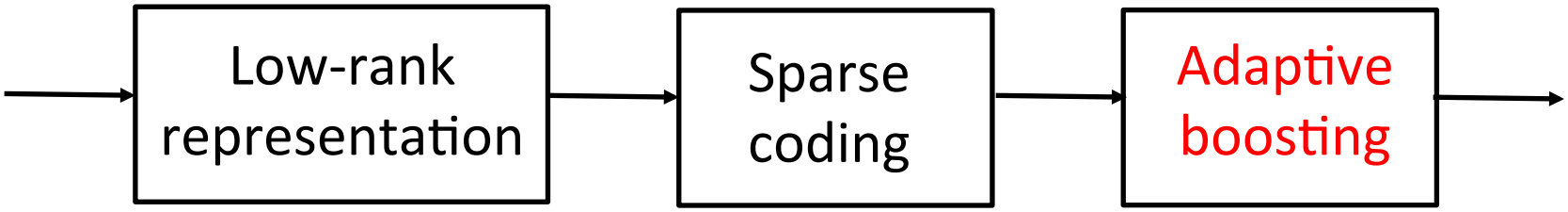
- With so much progress being made
  - Is denoising dead? [Chatterjee & Milanfar'10]




Chatterjee and Milanfar. *Is denoising dead?* IEEE Trans. Image Proc., April 2010.

# What is next?

- With so much progress being made
  - Is denoising dead? [Chatterjee & Milanfar'10]
- **How can we do better than WNNM (2014)?**
- **Beyond low-rank representation and sparse coding**




- **To get good image denoising results, one needs to delicately balance mathematical rigor and engineering approximations**

Chatterjee and Milanfar. *Is denoising dead?* IEEE Trans. Image Proc., April 2010.

# Boosting

- Formulate denoising as a regularized optimization problem: iteratively minimizing the objective function of minimizing the MSE
- Boosting performance by adaptively feeding back the residual/ method noise image

Noisy image
$\mathbf{y}$

Denoised image
$\mathbf{x} = f(\mathbf{y})$

Method noise
$\mathbf{y} - \mathbf{x}$

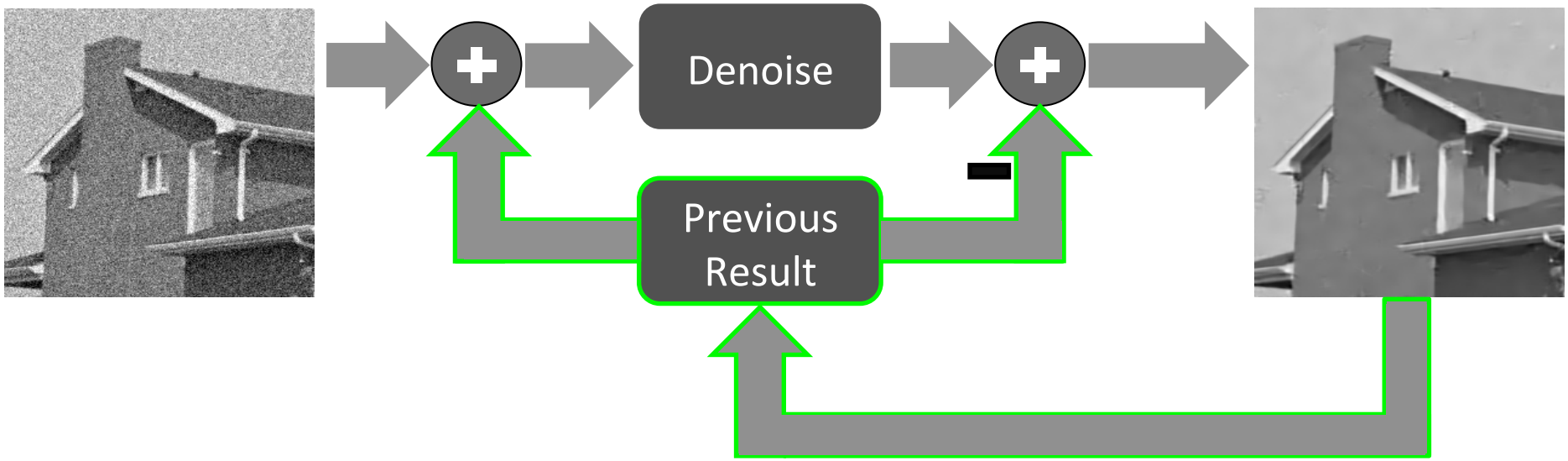
# Boosting

Multiple boosting algorithms have been explored in the past:

- Twicing [Tukey'77, Charest et al.'06]
  - $\mathbf{x}^{\uparrow k} = \mathbf{x}^{\uparrow k-1} + f(\mathbf{y} - \mathbf{x}^{\uparrow k-1})$
- Diffusion [Perona-Malik'90, Coifman et al.'06, Milanfar'12]
  - Removes the noise leftovers that are found in the denoised image
  - $\mathbf{x}^{\uparrow k} = f(\mathbf{x}^{\uparrow k-1})$
- Spatially adaptive iterative filtering (SAIF) [Talebi et al.'12]
  - Automatically chooses the local improvement mechanism:
    - Diffusion
    - Twicing

# Boosting

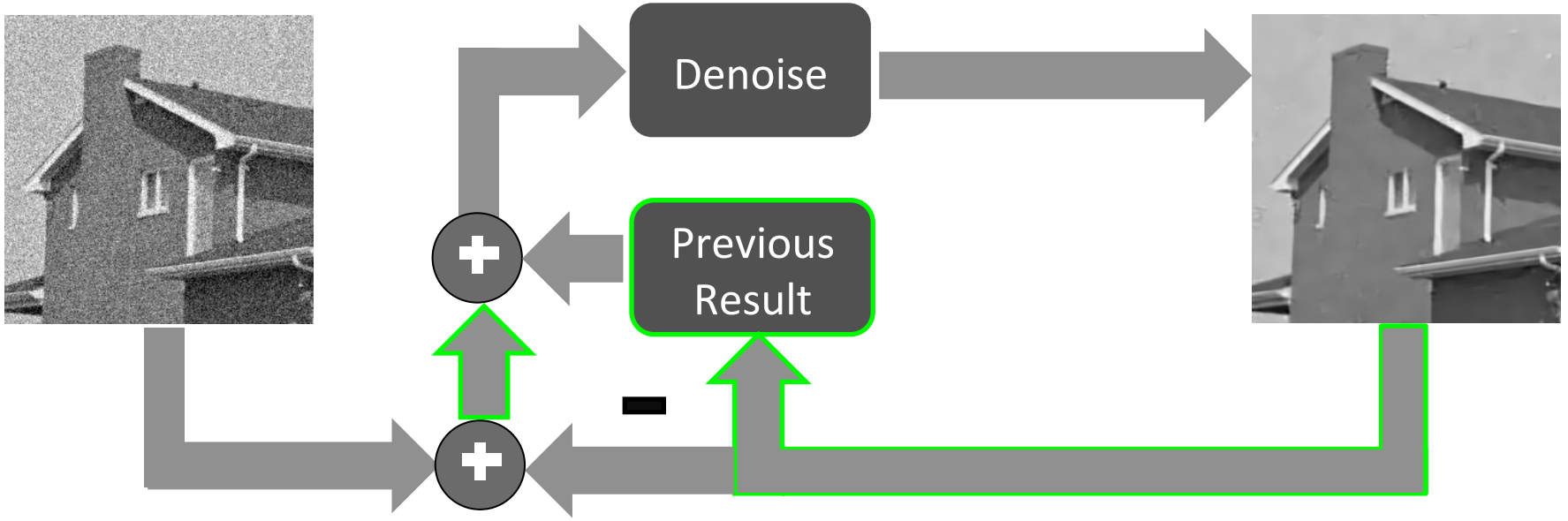
- SOS [Romano & Elad'15] (Strengthen-Operate-Subtract)
  - $\mathbf{x}\uparrow k = f(\mathbf{y} + \mathbf{x}\uparrow k-1) - \mathbf{x}\uparrow k-1$




- More general : using parameter $\rho$ to control signal emphasis
  - $\mathbf{x}\uparrow k = f(\mathbf{y} + \rho\mathbf{x}\uparrow k-1) - \rho\mathbf{x}\uparrow k-1$

# Boosting

- Annealing: WNNM [Zhang et al. '14]
  - $\mathbf{x}^{k+1} = f(\rho \mathbf{x}^{k} + (1-\rho)\mathbf{y}) = f(\mathbf{x}^{k} + (1-\rho)(\mathbf{y}-\mathbf{x}^{k}))$




- WNNM uses a constant $\rho$ for the whole image
  - $\rho$ =0.9 in experiments

# Review of WNNM

**Algorithm 1** Image denoising by WNNM

**Input:** Noisy image $y$

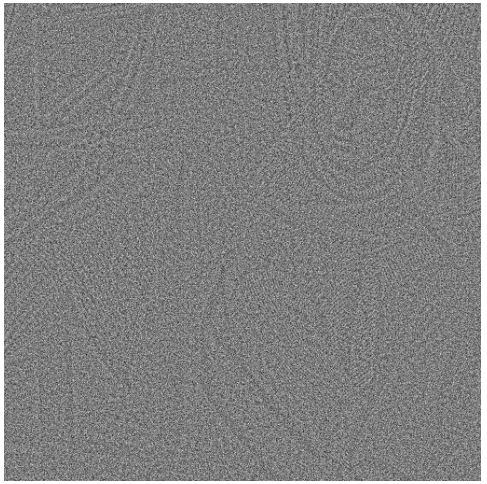
1: Initialize $\hat{x}^{(0)} = y$, $y^{(0)} = y$
2: **for** $k = 1 : K$ **do**
3: $y^{(k)} = \hat{x}^{(k-1)} + (1 - \rho)(y - \hat{x}^{(k-1)})$
4: **for** each patch $\mathbf{y}_j$ in $y^{(k)}$ **do**
5: Find similar patch group $Y_j$
6: Estimate weight vector $[w_1, w_2, ..., w_m]^T$
7: $[U, \Sigma, V] = SVD(Y_j)$
8: Get the estimation: $\hat{X}_j = US_w(\Sigma)V^T$
9: **end for**
10: Aggregate $\hat{X}_j$ to form the clean image $\hat{x}^{(k)}$
11: **end for**

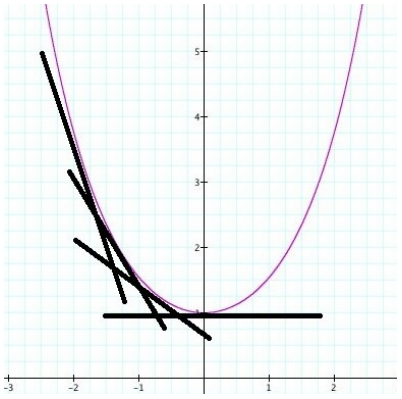
**Output:** Clean image $\hat{x}^{(K)}$

**fixed**

# Basic idea of adaptive boosting (AB)

- Instead of using a fixed feedback/boosting factor $1-\rho$, we adaptively change it from one iteration to another
- This is very intuitive
  - As the denoising performance improves from iteration to iteration, there is less and less useful structure in the method/ residual noise
  - The feedback factor $1-\rho$ should decrease from one iteration to another
- How to do adaptive boosting systematically?
  - Rank-1 based fixed-point analysis





# Fixed-point analysis

- Setup: $\mathbf{y}=\mathbf{x}+\mathbf{n}$
  - $\mathbf{y}\in R^{m\times 1}$, noisy image patch arranged in an vector
  - $\mathbf{x}\in R^{m\times 1}$, clean image patch; $\mathbf{n}\in R^{m\times 1}$, AWGN noise $\mathbf{n}\sim N(0, \sigma^2)$
- A generic iterative patch-based denoising algorithm

$$\mathbf{x}^k = f(\mathbf{x}^{k-1} + (1-\rho^k)(\mathbf{y}-\mathbf{x}^{k-1}))$$

$$= f(\rho^k \mathbf{x}^{k-1} + (1-\rho^k)\mathbf{y}) \qquad k\in Z^+$$

  - $f(\cdot)$ is nonlinear in general, but most existing algorithms can be represented as row-stochastic positive definite matrix operations $W$ [Milanfar '13]
  - Assuming convergence, assign $\mathbf{x}^k = \mathbf{x}^{k-1} = \mathbf{x}^*$

$$\mathbf{x}^* = W(\rho^* \mathbf{x}^* + (1-\rho^*)\mathbf{y})$$

# Fixed-point analysis

Fixed Point: $\mathbf{x}^{*} = (I - \rho^{*} W)^{-1} (1 - \rho^{*}) W\mathbf{y}$

- Minimize the mean square error of estimator $\mathbf{x}^{*}$ :

$$MSE(\mathbf{x}^{*}) = E[(\mathbf{x}^{*} - \mathbf{x})^{2}]$$
$$= \|bias(\mathbf{x}^{*})\|^{2} + \mathrm{var}(\mathbf{x}^{*})$$

- Bias:

$$bias(\mathbf{x}^{*}) = E(\mathbf{x}^{*}) - \mathbf{x}$$
$$= [(I - \rho^{*} W)^{-1} (1 - \rho^{*}) W - I]\mathbf{x}$$
$$= (I - \rho^{*} W)^{-1} (W - I)\mathbf{x}$$

  - Row-stochastic positive definite matrix admits an eigen-decomposition: $W = U\Lambda U^{T}$; the clean patch $\mathbf{x} = U\mathbf{b}$

$$\|bias\|^{2} = \|U(I - \rho^{*}\Lambda)^{-1} (\Lambda - I)\mathbf{b}\|^{2}$$
$$= \|(I - \rho^{*}\Lambda)^{-1} (\Lambda - I)\mathbf{b}\|, \quad (\Lambda = diag[\lambda_1, \cdots, \lambda_m])$$
$$= \sum_{i=1}^{m} (\lambda_i - 1)^{2} b_i^{2} / (1 - \rho^{*}\lambda_i)^{2}$$

# Fixed-point analysis

Fixed Point: $\mathbf{x}^{*} = (I - \rho^{*} W)^{-1} (1 - \rho^{*}) W\mathbf{y}$

- Variance:

$$var(\mathbf{x}^{*}) = tr[cov(\mathbf{x}^{*})]$$
$$= tr[cov((I - \rho^{*} W)^{-1} (1 - \rho^{*}) W\mathbf{n})]$$
$$= \sigma^2 \sum_{i=1}^{m} (1 - \rho^{*})^2 \lambda_i^2 / (1 - \rho^{*} \lambda_i)^2$$

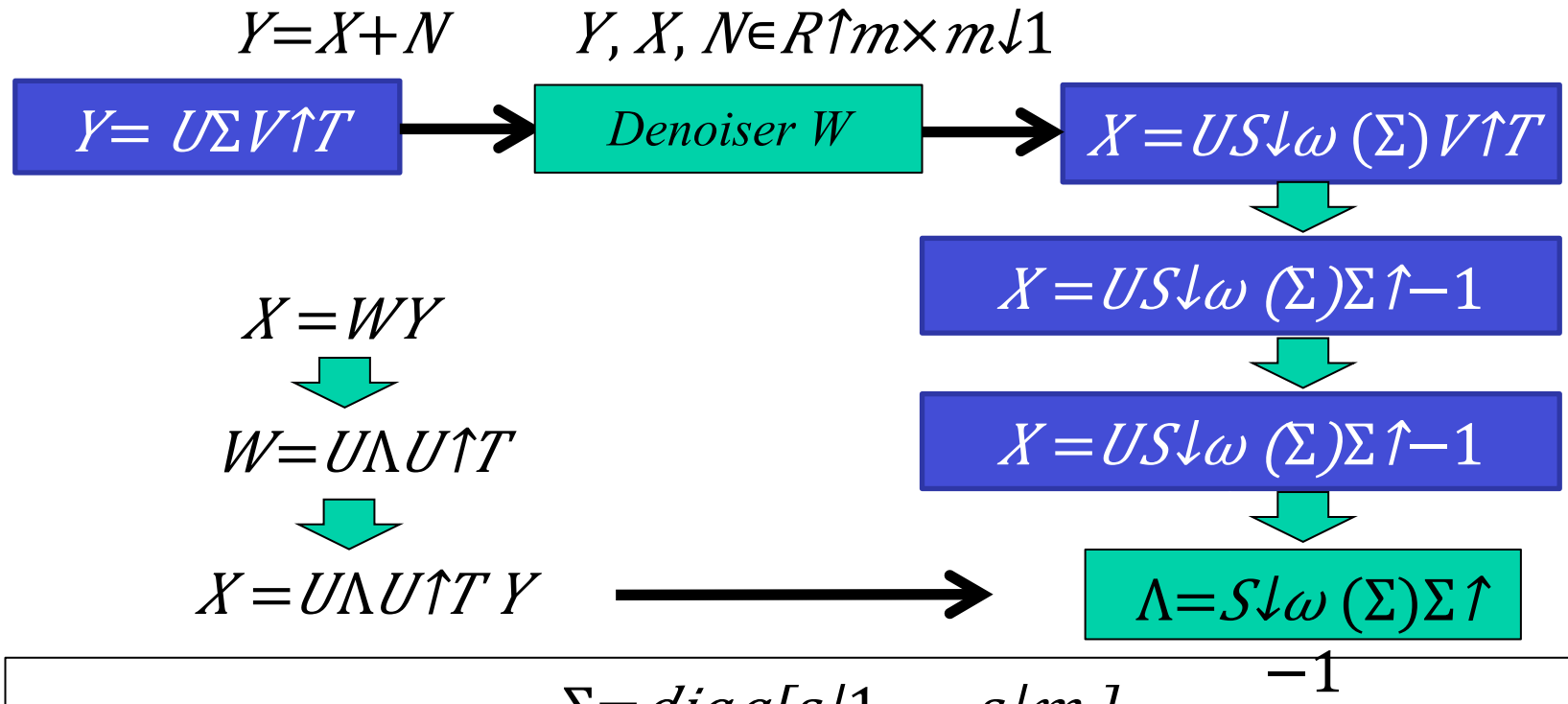
- Overall MSE:

$$MSE = \sum_{i=1}^{m} \lambda_i^2 (1 - \rho^{*})^2 \sigma^2 + (\lambda_i - 1)^2 b_i^2 / (1 - \rho^{*} \lambda_i$$

  - noise variance → $\sigma^2$
  - patch property → $b_i^2$
  - denoising algorithm → $\Lambda$, $\lambda_i^2$
  - try to find the optimal $\rho^{*}$ to minimize the MSE
- Specialize the denoising matrix $W$ to SVD-based soft thresholding and look for the optimal $\rho^{*}$

# SVD-based soft thresholding in WNNM

- Group-level setup: stack $m\downarrow 1$ similar blocks together




$$\Sigma=diag[s\downarrow 1,\cdots,s\downarrow m]$$

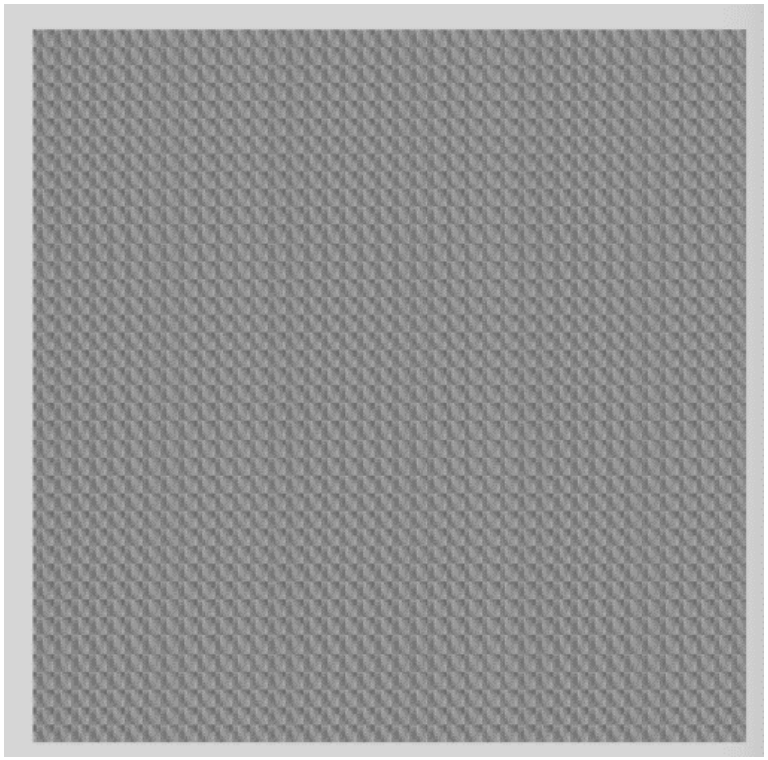
$$S\downarrow\omega(\Sigma)=diag[\max(s\downarrow 1-\omega\downarrow 1,0),\cdots,\max(s\downarrow m-\omega\downarrow m,0)]$$

$$\lambda\downarrow i=\max(1-\omega\downarrow i/s\downarrow i,0) \quad \text{for } i=1,\cdots,m$$

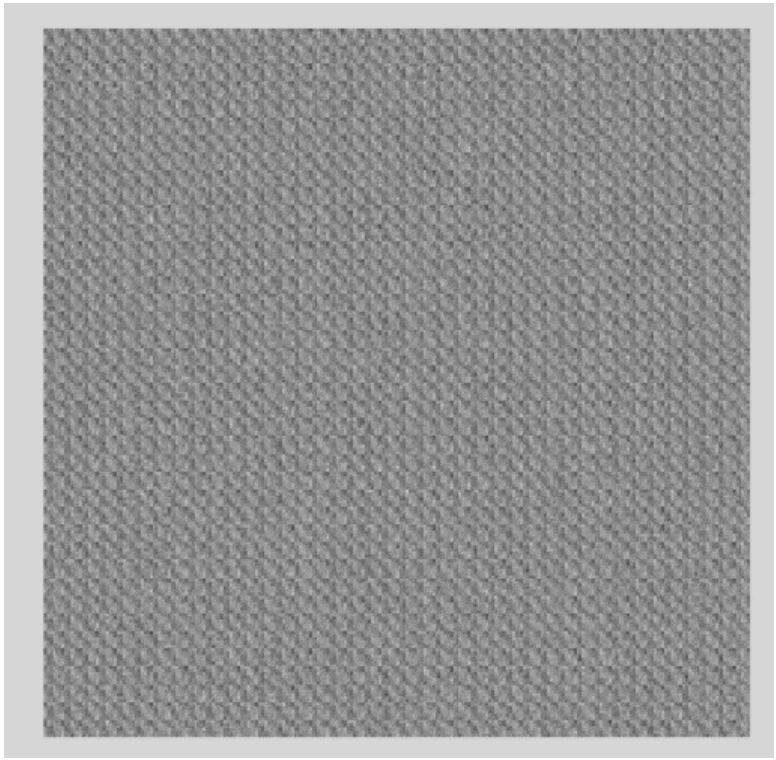
# Rank-1 based fixed-point analysis

$$\mathrm{MSE}=\sum_{i=1}^{m}\lambda_i^2\,(1-\rho^*)^2\,\sigma^2+(\lambda_i-1)^2\,b_i^2\,/(1-\rho^*\lambda_i)^2$$

- Rank-1 Assumption: All nonlocal blocks similar to $\mathbf{x}$ in the original image are identical! $Y$ is a rank-1 matrix plus Gaussian noise



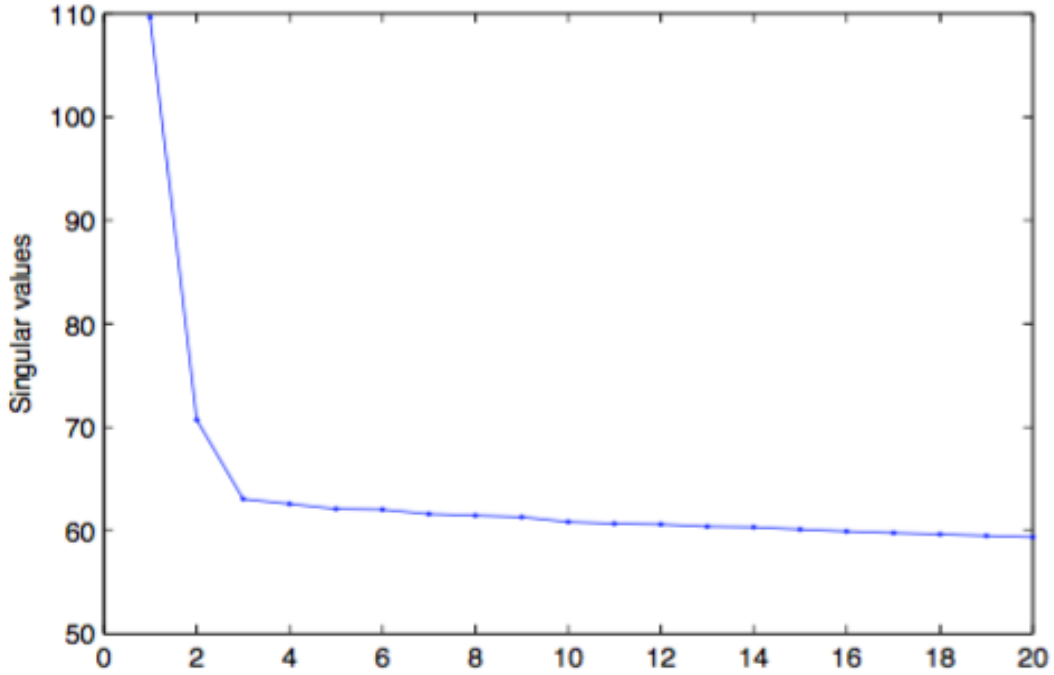
$\mathbf{x}$



$\mathbf{y}$

# Rank-1 based fixed-point analysis

- Rank-1 Assumption: All nonlocal blocks similar to $\mathbf{x}$ in the original image are identical! $Y$ is a rank-1 matrix plus Gaussian noise
  - The first singular value in $\Sigma$ dominates the rest: $s_1 \gg s_i$, for $i=2, \cdots, m$



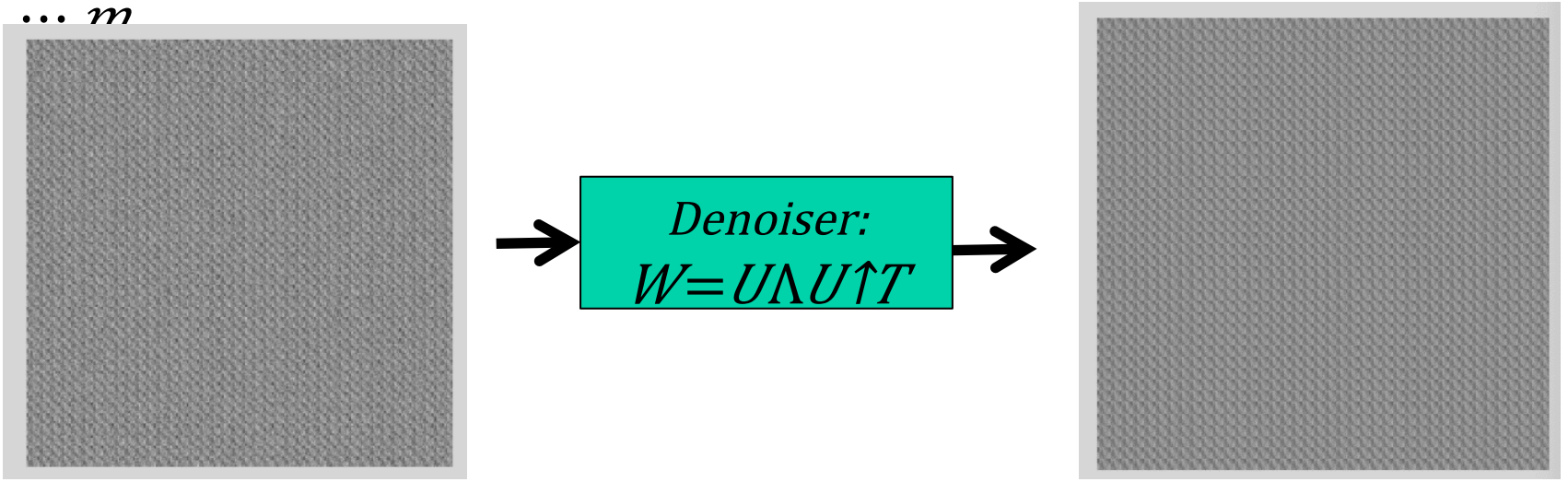

Shabalin and Nobel. *Reconstruction of a low-rank matrix in the presence of Gaussian noise*. Journal of Multivariate Analysis, 2013.

Nadakuditi. *Optshrink: An algorithm for improved low-rank signal matrix denoising by optimal, data-driven singular value shrinkage*. IEEE Trans. Info. Theory, May 2014.

# Rank-1 based fixed-point analysis

- Since $s_1 \gg s_i$, for $i=2,\cdots,m$, we have $s_i \leq \omega_i$

  $\lambda_i = \max(1-\omega_i / s_i, 0)$ for $i=1,\cdots,m$ $\rightarrow \lambda_i = 0$, for $i=2, \cdots m$




$\lambda_i^2 (1-\rho^*)^2 \sigma^2 + (\lambda_i - 1)^2 b_i^2 / (1-\rho^* \lambda_i)^2$ $= \lambda_1^2 (1-\rho^*)^2 \sigma^2 + (\lambda_1 - 1)^2 b_1^2 / (1-$

- The optimal $\rho^*$ that minimize the MSE is

  $\rho^* = \lambda_1 \sigma^2 - (1-\lambda_1) b_1^2 / \lambda_1 \sigma^2$

# Adaptive boosting (AB)

- Given the formula of optimal $\rho\uparrow *$ from fixed point analysis $\rho\uparrow * = \lambda\downarrow 1\ \sigma\uparrow 2 - (1-\lambda\downarrow 1\ ) b\downarrow 1\uparrow 2\ / \lambda\downarrow 1\ \sigma\uparrow 2$ , estimate $\lambda\downarrow 1 = \sqrt{}$ $E\downarrow \mathbf{x}\ / E\downarrow \boldsymbol{y}$ , $b\downarrow 1\uparrow 2 = E\downarrow \mathrm{x}$ , we have

$$\rho\uparrow * = \sqrt{E\downarrow \boldsymbol{y}\ } / \sqrt{E\downarrow \boldsymbol{y}\ } + \sqrt{E\downarrow \mathbf{x}}$$

- The $b\downarrow 1\uparrow 2$ stands for the energy strength in patch $\mathbf{x}$ → patches with different characteristics should be assigned different $\rho\uparrow *$ s.
- $\rho\uparrow * \in (0.5,\ 1)$ which satisfies the convergence condition
- We use the same formula for $\rho\uparrow k$ in the $k$-th iteration of AB, the adaptive boosting becomes

$$\rho\uparrow k = \sqrt{E\downarrow \mathbf{x}\ \uparrow k-1\ } / \sqrt{E\downarrow \mathbf{x}\ \uparrow k-1\ } + \sqrt{\max(E\downarrow \mathbf{x}\ \uparrow k-1\ - \sigma\uparrow 2\ , 0)}$$

- The feedback factor 1-$\rho\uparrow k$ decreases as $k$ increases ✓

# Adaptive boosting

**Algorithm 2** AB for image denoising

**Input:** Noisy image $y$

Initialize $\hat{x}^{(0)} = y$, $y^{(0)} = y$
**for** $k = 1 : K$ **do**
  **for** each patch $\mathbf{y}_j$ in $y^{(k)}$ **do**
    **Calculate** $\rho_j^{(k)}$
    $\mathbf{y}_j^{(k)} = \hat{\mathbf{x}}_j^{(k-1)} + (1 - \rho_j^{(k)})(\mathbf{y}_j - \hat{\mathbf{x}}_j^{(k-1)})$
  **end for**
  **for** each patch $\mathbf{y}_j$ in $y^{(k)}$ **do**
    Find similar patch group $Y_j$
    Estimate weight vector $w$
    $[U, \Sigma, V] = SVD(Y_j)$
    Get the estimation: $\hat{X}_j = US_w(\Sigma)V^T$
  **end for**
  Aggregate $\hat{X}_j$ to form the clean image $\hat{x}^{(k)}$
**end for**

**Output:** Clean image $\hat{x}^{(K)}$

$$\rho\uparrow k = \sqrt{E\downarrow \mathbf{x}\uparrow k-1} \;/\sqrt{E\downarrow \mathbf{x}\uparrow k-1} + \sqrt{\max(E\downarrow \mathbf{x}\uparrow k-1 - \sigma\uparrow 2, 0)}$$

# Convergence analysis

Assuming matrix approximation of $f(\cdot)\approx W$

- Difference between the $k$-th estimate and the fixed point

$$\boldsymbol{e}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^{(*)}$$

$$= W(\rho^{(k)}\mathbf{x}^{(k-1)} + (1-\rho^{(k)})\mathbf{y}) - W(\rho^{(*)}\mathbf{x}^{(*)} + (1-\rho^{(*)})\mathbf{y})$$

- After a large number of iterations, replace $\rho^{(k)}$ by $\rho^{(*)}$

$$\boldsymbol{e}^{(k)} = \rho^{(*)} W(\mathbf{x}^{(k-1)} - \mathbf{x}^{(*)}) = \rho^{(*)} W\boldsymbol{e}^{(k-1)} = \rho^{(*)k} W^{k} \boldsymbol{e}^{(0)}$$

- We have $0<\rho^{(*)}<1$ and $0\leq W<1$ (the denoising operator is contractive)
  - $\boldsymbol{e}^{(k)} \to \mathbf{0}$ as $k\to\infty$

# Test images

The 20 test images commonly used in experiments:

# Numerical results

AB gives the best PSNR result for every images (20 test images) and the improvement over others increases with noise variance

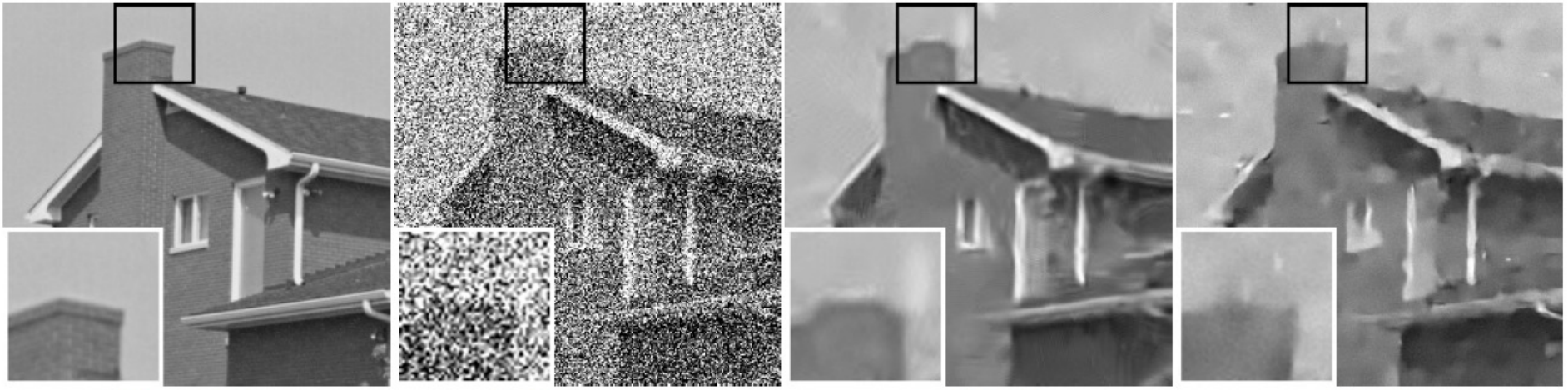
| | $\sigma_n = 10$ | | | | | | | $\sigma_n = 30$ | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BM3D | EPLL | LSSC | NCSR | SAIST | WNNM | AB | BM3D | EPLL | LSSC | NCSR | SAIST | WNNM | AB |
| C.Man | 34.18 | 34.02 | 34.24 | 34.18 | 34.30 | 34.44 | **34.49** | 28.64 | 28.36 | 28.63 | 28.59 | 28.36 | 28.80 | **28.96** |
| House | 36.71 | 35.75 | 36.95 | 36.80 | 36.66 | 36.95 | **37.03** | 32.09 | 31.23 | 32.41 | 32.07 | 32.30 | 32.52 | **32.68** |
| Peppers | 34.68 | 34.54 | 34.80 | 34.68 | 34.82 | 34.95 | **34.97** | 29.28 | 29.16 | 29.25 | 29.10 | 29.24 | 29.49 | **29.66** |
| Montage | 37.35 | 36.49 | 37.26 | 37.17 | 37.46 | 37.84 | **37.91** | 31.38 | 30.17 | 31.10 | 30.92 | 31.06 | 31.65 | **31.83** |
| Leaves | 34.04 | 33.29 | 34.52 | 34.53 | 34.92 | 35.20 | **35.29** | 27.81 | 27.18 | 27.65 | 28.14 | 28.29 | 28.60 | **28.71** |
| StarFish | 33.30 | 33.29 | 33.74 | 33.65 | 33.72 | 33.99 | **34.04** | 27.65 | 27.52 | 27.70 | 27.78 | 27.92 | 28.08 | **28.16** |
| Monarch | 34.12 | 34.27 | 34.44 | 34.51 | 34.76 | 35.03 | **35.07** | 28.36 | 28.35 | 28.20 | 28.46 | 28.65 | 28.92 | **29.01** |
| Airplane | 33.33 | 33.39 | 33.51 | 33.40 | 33.43 | 33.64 | **33.67** | 27.56 | 27.67 | 27.53 | 27.53 | 27.66 | 27.83 | **27.95** |
| Paint | 34.00 | 34.01 | 34.35 | 34.15 | 34.28 | 34.50 | **34.53** | 28.29 | 28.33 | 28.29 | 28.10 | 28.44 | 28.58 | **28.71** |
| J.Bean | 37.91 | 37.63 | 38.69 | 38.31 | 38.37 | 38.93 | **39.04** | 31.97 | 31.56 | 32.39 | 32.13 | 32.14 | 32.46 | **32.67** |
| Fence | 33.50 | 32.89 | 33.60 | 33.65 | 33.76 | 33.93 | **33.96** | 28.19 | 27.23 | 28.16 | 28.23 | 28.26 | 28.56 | **28.66** |
| Parrot | 33.57 | 33.58 | 33.62 | 33.56 | 33.66 | 33.81 | **33.85** | 28.12 | 28.07 | 27.99 | 28.07 | 28.12 | 28.33 | **28.40** |
| Lena | 35.93 | 35.58 | 35.83 | 35.85 | 35.90 | 36.03 | **36.08** | 31.26 | 30.79 | 31.18 | 31.06 | 31.27 | 31.43 | **31.54** |
| Barbara | 34.98 | 33.61 | 34.98 | 35.00 | 35.24 | 35.51 | **35.55** | 29.81 | 27.57 | 29.60 | 29.62 | 30.14 | 30.31 | **30.41** |
| Boat | 33.92 | 33.66 | 34.01 | 33.91 | 33.91 | 34.09 | **34.12** | 29.12 | 28.89 | 29.06 | 28.94 | 28.98 | 29.24 | **29.36** |
| Hill | 33.62 | 33.48 | 33.66 | 33.69 | 33.65 | 33.79 | **33.85** | 29.16 | 28.90 | 29.09 | 28.97 | 29.06 | 29.25 | **29.36** |
| F.print | 32.46 | 32.12 | 32.57 | 32.68 | 32.69 | 32.82 | **32.84** | 26.83 | 26.19 | 26.68 | 26.92 | 26.95 | 26.99 | **27.13** |
| Man | 33.98 | 33.97 | 34.10 | 34.05 | 34.12 | 34.23 | **34.28** | 28.86 | 28.83 | 28.87 | 28.78 | 28.81 | 29.00 | **29.14** |
| Couple | 34.04 | 33.85 | 34.01 | 34.00 | 33.96 | 34.14 | **34.18** | 28.87 | 28.62 | 28.77 | 28.57 | 28.72 | 28.98 | **29.07** |
| Straw | 30.89 | 30.74 | 31.25 | 31.35 | 31.49 | 31.62 | **31.63** | 24.84 | 24.64 | 24.99 | 25.00 | 25.23 | 25.27 | **25.35** |
| AVE. | 34.33 | 34.01 | 34.51 | 34.46 | 34.56 | 34.77 | **34.82** | 28.91 | 28.46 | 28.88 | 28.85 | 28.98 | 29.21 | **29.34** |

# Numerical results

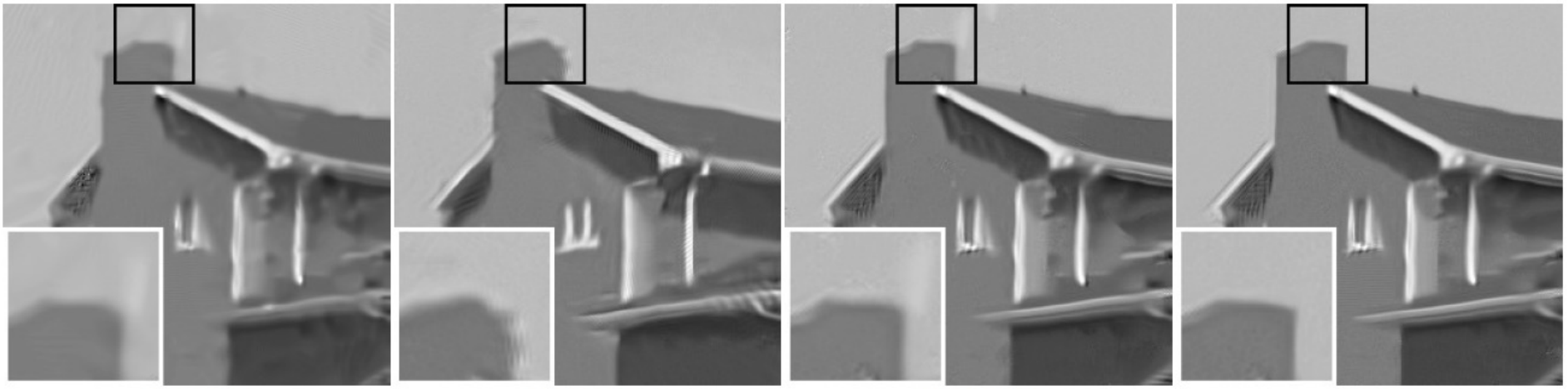
AB gives the best PSNR result for every images (20 test images) and the improvement over others increases with noise variance

| | $\sigma_n = 50$ | | | | | | | $\sigma_n = 100$ | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C.Man | 26.12 | 26.02 | 26.35 | 26.14 | 26.15 | 26.42 | **26.64** | 23.07 | 22.86 | 23.15 | 22.93 | 23.09 | 23.36 | **23.75** |
| House | 29.69 | 28.76 | 29.99 | 29.62 | 30.17 | 30.32 | **30.62** | 25.87 | 25.19 | 25.71 | 25.56 | 26.53 | 26.68 | **27.35** |
| Peppers | 26.68 | 26.63 | 26.79 | 26.82 | 26.73 | 26.91 | **27.11** | 23.39 | 23.08 | 23.20 | 22.84 | 23.32 | 23.46 | **23.85** |
| Montage | 27.90 | 27.17 | 28.10 | 27.84 | 28.00 | 28.27 | **28.54** | 23.89 | 23.42 | 23.77 | 23.74 | 23.98 | 24.16 | **24.60** |
| Leaves | 24.68 | 24.38 | 24.81 | 25.04 | 25.25 | 25.47 | **25.65** | 20.91 | 20.25 | 20.58 | 20.86 | 21.40 | 21.57 | **21.80** |
| StarFish | 25.04 | 25.04 | 25.12 | 25.07 | 25.29 | 25.44 | **25.59** | 22.10 | 21.92 | 21.77 | 21.91 | 22.10 | 22.22 | **22.50** |
| Monarch | 25.82 | 25.78 | 25.88 | 25.73 | 26.10 | 26.32 | **26.48** | 22.52 | 22.23 | 22.24 | 22.11 | 22.61 | 22.95 | **23.21** |
| Airplane | 25.10 | 25.24 | 25.25 | 24.93 | 25.34 | 25.43 | **25.61** | 22.11 | 22.02 | 21.69 | 21.83 | 22.27 | 22.55 | **22.91** |
| Paint | 25.67 | 25.77 | 25.59 | 25.37 | 25.77 | 25.98 | **26.19** | 22.51 | 22.50 | 22.14 | 22.11 | 22.42 | 22.74 | **23.10** |
| J.Bean | 29.26 | 28.75 | 29.42 | 29.29 | 29.32 | 29.62 | **29.79** | 25.80 | 25.17 | 25.64 | 25.66 | 25.82 | 26.04 | **26.31** |
| Fence | 25.92 | 24.58 | 25.87 | 25.78 | 26.00 | 26.43 | **26.56** | 22.92 | 21.11 | 22.71 | 22.23 | 22.98 | 23.37 | **23.69** |
| Parrot | 25.90 | 25.84 | 25.82 | 25.71 | 25.95 | 26.09 | **26.17** | 22.96 | 22.71 | 22.79 | 22.53 | 23.04 | 23.19 | **23.38** |
| Lena | 29.05 | 28.42 | 28.95 | 28.90 | 29.01 | 29.24 | **29.40** | 25.95 | 25.30 | 25.96 | 25.71 | 25.93 | 26.20 | **26.52** |
| Barbara | 27.23 | 24.82 | 27.03 | 26.99 | 27.51 | 27.79 | **27.96** | 23.62 | 22.14 | 23.54 | 23.20 | 24.07 | 24.37 | **24.68** |
| Boat | 26.78 | 26.65 | 26.77 | 26.66 | 26.63 | 26.97 | **27.15** | 23.97 | 23.71 | 23.87 | 23.68 | 23.80 | 24.10 | **24.36** |
| Hill | 27.19 | 26.96 | 27.14 | 26.99 | 27.04 | 27.34 | **27.52** | 24.58 | 24.43 | 24.47 | 24.36 | 24.29 | 24.75 | **25.11** |
| F.print | 24.53 | 23.59 | 24.26 | 24.48 | 24.52 | 24.67 | **24.81** | 21.61 | 19.85 | 21.30 | 21.39 | 21.62 | 21.81 | **21.96** |
| Man | 26.81 | 26.72 | 26.72 | 26.67 | 26.68 | 26.94 | **27.14** | 24.22 | 24.07 | 23.98 | 24.02 | 24.01 | 24.36 | **24.65** |
| Couple | 26.46 | 26.24 | 26.35 | 26.19 | 26.30 | 26.65 | **26.85** | 23.51 | 23.32 | 23.27 | 23.15 | 23.21 | 23.55 | **23.86** |
| Straw | 22.29 | 21.93 | 22.51 | 22.30 | 22.65 | 22.74 | **22.86** | 19.43 | 18.84 | 19.43 | 19.10 | 19.42 | 19.67 | **19.98** |
| AVE. | 26.41 | 25.97 | 26.44 | 26.33 | 26.52 | 26.75 | **26.93** | 23.25 | 22.71 | 23.06 | 23.00 | 23.30 | 23.56 | **23.88** |

# Visual comparison



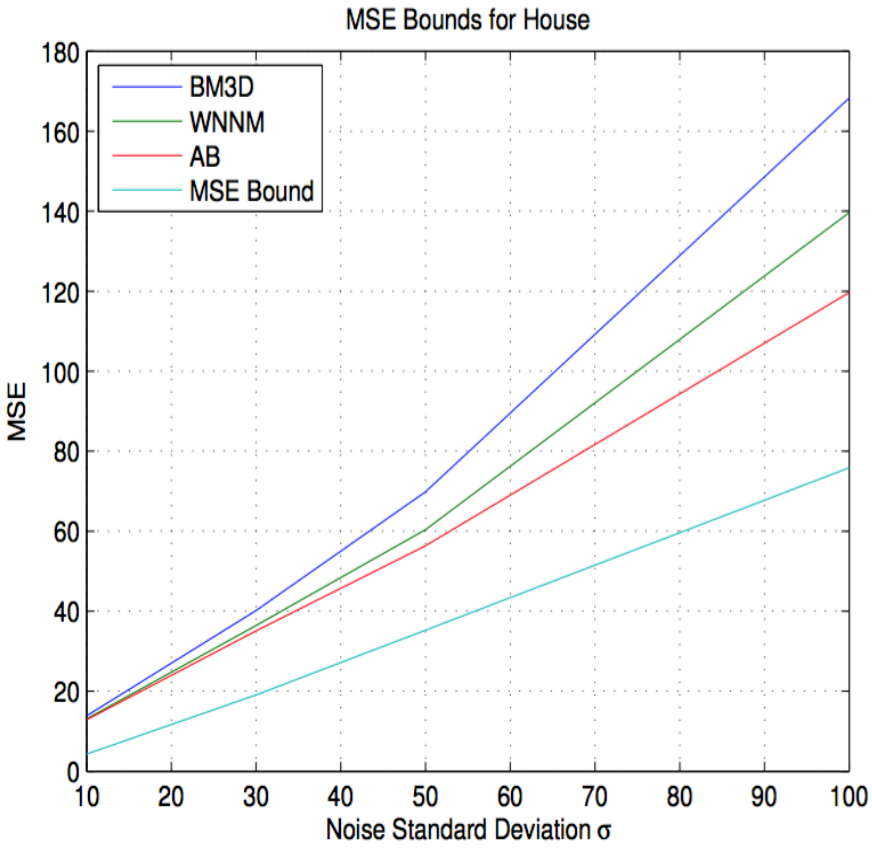
(a) Ground Truth (b) Noisy $\sigma$=100 (c) BM3D (d) EPLL

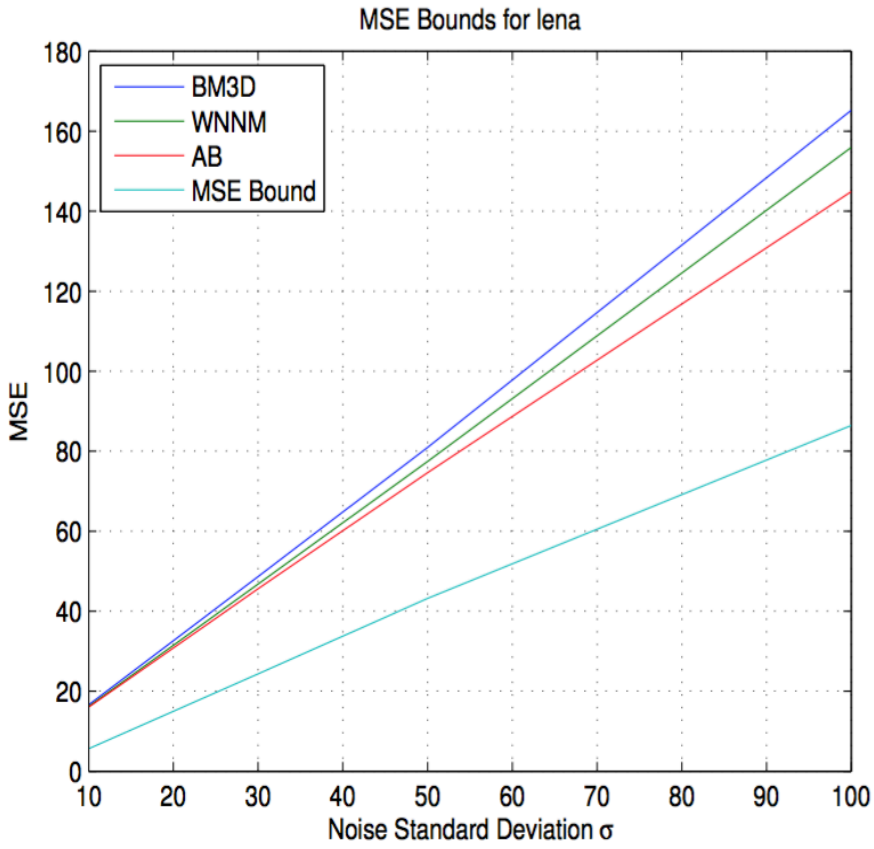


(e) NCSR (f) SAIST (g) WNNM (h) AB

# Comparison with lower bound

AB performs closest to the Cramer-Rao lower bound




Chatterjee and Milanfar. *Is denoising dead?* IEEE Trans. Image Proc., April 2010.

# Remarks

- AB works as a preprocessing step in each iteration before the image is denoised by leading method
- Complexity of AB is ignorable compared with the main denoising algorithm
- AB is block-based to adapt to non-stationarity
- The idea of AB is very generic
  - Can be combined with training/learning-based approach
- AB is applicable to other representative methods
  - Such as BM3D and K-SVD

# Remarks

- AB is applicable to other representative methods
  - Fresh PSNR results (in dB) from BM3D+AB

| Sigma = 25 | BM3D | BM3D+SOS | BM3D+AB |
|---|---|---|---|
| Foreman | 33.41 | 33.48 | 33.49 |
| Lena | 32.02 | 32.04 | 32.05 |
| House | 32.90 | 32.90 | 32.93 |
| FingerPrint | 27.72 | 27.72 | 27.74 |
| Peppers | 31.87 | 31.89 | 31.90 |

# Image denoising -- for real!

- Images taken by low- and high-end smart phones


4160x3120


3264x2448

# Image denoising -- for real!

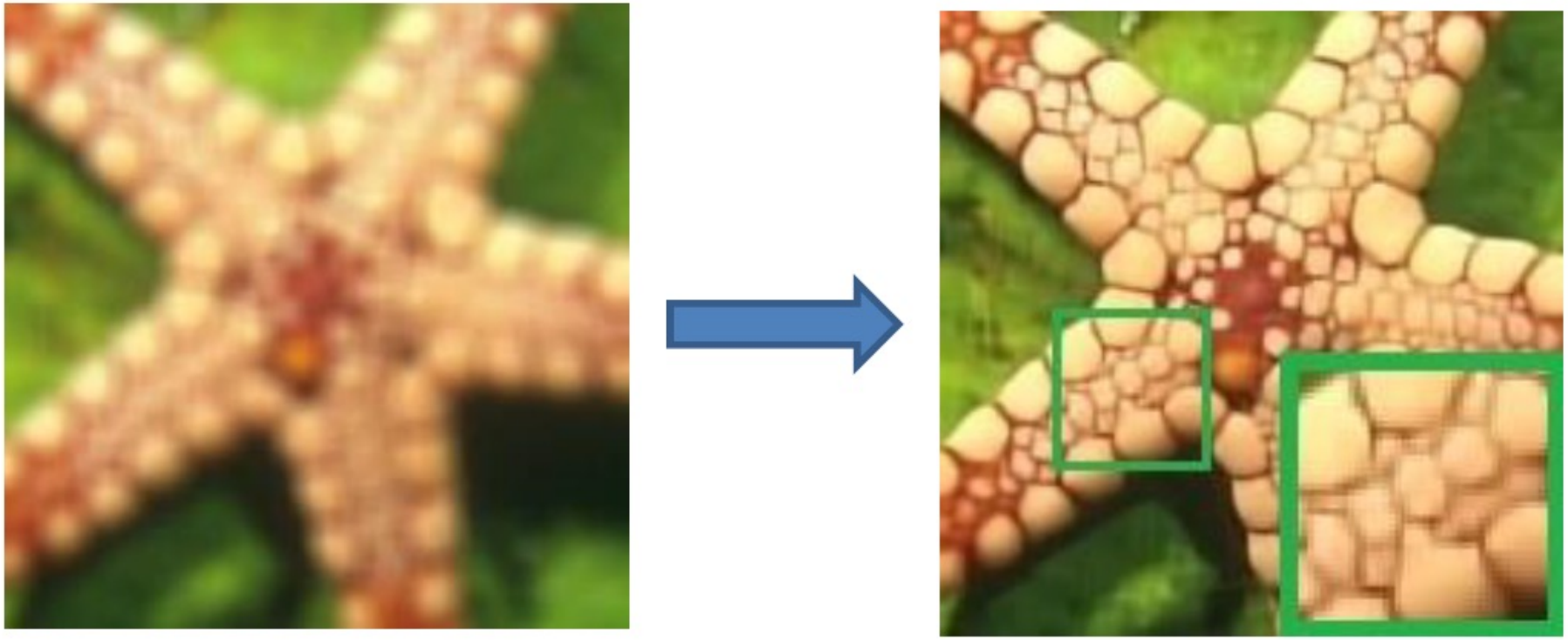
- Images taken by low- and high-end smart phones


1259x771 window

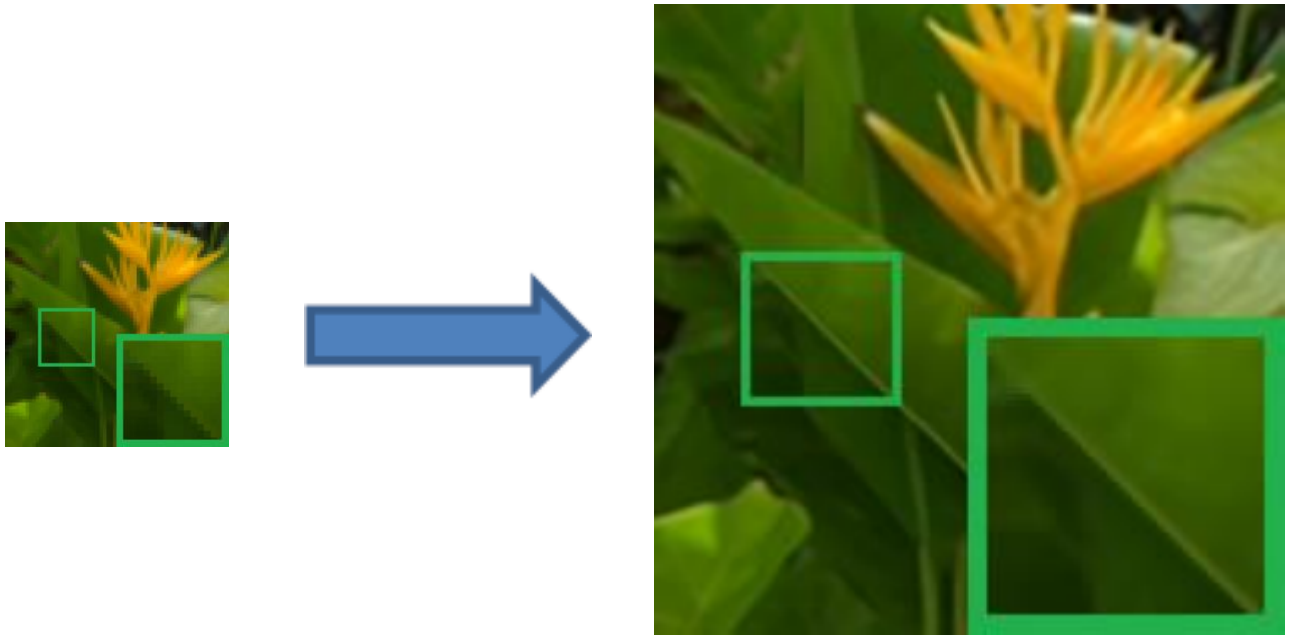

1031x754 window

# Image deblurring

- Could be part of denoising



# Image super-resolution



# Face super-resolution for recognition

# Image inpainting