Image Filters

C.-C. Jay Kuo
University of Southern California

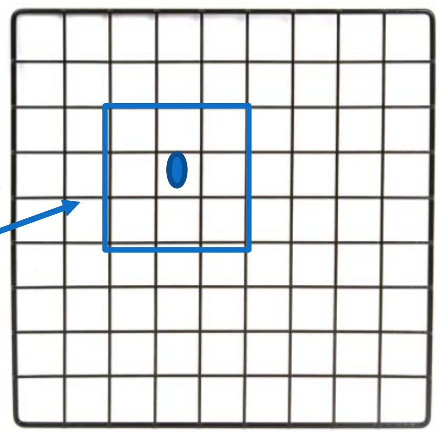
Image Filters

- Usually, an image filter is represented by an NxN array.
  - Each number in the array represents a weight coefficient for each pixel covered by the array

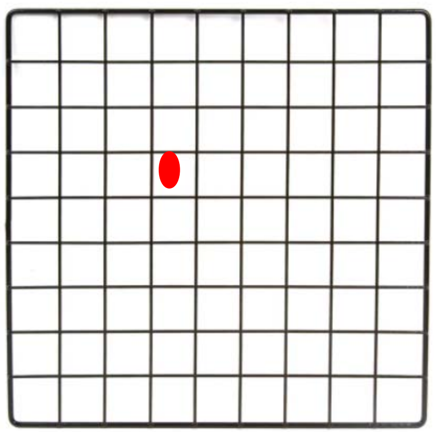
$$\begin{bmatrix} a_{-1,1} & a_{0,1} & a_{1,1} \\ a_{-1,0} & a_{0,0} & a_{1,0} \\ a_{-1,-1} & a_{0,-1} & a_{1,-1} \end{bmatrix}$$

element-wise multiplication

Input image



output image



Example: Lowpass filters

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

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

Example: Highpass filters

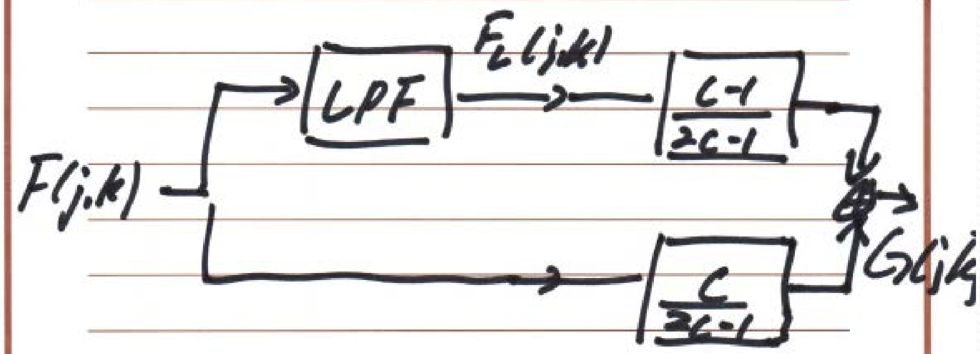
$$\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} \qquad \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Example: Edge Sharpening

- Unsharp masking

$$G(j,k) = \frac{c}{2c-1}F(j,k) - \frac{1-c}{2c-1}F_L(j,k)$$

Pratt's Book
Page 286



$$3/5 \le c \le 5/6$$

$$c = 0.6 \qquad \frac{c-1}{2c-1} = -\frac{0.4}{0.2} = -2$$

$$\frac{c}{2c-1} = \frac{0.6}{0.2} = 3$$

Illustration of 1D Unsharp Masking

Pratt’s Book
Page 286

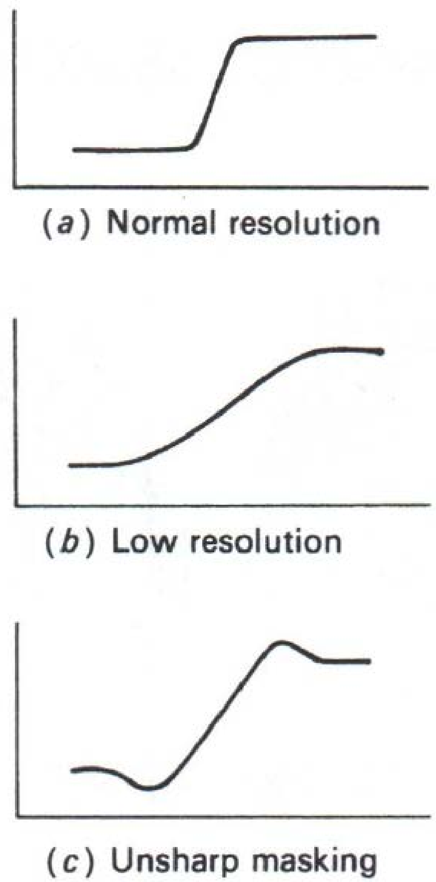


FIGURE 10.4-2. Waveforms in an unsharp masking image enhancement system.

Image Denoising

C.-C. Jay Kuo
University of Southern California

Outline

- Noise Types
- Removal of impulse noise
- Removal of additive white Gaussian noise
- Conclusions

Two Noise Types

Additive Noise

$$X(j,k)=S(j,k)+N(j,k)$$

Clean Source Image

Gaussian Noise

Uniform Noise

AWGN

- Additive White Gaussian Noise
  - White
    - N(j,k) and N(j',k') are uncorrelated
  - Gaussian
    - N(j,k) is a Gaussian random variable

Impulse Noise (Pepper/Salt Noise)

White dots: salt noise (caused by sensor saturation)

Black dots: pepper noise (caused by dead sensor)

Mixed Noise

Removal of Salt & Pepper Noise

Outlier Detection

$O_1 \quad O_2 \quad O_3$

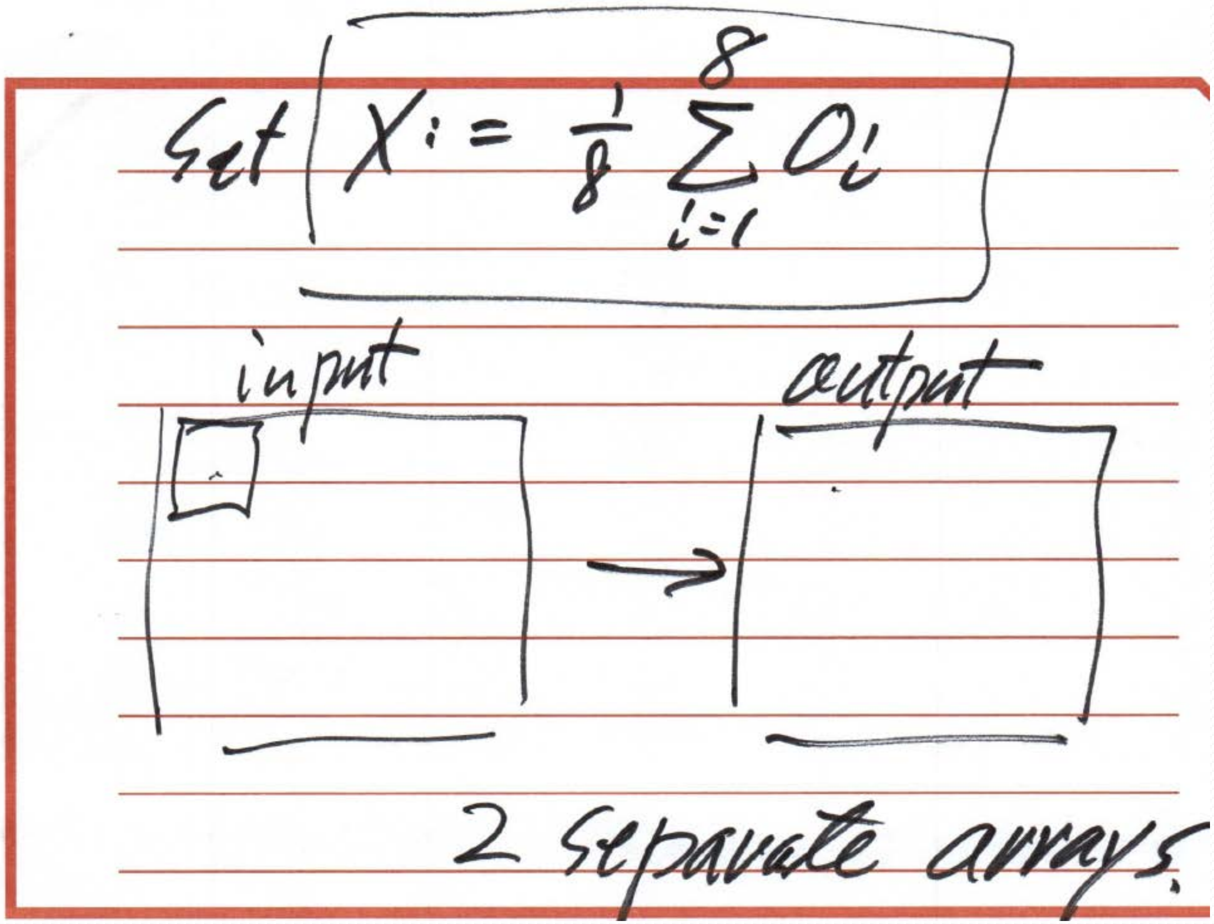
$O_4 \quad X \quad O_5$

$O_6 \quad O_7 \quad O_8$

If $\left| X - \frac{1}{8}\sum_{i=1}^{8} O_i \right| > T$

X is an outlier.

Outlier Removal (1)



Outlier Removal (2)

- Median Filtering
  - What is a median?
  - Rank order the samples from the smallest to the largest and pick the middle one
  - Example: 7, 1, 2, 3, 1, 5, 3
  - Sorting: 1, 1, 2, 3, 3, 5, 7
  - Choose the middle one -> 3
  - If we replace 7 with 1000, the median is still the same

Comparison of Mean and Median Filters

- **1**D case

Pratt's Book
Page 278

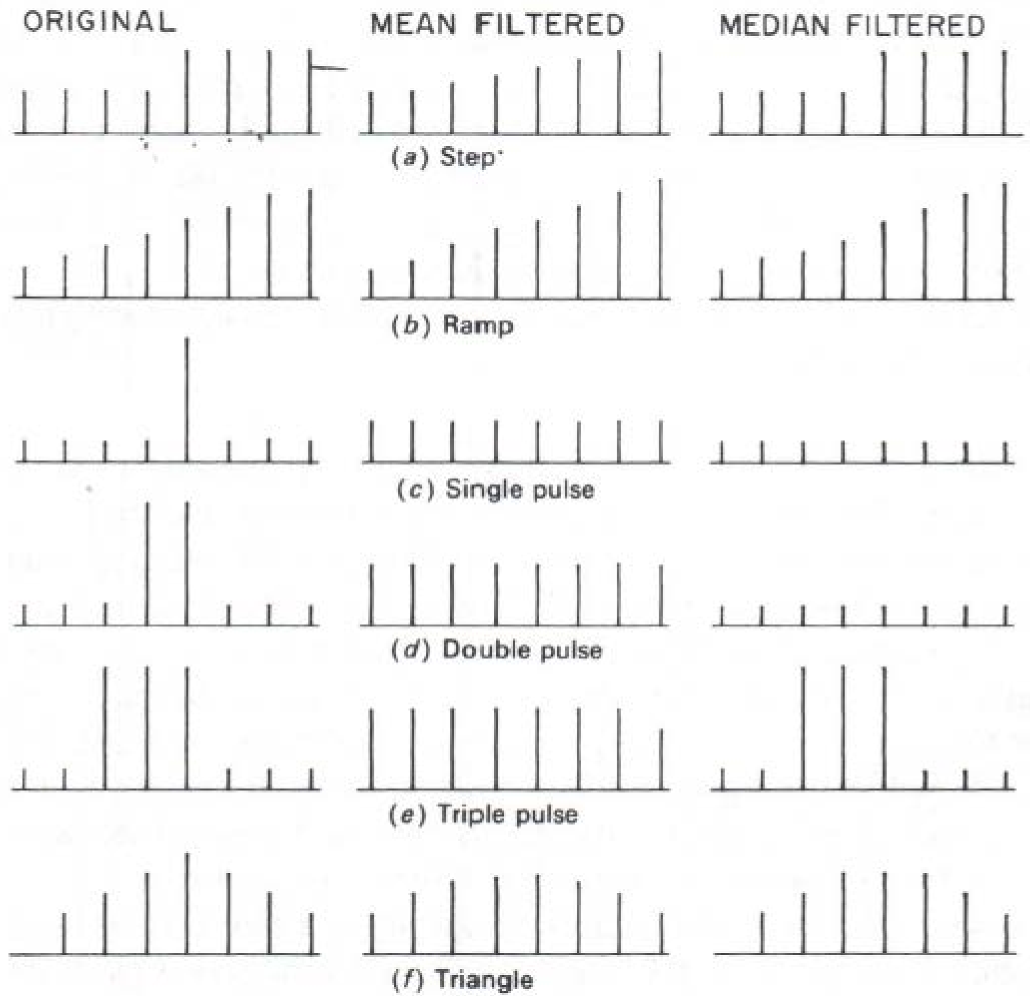


FIGURE 10.3-11. Median filtering on one-dimensional test signals.

Comparison of 2D Median Filters

Pratt's Book
Page 279

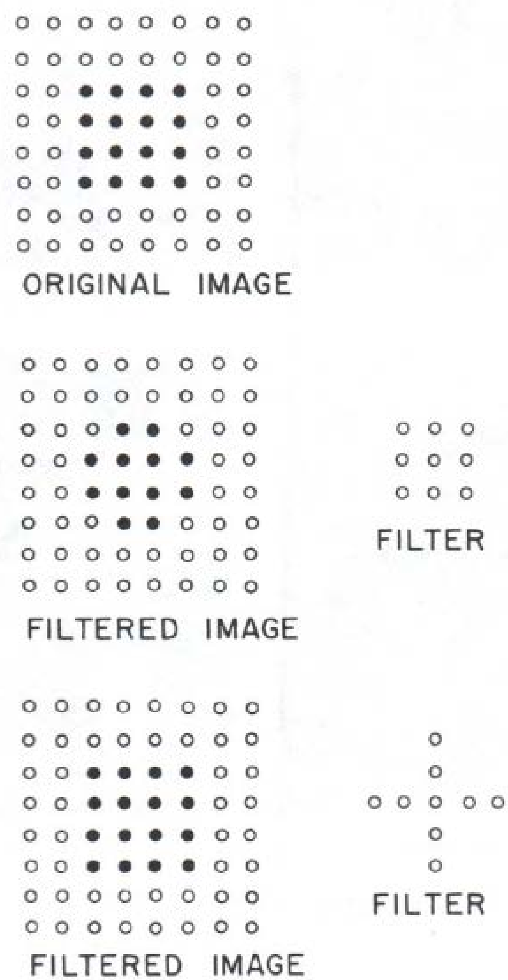
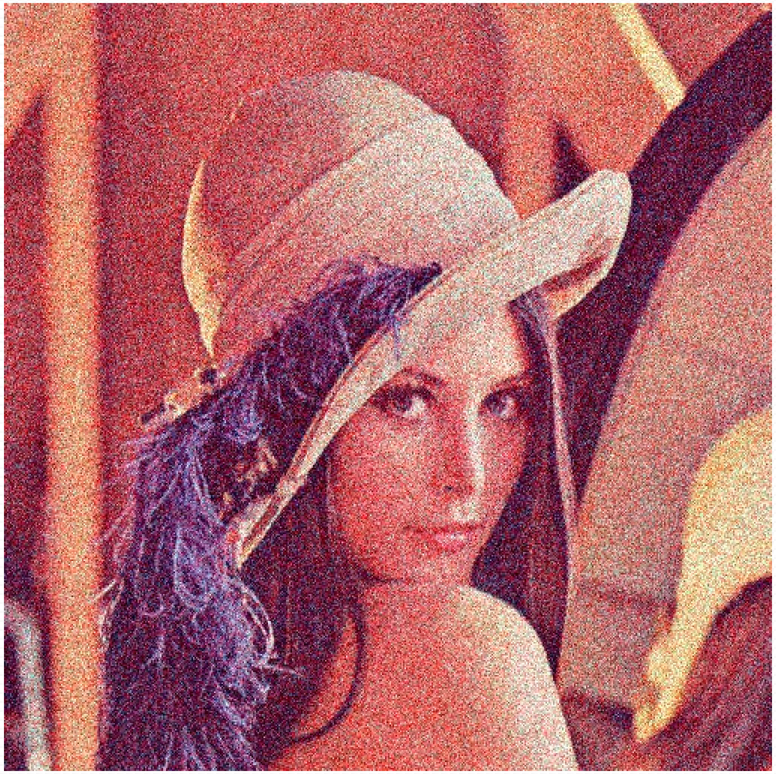


FIGURE 10.3-12. Median filtering on two-dimensional test signals.

Removal of AWGN

Image Denoising



Noisy image

ANL Denoised image

Denoising Overview

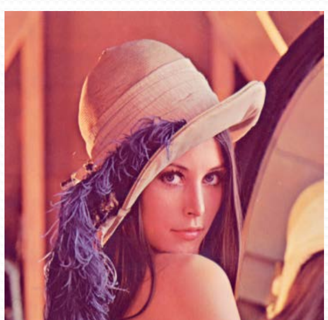
Brief History of Image Denoising

1970s

- USC
- Frequency domain techniques, direct inversion, or recursive Kalman filtering, etc

Lena Image

- Lena = Lena Söderberg
- Swedish model who posed nude for the November 1972 issue
- Signal and Image Processing Institute (SIPI)
- Reason:
  - Tired of usual test images
  - Good output dynamic range
  - Human face
- “Somebody happened have a recent issue of *Playboy*”



History of Image Denoising

1980s

•J-S. LEE, "*Digital image enhancement and noise filtering by use of local statistics*," IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. PAMI-2, pp. 165-168. Mar. 1980 (Cited by 759)

1990s

- Wavelet transforms
- Wiener filter
- Total variation minimization

History of Image Denoising

1998

- Bilateral Filter

Tomasi, Carlo, and Roberto Manduchi. "Bilateral filtering for gray and color images." Sixth international conference on computer vision (IEEE Cat. No. 98CH36271). IEEE, 1998.

2005

- Nonlocal mean (NLM) algorithm

Buades, Antoni, Bartomeu Coll, and J-M. Morel. "A non-local algorithm for image denoising." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 2. IEEE, 2005.

History of Image Denoising

2006

- Sparse Coding for Image Denoising

Elad, Michael, and Michal Aharon. "Image denoising via sparse and redundant representations over learned dictionaries." *IEEE Transactions on Image processing* 15.12 (2006): 3736-3745.

2007

- Block Matching 3D (BM3D)

Dabov, Kostadin, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. "Image denoising by sparse 3-D transform-domain collaborative filtering." IEEE Transactions on image processing 16, no. 8 (2007): 2080-2095.

History of Image Denoising

2012-

- Deep Learning for Image Denoising

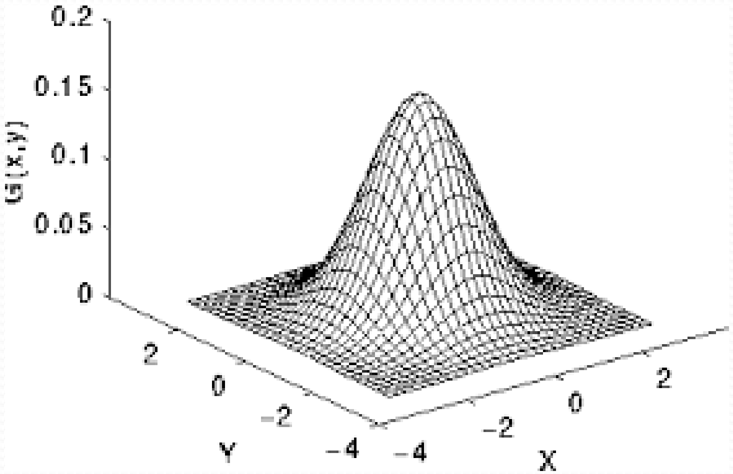
Xie, Junyuan, Linli Xu, and Enhong Chen. "Image denoising and inpainting with deep neural networks." Advances in neural information processing systems 25 (2012): 341-349.

Tian, Chunwei, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. "Deep learning on image denoising: An overview." Neural Networks (2020).

Gaussian Smoothing

Basic Denoising Idea

- Lowpass filters
  - Most image contents are low frequency
  - Noise components are high frequency
  - Use lowpass filters to suppress noise
    - Side effect: edges contain high frequency and thus are blurred
- Gaussian weighted lowpass filter



2D Gaussian Filters

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

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

$$\frac{1}{273} \times \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Bilateral Filtering

Denoising via Bilateral Filtering

$$Y(i,j) = \frac{\sum_{k,l} I(k,l)w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}$$

I (k,l): intensity level at pixel *(k,l)*

$$w(i,j,k,l) = exp\left(-\underbrace{\frac{(i-k)^2+(j-l)^2}{2\sigma_c^2}}_{\text{Gaussian weighting according to spatial distance}} - \underbrace{\frac{\|I(i,j)-I(k,l)\|^2}{2\sigma_s^2}}_{\text{Gaussian weighting according to intensity distance}}\right)$$

Nonlocal Means Algorithm

Denoising via Non-Local Mean

- Summary
  - Classical problem in image/video processing
  - X = S+N
    - X = noisy signal
    - S = original signal
    - N = Noise

  - Nonlocal means (NL-means) algorithm
    - A. Buades, B. Coll., and J. Morel, "*A non local algorithm for image denoising*," in Proc. Int. Conf. Computer Vision and Pattern Recognition (CVPR), vol. 2, 2005, pp. 60–65.
    - Sep 2009 ➔ cited by 6388 (01/24/2021)

Why Non-Local?

- $X_1 = S+N_1$
- $X_2 = S+N_2$
- .
- .
- $X_n = S+N_n$

- $(X_1+X_2+\ldots+X_n)/N = (S+S+\ldots+S)/N + (N_1+N_2+\ldots+N_n)/N$
  $= S+(N_1+N_2+\ldots+N_n)/N$

N is AWGN ➔ $(N_1+N_2+\ldots+N_n)/N \sim 0$


1 Sample
10 Samples
100 Samples

How to Get Similar Patches (1)?

If repeated patterns exist

Repeated Pattern
"High correlation"

How to Get Similar Patches (2)?

If repeated patterns do not exist -> self similarity

Self-similarity
"High correlation"

Nonlocal Means Algorithm

- For given noisy image $f = \{ f(i) \mid i \in \Omega \}$, the NL-means denoised value $\hat{f}(i)$ at pixel i is obtained by a weighted average of all pixels in its neighborhood

$$\hat{f}(i) = \frac{1}{C(i)} \sum_{j \in \Omega_S} w(i, j) f(j)$$

$C(i) = \sum_{j \in \Omega_S} w(i, j)$ is a normalization constant

$w(i, j)$ is determined by the similarity of the Gaussian neighborhood between pixels i and j

$$w(i, j) = \exp\left(-\frac{\left\| N_i - N_j \right\|_{2,a}^2}{h^2}\right)$$

Measure of Patch Distance

$$\left\|I(N_{i,j}) - I(N_{k,l})\right\|_{2,a}^{2} =$$

$$\sum_{n_1,n_2 \in N} G_a(n_1,n_2)\left(I(i-n_1, j-n_2) - I(k-n_1, l-n_2)\right)^2$$

$$G_a(n_1,n_2) = \frac{1}{\sqrt{2\pi}a}\exp\left(-\frac{n_1^2+n_2^2}{2a^2}\right)$$

BM3D:
Block Matching 3D

Flowchart

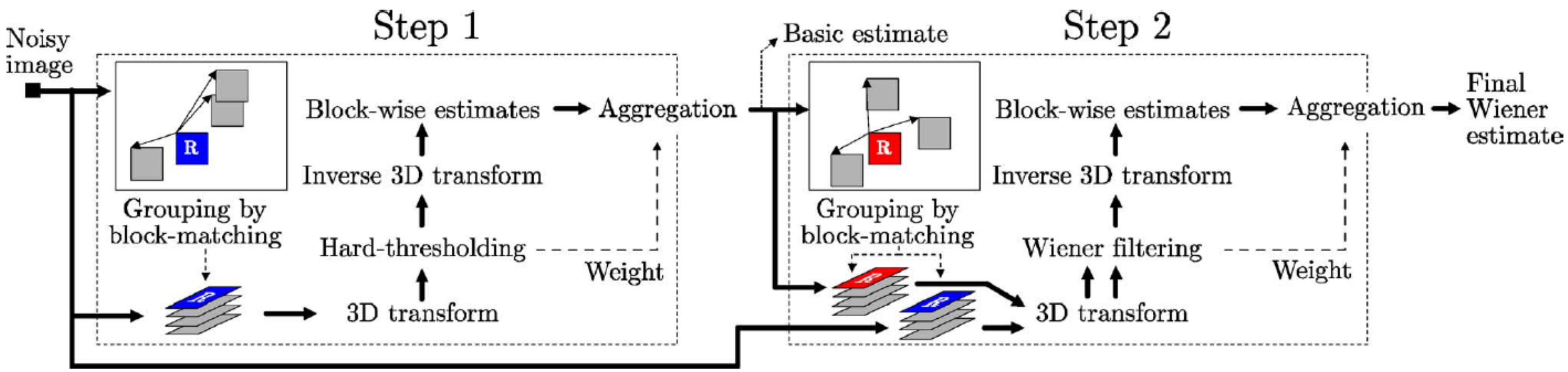


Fig. 3. Flowchart of the proposed image denoising algorithm. The operations surrounded by dashed lines are repeated for each processed block (marked with "R").

- find blocks that are similar to the reference one (block-matching) and stack them together to form a 3-D array (group);
- perform collaborative filtering of the group and return the obtained 2-D estimates of all grouped blocks to their original locations.

Block Matching: Example-1

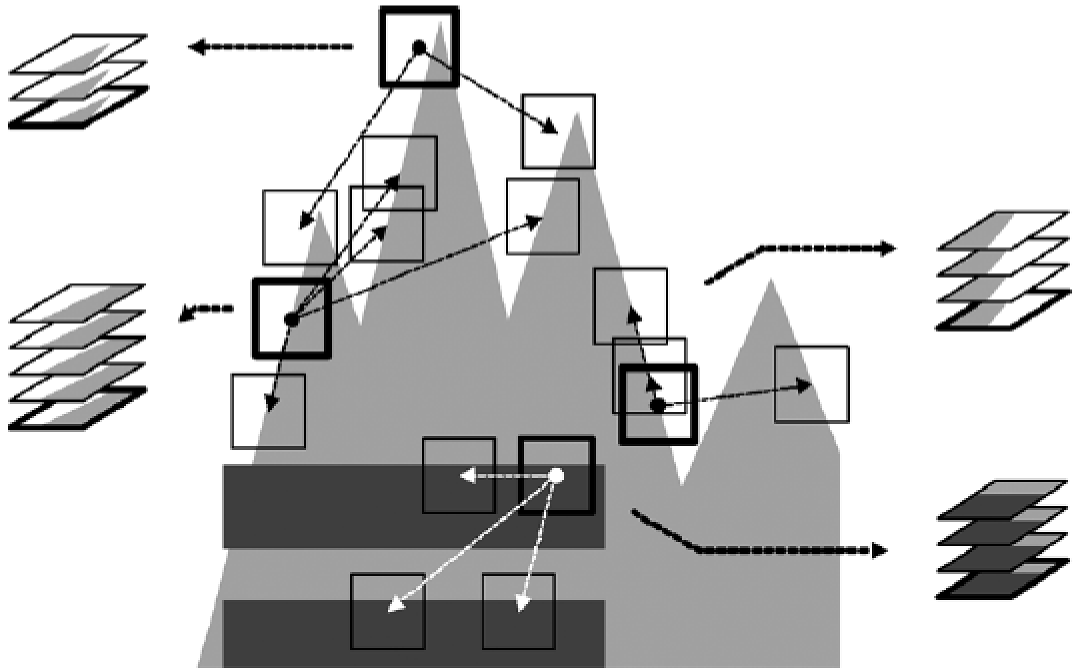


Fig. 2. Simple example of grouping in an artificial image, where for each reference block (with thick borders) there exist perfectly similar ones.

Block Matching: Example-2

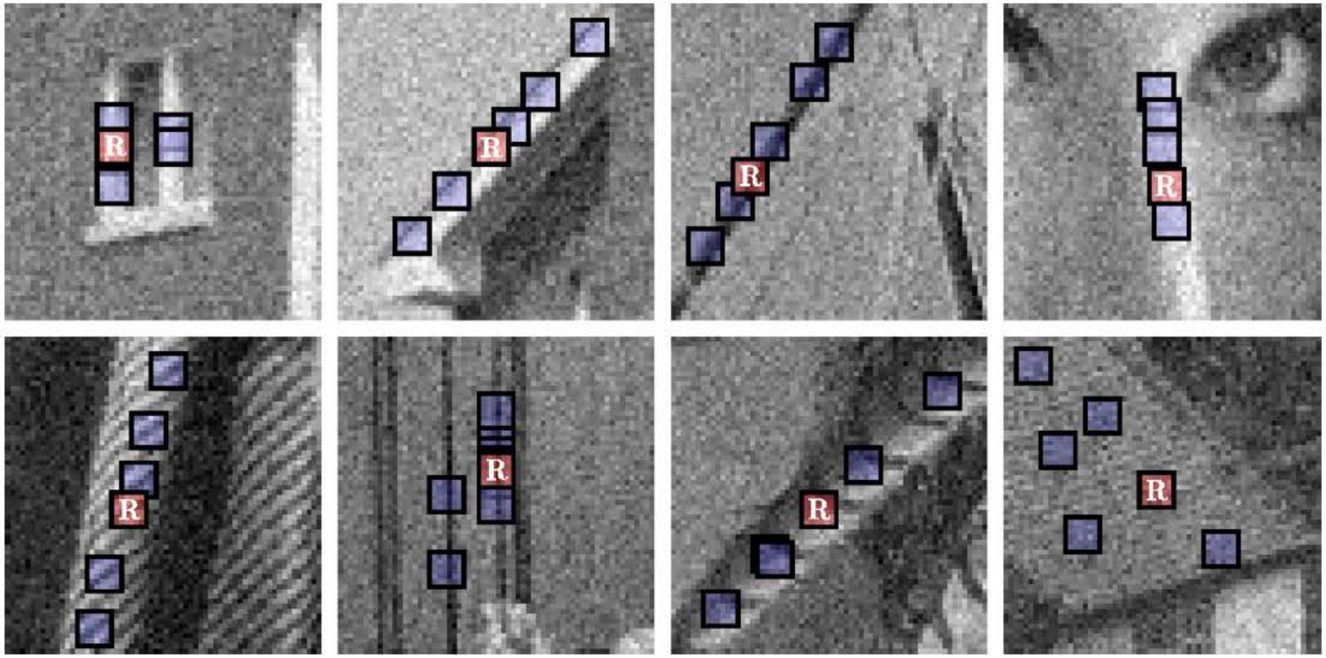


Fig. 1. Illustration of grouping blocks from noisy natural images corrupted by white Gaussian noise with standard deviation 15 and zero mean. Each fragment shows a reference block marked with "R" and a few of the blocks matched to it.

Difference between Steps 1 & 2

Collaborative hard-thresholding. Apply a 3-D transform to the formed group, attenuate the noise by hard-thresholding of the transform coefficients, invert the 3-D transform to produce estimates of all grouped blocks, and return the estimates of the blocks to their original positions.

Collaborative Wiener filtering. Apply a 3-D transform on both groups. Perform Wiener filtering on the noisy one using the energy spectrum of the basic estimate as the true (pilot) energy spectrum. Produce estimates of all grouped blocks by applying the inverse 3-D transform on the filtered coefficients and return the estimates of the blocks to their original positions.

Image Denoising Performance Evaluation

Subjective Evaluation

- Evaluated by humans

Objective Evaluation

- MSE (Mean Squared Errors)
- PSNR (Peak-Signal-to-Noise Ratio)

$$\text{PSNR (dB)} = 10\log_{10}\left(\frac{\text{Max}^2}{\text{MSE}}\right)$$

$$\text{where MSE} = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(Y(i,j) - X(i,j)\right)^2$$

X : Original Noise-free Image of size $N \times M$

Y : Filterd Image of size $N \times M$

Max: Maximum possible pixel intensity $= 255$

Adaptive Nonlocal Means Algorithm

- Proposed Algorithm
  - Experimental Results

Adaptive NL-means Algorithm

- Summarized idea
  - Adjust denoising parameter based on local content
  - Enhance the similarity matching process in order to fully exploit self-similarity existing within an image

- Proposed scheme
  1. Block classification
  2. Adaptive block matching

I. Block Classification

- Summarized idea
  - Determine block type of each local point so that different approach can be selectively applied to each block type
- Challenge
  - Noisy data ➔ difficult to classify
- Solution
  - Gradient vector
  - Singular Value Decomposition (SVD)
  - K-means clustering technique (K-means)

Simple Edge Detection (Canny)

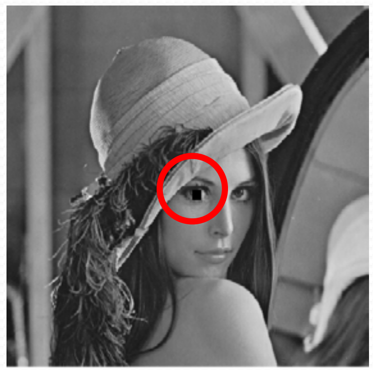
Block Classification Scheme

- For given noisy image, $f = \{ f(i) \mid i \in \Omega \}$
- Gradient vector: $G = \left[\nabla f(1)^T \; \nabla f(2)^T \cdots \nabla f(N)^T \right]^T ; \nabla f(i) = \left[\frac{\partial f(i)}{\partial x} \quad \frac{\partial f(i)}{\partial y} \right]^T$
- SVD: $G = USV^T$
  - U is an N x N orthogonal matrix
  - S is a diagonal 2 x 2 matrix contains singular values
  - V is an 2 x 2 orthogonal matrix which describes the dominant orientation of the gradient field

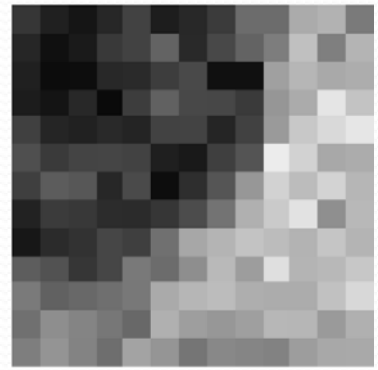
Block Classification Scheme

- SVD
  - Singular value ~ Eigenvalue ➔ energy in dominant direction
  - Classification feature
    - Smooth region: no dominant direction and small eigenvalues
    - Oriented edge/texture region: dominant direction and the corresponding eigenvalue is significantly larger than others
    - Noise has minimal effect ➔ noise does not have any preferred direction

Example Of Energy In Dominant Direction

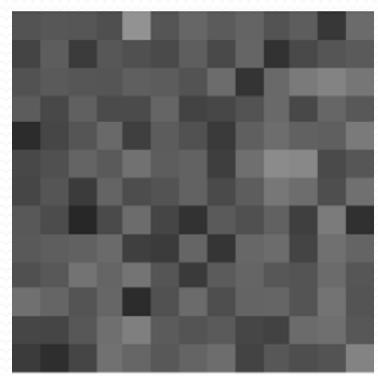


Texture/Edge region



E = 7.66

Smooth region



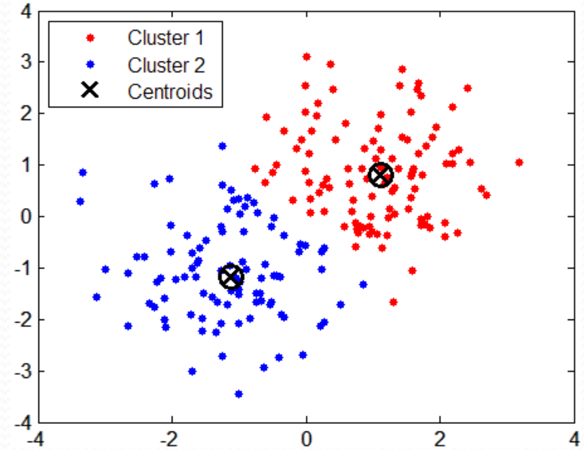
E = 2.73

Block Classification Scheme

- K-means
  - Adaptively classify the acquired data
  - K-means algorithm partitions $\{s(i) \mid i \in \Omega\}$ into K classes $C = \{c_1, c_2 \ldots, c_K\}$ while minimizing the within-cluster sum of squares

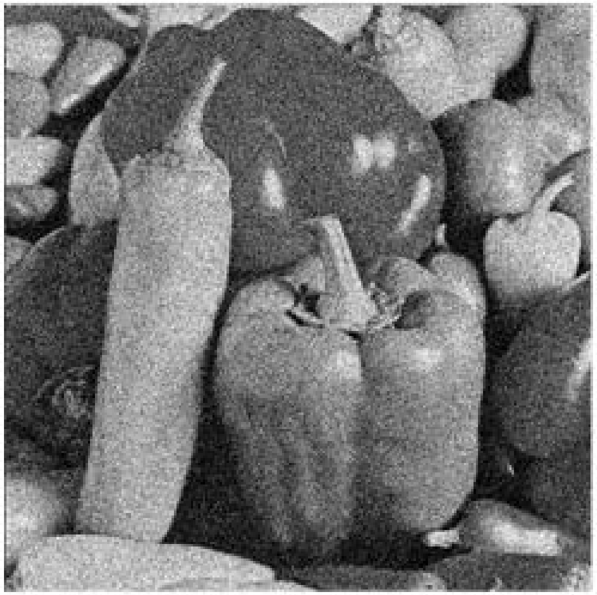
$$\arg\min_{C} \sum_{k=1}^{K} \sum_{s(i)\in c_k} \left| s(i) - \mu_k \right|^2$$

where μ_k is the mean of c_k

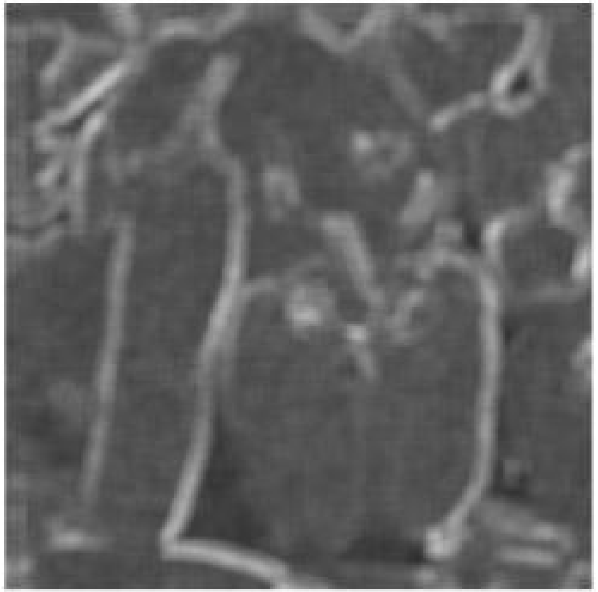


Sample: Block Classification

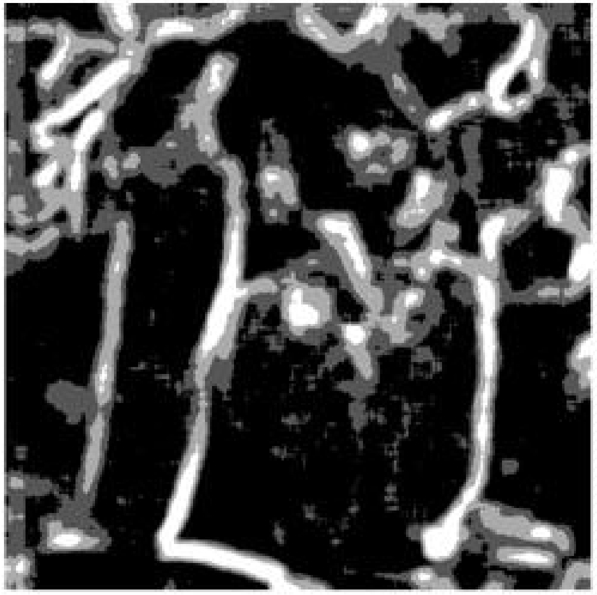
- Singular value decomposition / K-means clustering



Noisy image (σ=40)
PSNR = 16.48 dB



Energy in the dominant
directions (SVD)

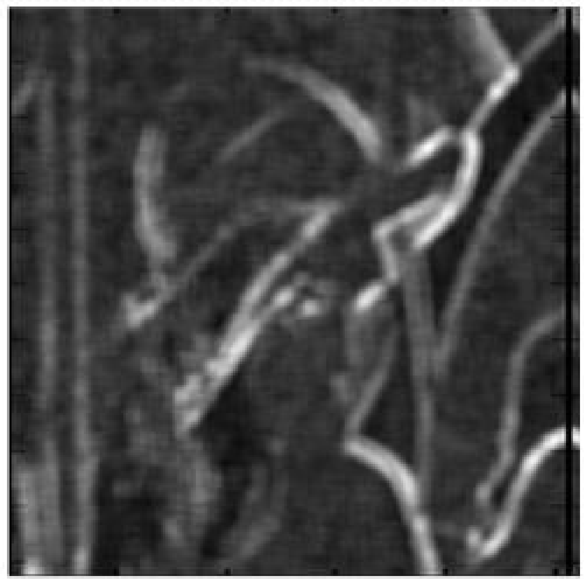


Classification Results (K-means)

Sample: Block Classification

- Singular value decomposition / K-means clustering

Noisy image (σ=40)
PSNR = 16.35 dB



Energy in the dominant directions (SVD)

Classification Results (K-means)

II. Adaptive Block Matching

- Summarized idea
  - Adjust denoising parameter based on block type and enhance the similarity matching process

- Solution
  - Adaptive block matching
    - Edge / Texture area ➔ Block transformation (Fractal-like matching)
    - Smooth area ➔ Adaptively adjusted window size

Block Transformation

- Summarized idea
  - Increase the number of matching candidate so that the better matching similarity can be determined
- Challenge
  - Matching candidate → Complexity
- Solutions
  - Smart block transformation → dominant orientation alignment
  - Selectively apply → Edge / Texture area

Block Transformation Scheme

Normal Scheme

- T1: Identity
- T2: Orthogonal reflection about mid-vertical axis of block
- T3: Orthogonal reflection about mid-horizontal axis of block
- T4: Orthogonal reflection about first diagonal (i = j) of block
- T5: Orthogonal reflection about second diagonal (i + j = B - 1) of block
- T6: Rotation around center of block, through +90
- T7: Rotation around center of block, through +180
- T8: Rotation around center of block, through -90
- T9: Luminance shift by constant [-20 - 20]
- T10: Contrast scaling [0.85 – 1.15]

Proposed scheme

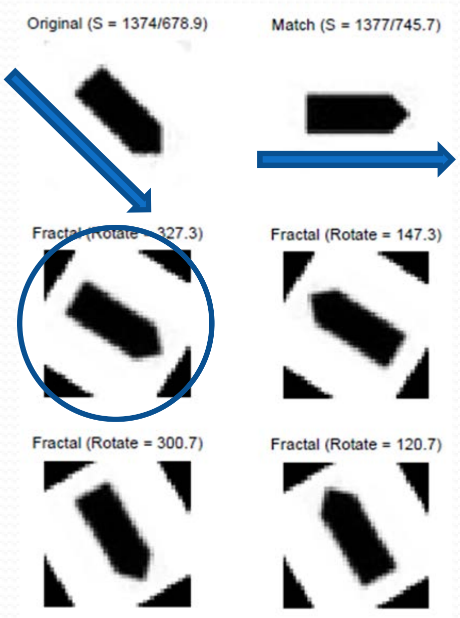
- SVD: $G = USV^T$
- V is an 2 x 2 orthogonal matrix which describes the dominant orientation of the gradient field
- Let $v_1 = [\upsilon_1 \ \upsilon_2]^T$ be the first column of V
- Dominant orientation

$$\theta = \arctan\left(\frac{\upsilon_1}{\upsilon_2}\right)$$

Dominant Orientation Alignment

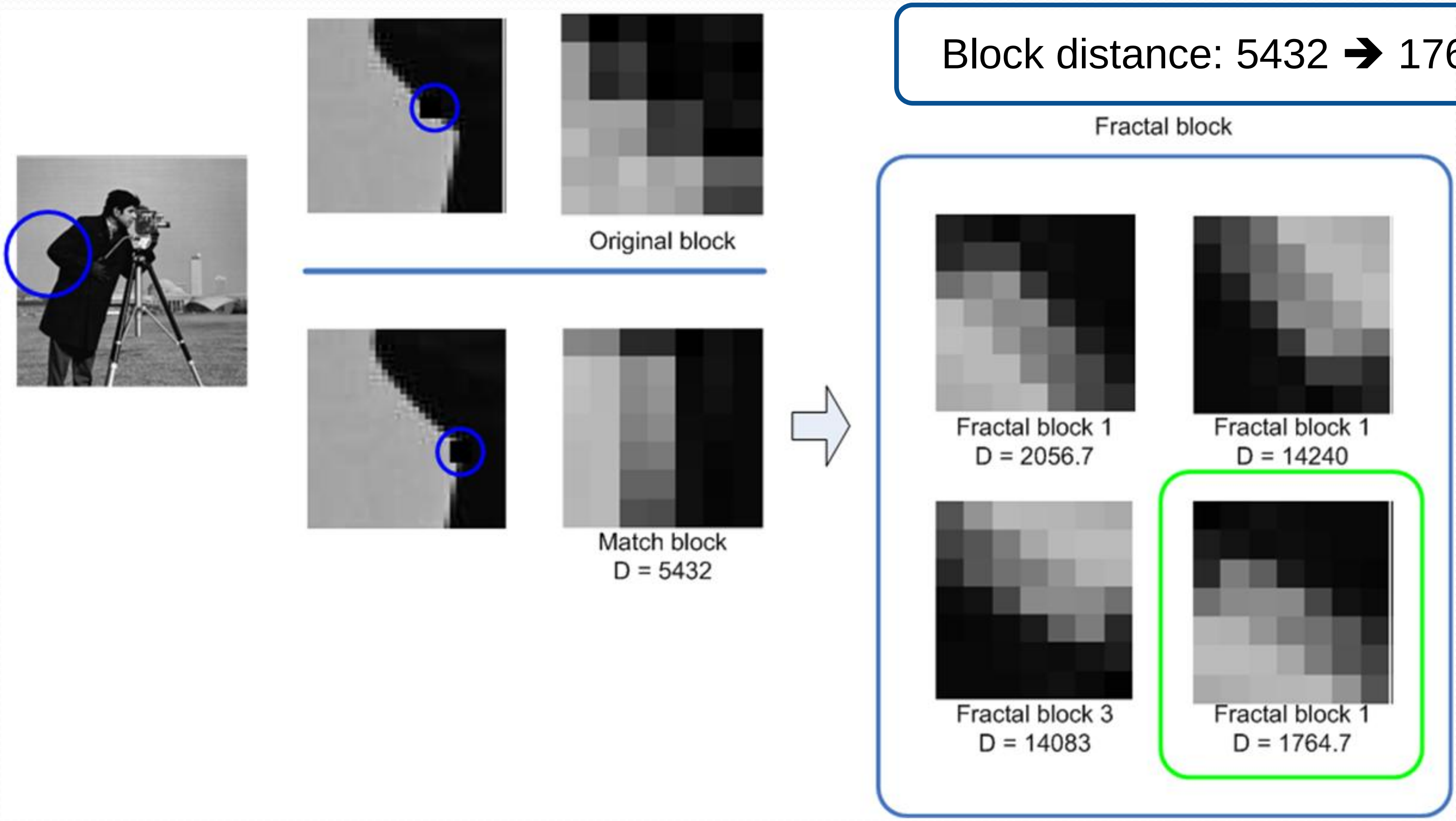
- Summarized idea
  - Rely on a set of rotated blocks that have their dominant orientation aligned well with that of the target block

- Solution
  - 4 fractal blocks ➔ Capture all possible gradient direction
  - Block rotation ➔ bilinear



Example: Dominant Orientation Alignment

Block distance: 5432 → 1764.7



Adaptive Matching Window

- Summarized idea
  - Adaptively adjust the size of matching window so that it can capture the local structure for better matching
  - Ex: Smooth + small matching window ➔ confuse noise as local structure/texture
- Solution
  - Adaptively choose the matching window size based on the block type
    - Strong edge/texture region ➔ small window (7x7)
    - Smooth region ➔ large window (25x25)
    - other regions ➔ medium window (13x13)

Adaptive Nonlocal Means Algorithm

- Proposed Algorithm
- Experimental Results

Denoising Benchmark

- Denoising algorithm
  - Mean filter (MF)
  - Gaussian filter (GF)
  - Partial differential equation (PDE)
  - Total variation minimization (TV)
  - Nonlocal-means (NL)

- Proposed technique
  - Adaptive nonlocal-means (ANL)

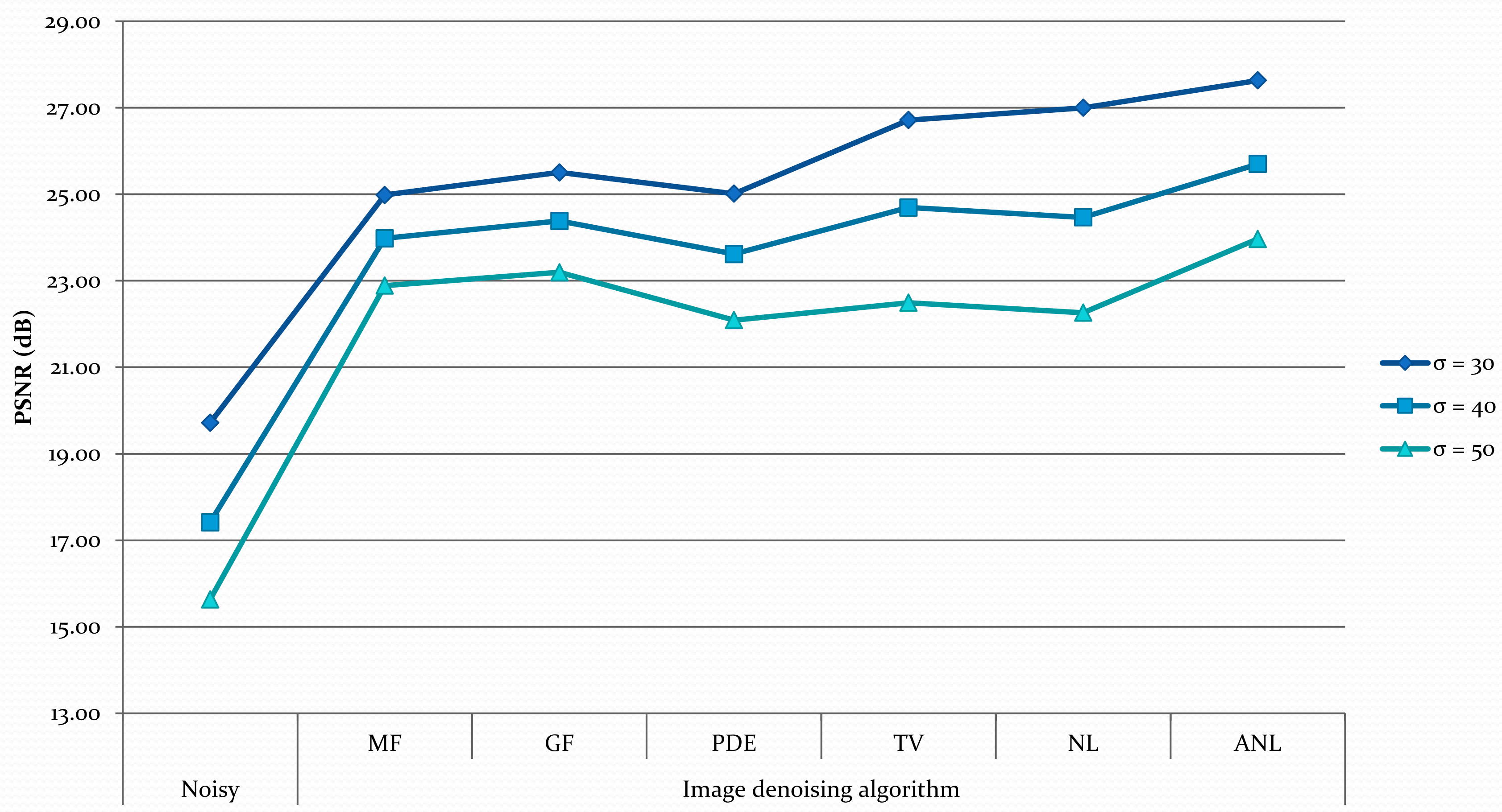
Experiment Setting

- Parameters
  - 7 representative test images
  - Additive white Gaussian noise (AWGN) with zero mean and standard deviation σ = 20, 30 and 40

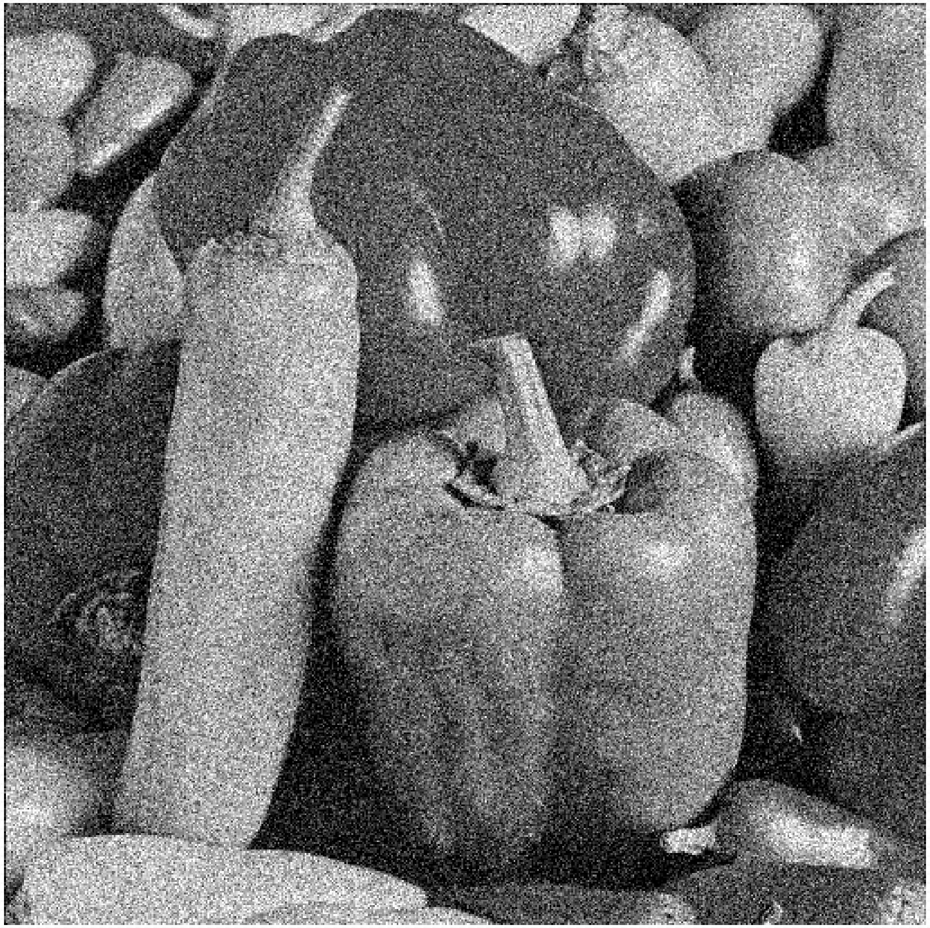
Experiment Results (NL vs. ANL)

Image	Average PSNR (dB)								
	Sigma = 20			Sigma = 30			Sigma = 40		
	NL	ANL	Δ	NL	ANL	Δ	NL	ANL	Δ
Lena	31.02	31.98	0.96	27.50	30.04	2.54	24.37	28.27	3.90
Zelda	31.85	32.83	0.98	28.18	30.72	2.55	25.06	28.76	3.70
Peppers	30.93	31.59	0.65	27.50	29.79	2.29	24.40	28.00	3.60
airplain	30.52	30.93	0.41	27.20	29.05	1.85	24.34	27.41	3.07
Barbara	29.85	30.30	0.45	26.65	28.41	1.76	23.89	26.74	2.85
Elaine	30.40	30.82	0.42	27.30	29.58	2.28	24.32	28.10	3.78
Girlface	31.75	32.29	0.54	28.12	29.98	1.86	25.06	27.92	2.86
Average	30.90	31.53	**0.63**	27.49	29.65	**2.16**	24.49	27.89	**3.39**

Experiment Results – Performance Comparison



Noisy Image – σ=40

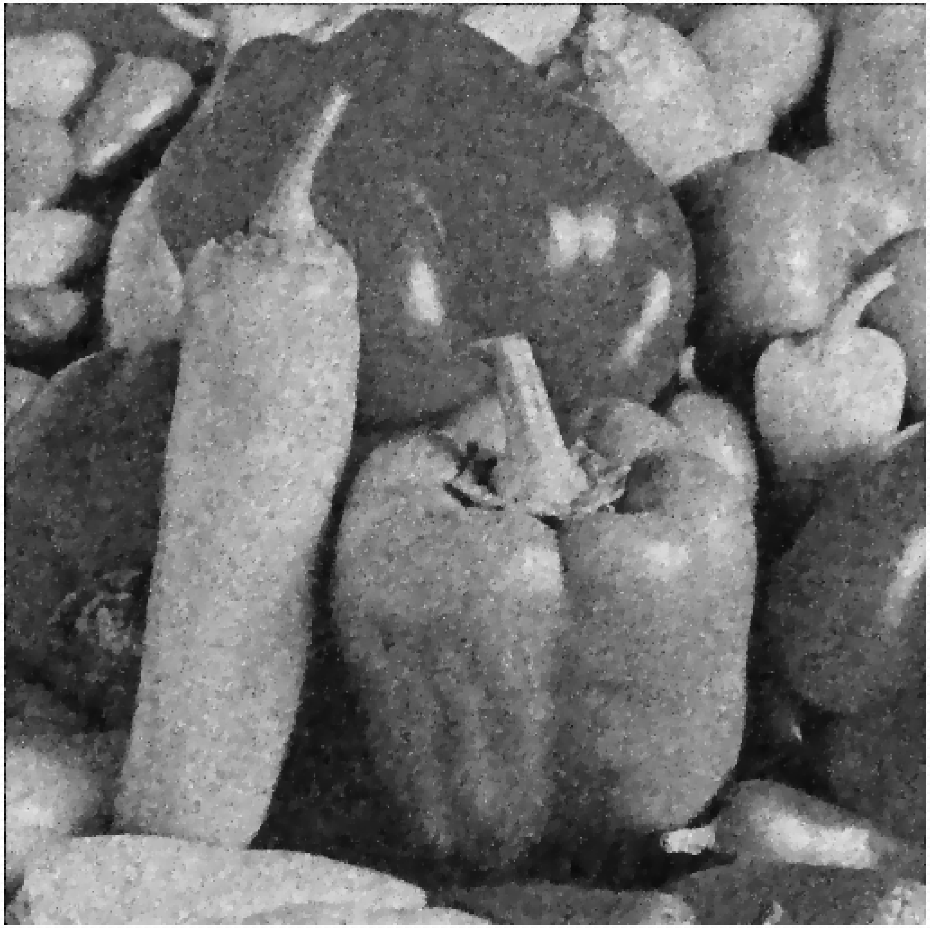


Mean Filter

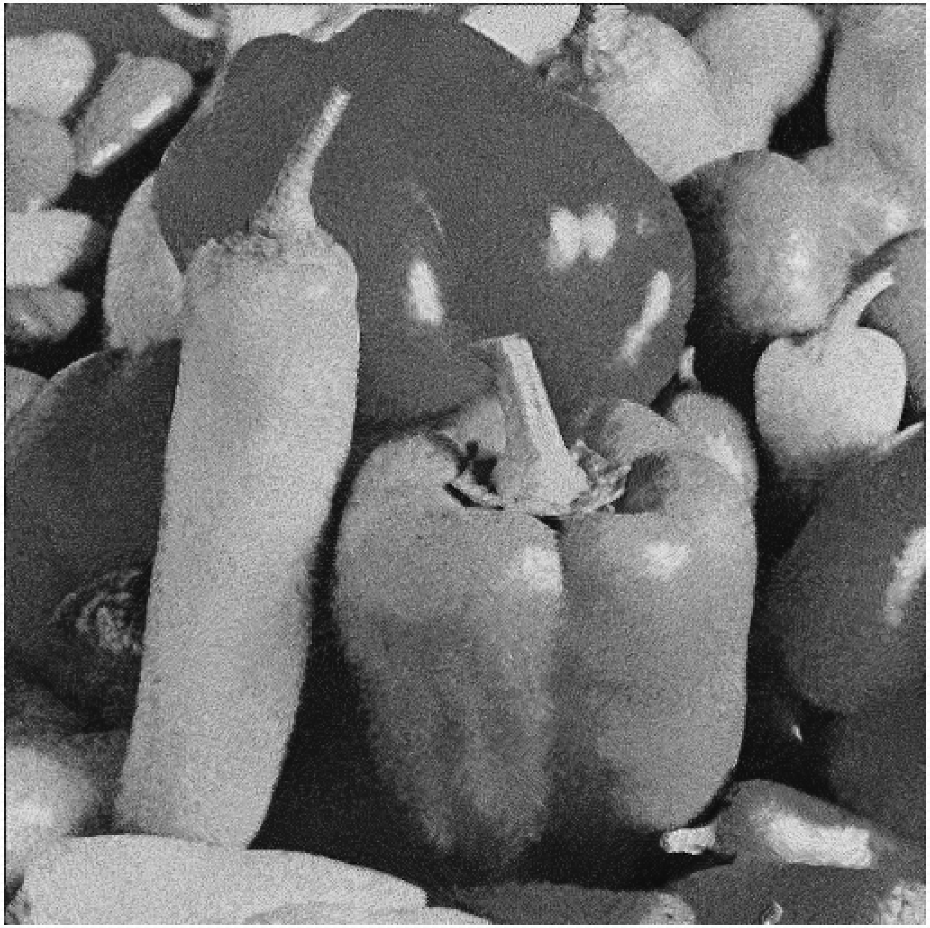
Gaussian Filter

Partial Differential Equation

Total Variation Minimization



Nonlocal Means



Adaptive Nonlocal Means

Other Example: Girlface

Noisy image (σ=40)

NL denoised image

ANL denoised image

Other Example: Zelda

Noisy image (σ=40)

NL denoised image

ANL denoised image

Thank you