Gene Cheung

National Institute of Informatics

6th July, 2015

Depth Image Coding & Processing Part 3: Depth Image Processing

.

Outline

- Depth Image Denoising
  - Graph Sparsity Prior
  - Graph-signal Smoothness Prior
- Bit-depth Enhancement

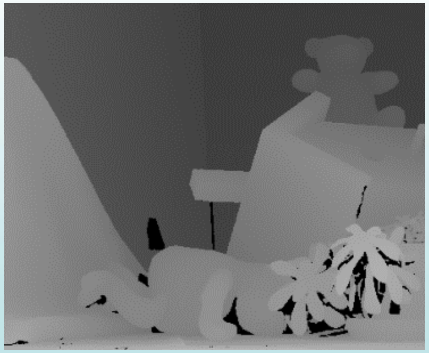
Outline

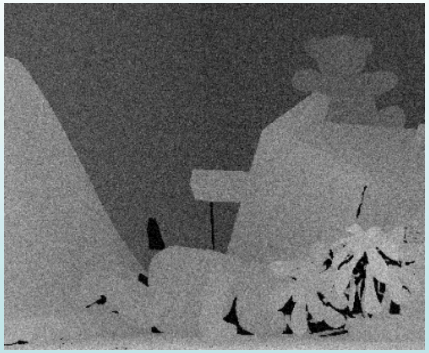
- Depth Image Denoising
  - Graph Sparsity Prior
  - Graph-signal Smoothness Prior
- Bit-depth Enhancement

Introduction to PWS Image Denoising

- Limitations of current sensing technologies

  - acquired PWS images are often corrupted by non-negligible acquisition noise.





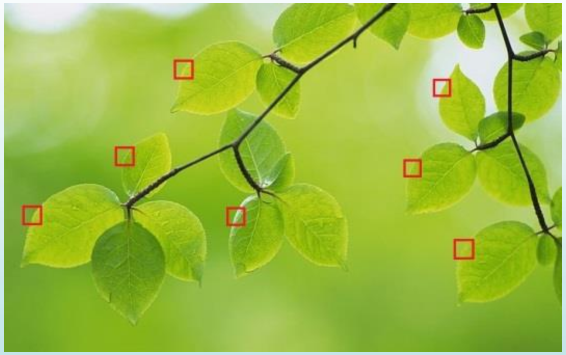
- Denoising is an inverse imaging problem.

$$\text{observation} \longrightarrow y = \overset{\text{desired signal}}{x} + v \longleftarrow \text{noise}$$

- ***Signal prior is key to inverse imaging problems!***
  - Depth images are PWS, self-similar.

Existing Image Denoising Methods

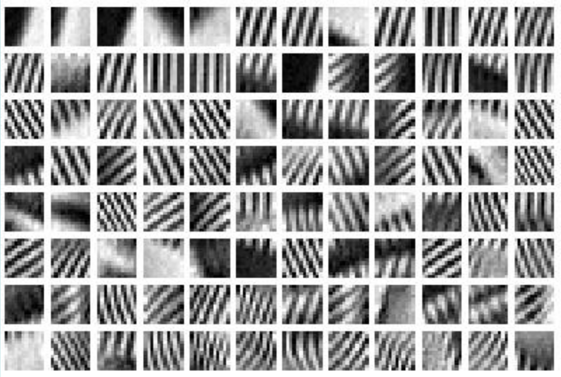
- Local methods (e.g., bilateral filtering)



- Nonlocal image denoising

Buades et al, "A non-local algorithm for image denoising," *CVPR 2005*

- Assumption: nonlocal self-similarity



- Dictionary learning based

Elad et al, "Image denoising via sparse and redundant representation over learned

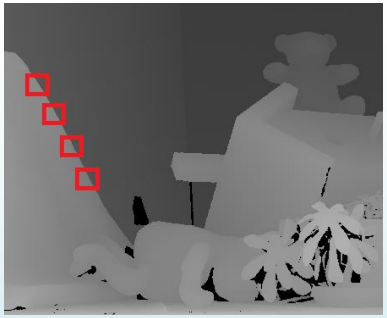
dictionaries," *TIP 2006.*

- represent a signal by the linear combination of a few atoms out of a dictionary

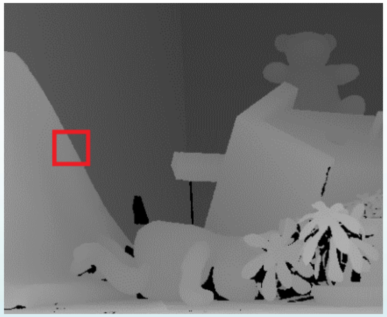
Other related works

- Huhle et al, “Robust non-local denoising of colored depth data,” CVPR Workshop 2008
- Tallon et al, “Upsampling and denoising of depth maps via joint segmentation,” EUSIPCO 2012

Key Idea in Non-local GFT



Nonlocal self-similarity



Local Piecewise Smoothness

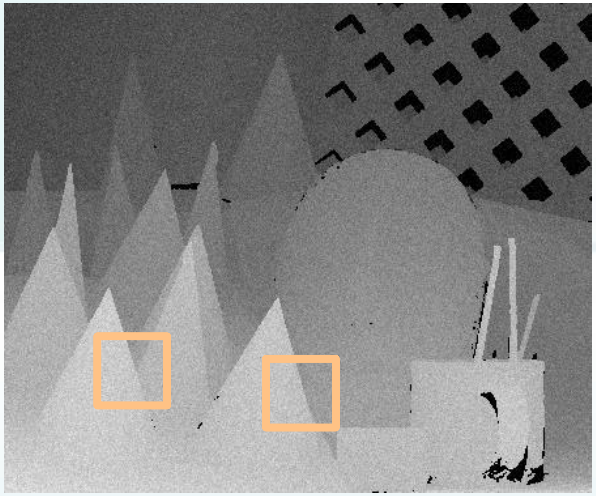
unify in GFT domain

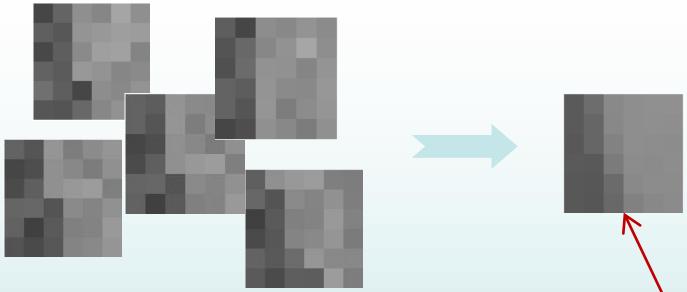
Challenges Our method

1. Adapt to nonlocal statistics --- adapt to nonlocal statistics via nonlocal self-similarity
2. Characterize PWS --- characterize PWS via GFT representation

+ learn GFT dictionary efficiently

NL-GFT Algorithm



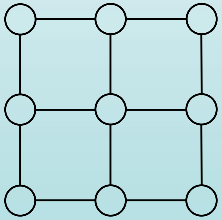


common GFT from avg. patch

observation *i*

$$\min_{U,\alpha}\sum_{i=1}^{N}\left\|y_i - U\alpha_i\right\|_2^2 + \mu\sum_{i=1}^{N}\left\|\alpha_i\right\|_0$$

code vector for observation *i*



$$W = [w_{ij}],$$

$$w_{ij} = e^{\frac{-\|y_i - y_j\|^2}{\sigma_w^2}}$$

$$\mathcal{L} = D - W$$

$$\mathcal{L}\mathbf{U} = \mathbf{U}\Lambda$$

Algorithm:

1. Identify similar patches, compute avg patch. (**self-similarity**)
2. Given avg patch, use Gaussian kernel to compute weights between adjacent pixels.
3. Compute graph Fourier transform (GFT).
4. Given GFT, soft thresholding on transform coeff. for sparse representation.

[1] W. Hu, X. Li, G. Cheung, O. Au, "**Depth Map Denoising using Graph-based Transform and Group Sparsity**," *IEEE International Workshop on Multimedia Signal Processing*, Pula (Sardinia), Italy, October, 2013. (**Top 10% paper award**.)

Justification of Sparsity Prior

- GFT domain sparsity prior in objective function:

$$\min_{\Phi, x_i} \sum_{i=1}^{K} \left\| y_i - x_i \right\|_2^2 + \lambda \sum_{i=1}^{K} \left\| \Phi x_i \right\|_0$$

- **"Argument"**:
  - GFT approximates KLT if statistical model is GMRF and each graph weight captures correlation of 2 connected pixels [2, 3].
  - Underlying "causes" of PWS signals are few; PWS signal can be sparsely represented in GFT domain [4, 5].

[2] C. Zhang and D. Florencio, "**Anaylzing the optimality of predictive transform coding using graph-based models**," in *IEEE Signal Processing Letters*, vol. 20, NO. 1, January 2013, pp. 106–109.

[3] W. Hu, G. Cheung, A. Ortega, O. Au, "**Multi-resolution Graph Fourier Transform for Compression of Piecewise Smooth Images**," *IEEE Transactions on Image Processing*, January 2015.

[4] G. Shen, W.-S. Kim, S.K. Narang, A. Ortega, J. Lee, and H. Wey, "**Edge-adaptive transforms for efficient depth map coding**," in *IEEE Picture Coding Symposium*, Nagoya, Japan, December 2010.

[5] W. Hu, G. Cheung, X. Li, O. Au, "**Depth Map Compression using Multi-resolution Graph-based Transform for Depth-image-based Rendering**," *IEEE International Conference on Image Processing*, Orlando, FL, September 2012.

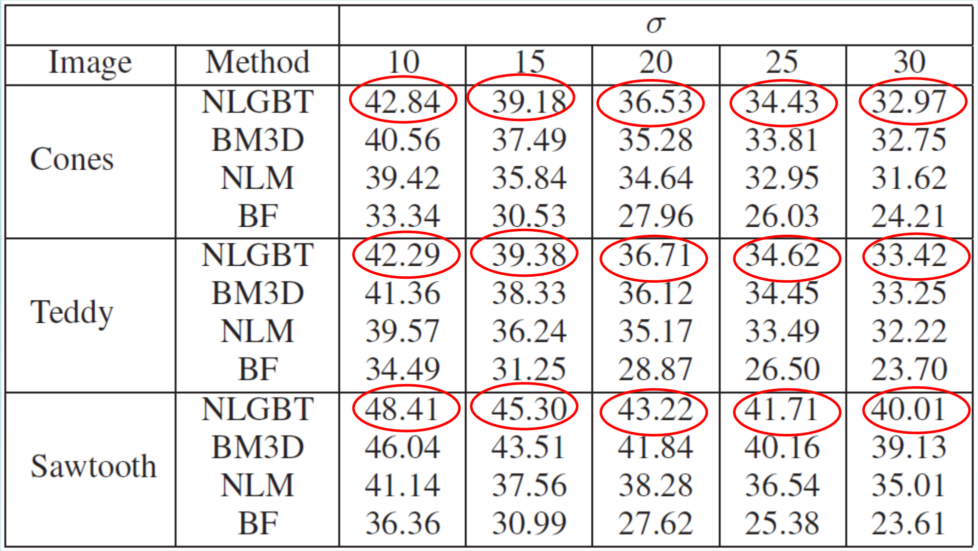
Experimental Results (1)

- Setup:

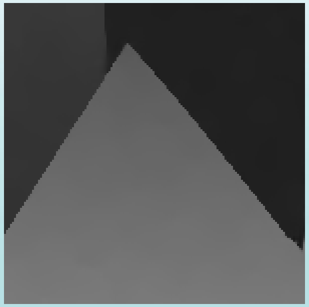
  - Test Middleburry depth maps: *Cones*, *Teddy*, *Sawtooth*
  - Add Additive White Gaussian Noise
  - Compare against Bilateral Filtering (BF), Non-Local Means Denoising (NLM) and Block-Matching 3D (BM3D)

- Results

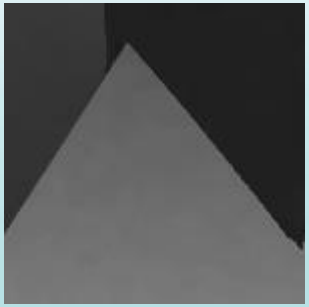
  – Up to 2.28dB improvement over BM3D.



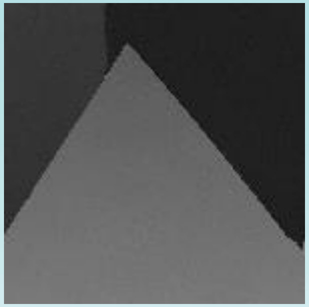
		σ				
Image	Method	10	15	20	25	30
Cones	NLGBT	42.84	39.18	36.53	34.43	32.97
	BM3D	40.56	37.49	35.28	33.81	32.75
	NLM	39.42	35.84	34.64	32.95	31.62
	BF	33.34	30.53	27.96	26.03	24.21
Teddy	NLGBT	42.29	39.38	36.71	34.62	33.42
	BM3D	41.36	38.33	36.12	34.45	33.25
	NLM	39.57	36.24	35.17	33.49	32.22
	BF	34.49	31.25	28.87	26.50	23.70
Sawtooth	NLGBT	48.41	45.30	43.22	41.71	40.01
	BM3D	46.04	43.51	41.84	40.16	39.13
	NLM	41.14	37.56	38.28	36.54	35.01
	BF	36.36	30.99	27.62	25.38	23.61



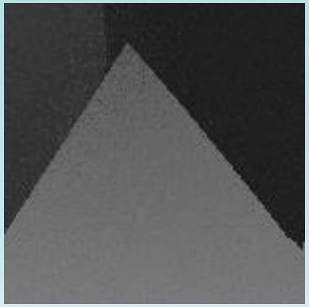
NLGFT



BM3D



NLM



BF

Experimental Results (2)

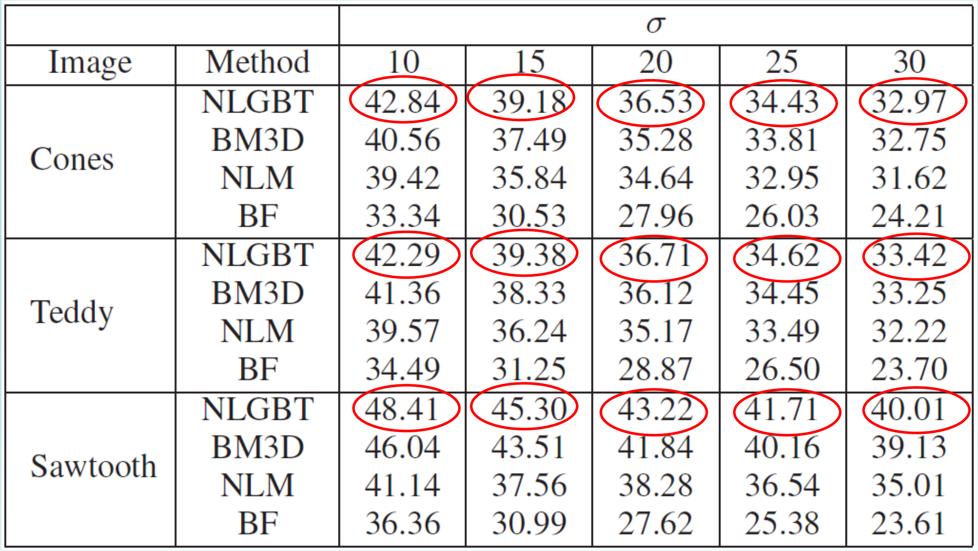
- Setup:

  - Test Middlebury depth maps: *Cones*, *Teddy, Sawtooth*

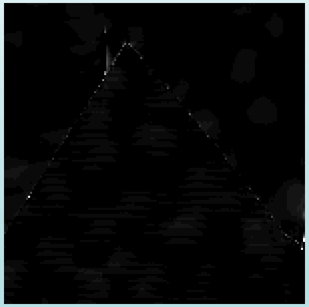
  - Add Additive White Gaussian Noise

  - Compare agaist Bilateral Filtering (BF), Non-Local Means Denoising (NLM) and Block-Matching 3D (BM3D)

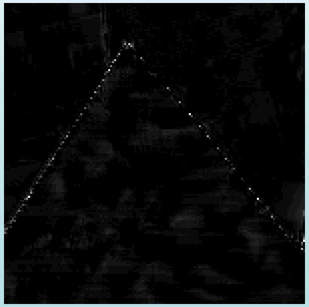
- Results
  - Up to 2.28dB improvement over BM3D.



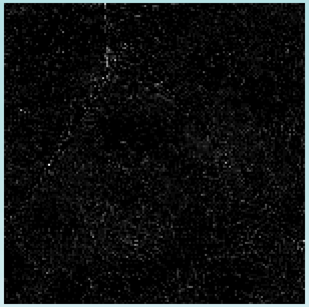
		σ				
Image	Method	10	15	20	25	30
Cones	NLGBT	42.84	39.18	36.53	34.43	32.97
	BM3D	40.56	37.49	35.28	33.81	32.75
	NLM	39.42	35.84	34.64	32.95	31.62
	BF	33.34	30.53	27.96	26.03	24.21
Teddy	NLGBT	42.29	39.38	36.71	34.62	33.42
	BM3D	41.36	38.33	36.12	34.45	33.25
	NLM	39.57	36.24	35.17	33.49	32.22
	BF	34.49	31.25	28.87	26.50	23.70
Sawtooth	NLGBT	48.41	45.30	43.22	41.71	40.01
	BM3D	46.04	43.51	41.84	40.16	39.13
	NLM	41.14	37.56	38.28	36.54	35.01
	BF	36.36	30.99	27.62	25.38	23.61



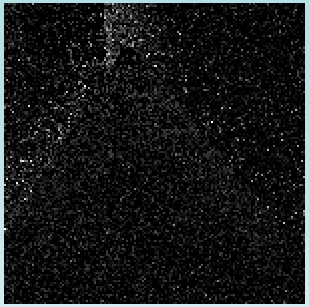
NLGFT



BM3D



NLM



BF

Outline

- Depth Image Denoising
  - Graph Sparsity Prior
  - Graph-signal Smoothness Prior
- Bit-depth Enhancement

Motivation (I)

- Image denoising—a basic restoration problem:

observation noise

$$\mathbf{y} = \mathbf{x} + \mathbf{e}$$

desired signal

- It is under-determined, needs image priors for regularization:

fidelity term

$$\min_{\mathbf{x}} \ \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \ \mathrm{prior}(\mathbf{x})$$

prior term

- Graph Laplacian regularizer: should be small for target patch $\mathbf{x}$

$$S_{\mathrm{G}}(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{L}\mathbf{x} \qquad \mathbf{L} = \mathbf{D} - \mathbf{A}$$

graph Laplacian matrix

- Many works use Gaussian kernel to compute graph weights [1, 6]:

$$w_{ij} = \exp\left(\frac{-dist(i, j)^2}{\sigma^2} \right)$$

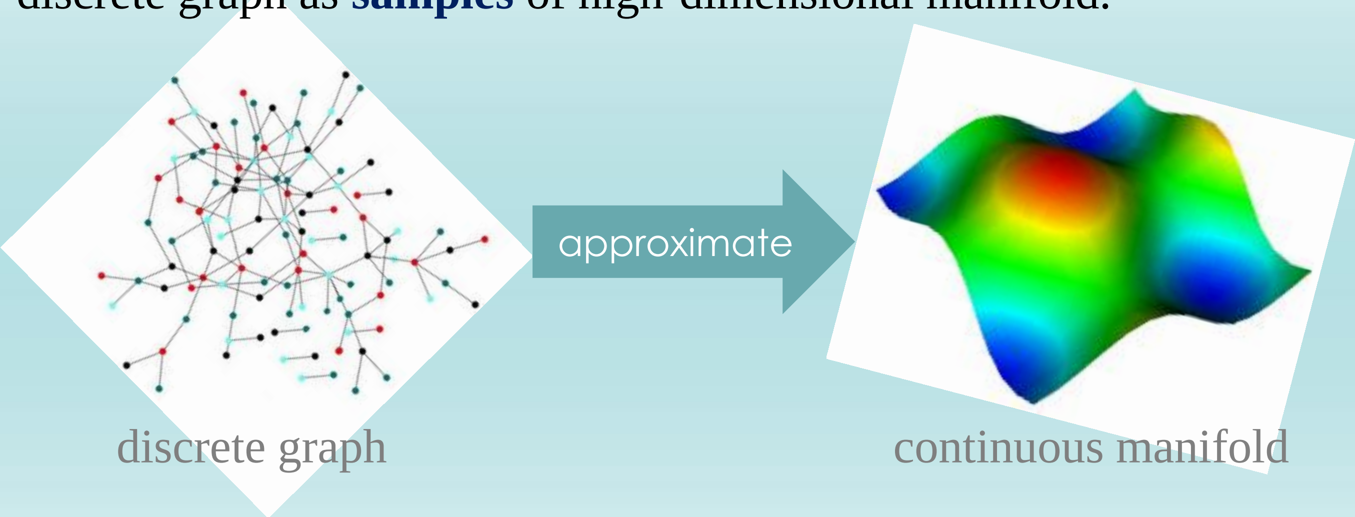
$\mathrm{dist}(i, j)$ is some distance metric between pixels *i* and *j*

[6] D. Shuman et al., "**The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains**," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

Motivation (II)

$$w_{ij} = \exp\left(\frac{-dist(i,j)^2}{\sigma^2}\right)$$

- However…
  a. Why is $S_{\mathrm{G}}(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{L}\mathbf{x}$ a good prior?
  b. Why using Gaussian kernel for edge weights?
  c. How to design a discriminant $\mathbf{x}^{\mathrm{T}}\mathbf{L}\mathbf{x}$ for restoration?
- We answer these basic questions by viewing:
  - discrete graph as **samples** of high-dimensional manifold.



discrete graph

continuous manifold

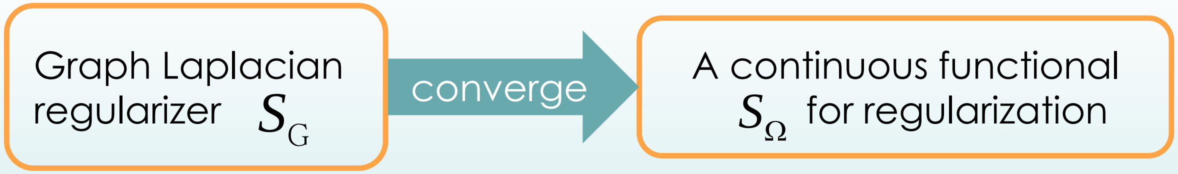
[7] Jiahao Pang, Gene Cheung, Antonio Ortega, Oscar C. Au, "**Optimal Graph Laplacian Regularization for Natural Image Denoising**," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Brisbane, Australia, April, 2015.

[8] Jiahao Pang, Gene Cheung, Wei Hu, Oscar C. Au, "**Redefining Self-Similarity in Natural Images for Denoising Using Graph Signal Gradient**," *APSIPA ASC*, Siem Reap, Cambodia, December, 2014.

Our Contributions

$$w_{ij} = \exp\left(\frac{-dist\left(i, j\right)^2}{\sigma^2}\right)$$

1. Using Gaussian kernel to compute graph weights, $S_{\mathrm{G}}(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{L}\mathbf{x}$ converges to a continuous functional S_{Ω} .

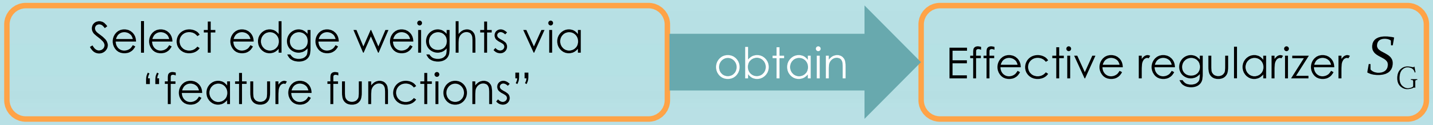


Graph Laplacian regularizer S_{G}

converge

A continuous functional S_{Ω} for regularization

2. Analysis of functional S_{Ω} provides understanding of how signals are being discriminated and to what extent; careful graph construction leads to ***discriminant*** signal prior.



Select edge weights via “feature functions”

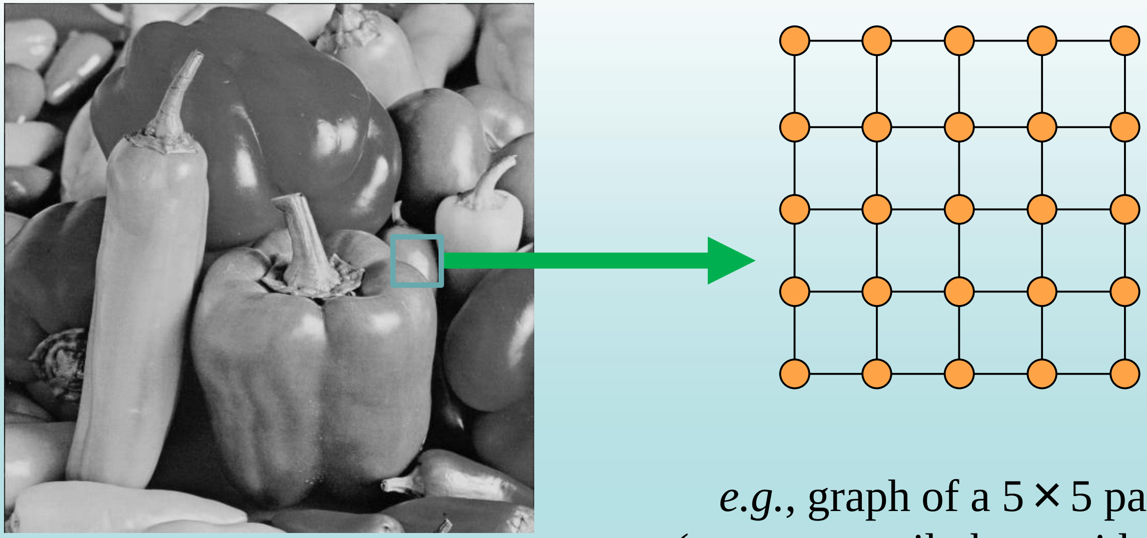
obtain

Effective regularizer S_{G}

3. We derive the optimal graph Laplacian regularizer for denoising, which is discriminant for small noise and robust when very noisy.

Graph-Based Image Processing

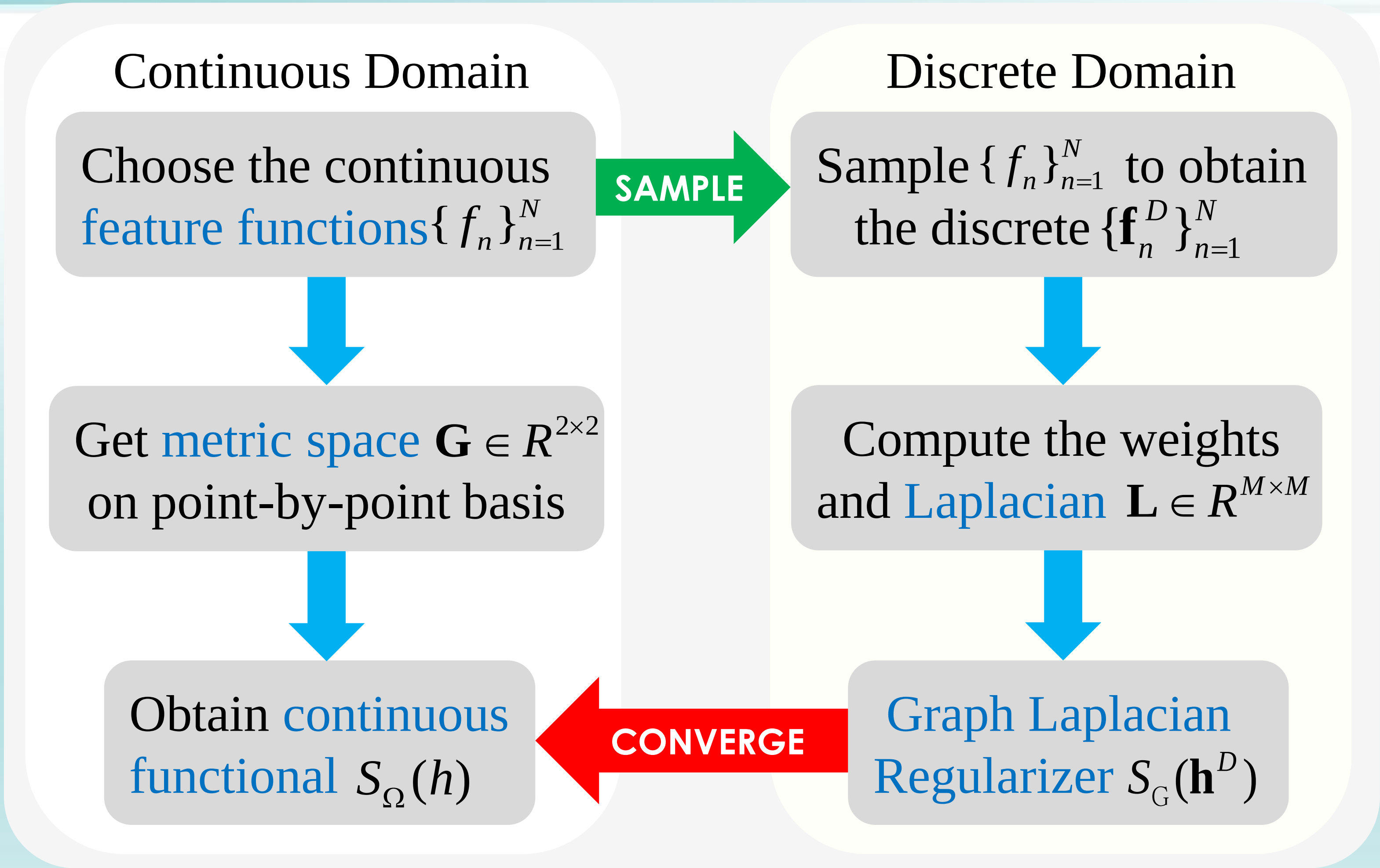
- Graph for image restoration
  - Each pixel corresponds to a vertex in a graph (denote # of pixels as M).



e.g., graph of a 5×5 patch,
(not necessarily be a grid graph)

  - Regard the image as a signal defined on a weighted graph.
  - With proper graph configuration, construct filter for image (graph signal) using prior knowledge (i.e., smooth on the graph).

Road Map



Continuous Domain

Choose the continuous feature functions $\{f_n\}_{n=1}^N$

Get metric space $\mathbf{G} \in R^{2\times 2}$ on point-by-point basis

Obtain continuous functional $S_\Omega(h)$

Discrete Domain

Sample $\{f_n\}_{n=1}^N$ to obtain the discrete $\{\mathbf{f}_n^D\}_{n=1}^N$

Compute the weights and Laplacian $\mathbf{L} \in R^{M\times M}$

Graph Laplacian Regularizer $S_G(\mathbf{h}^D)$

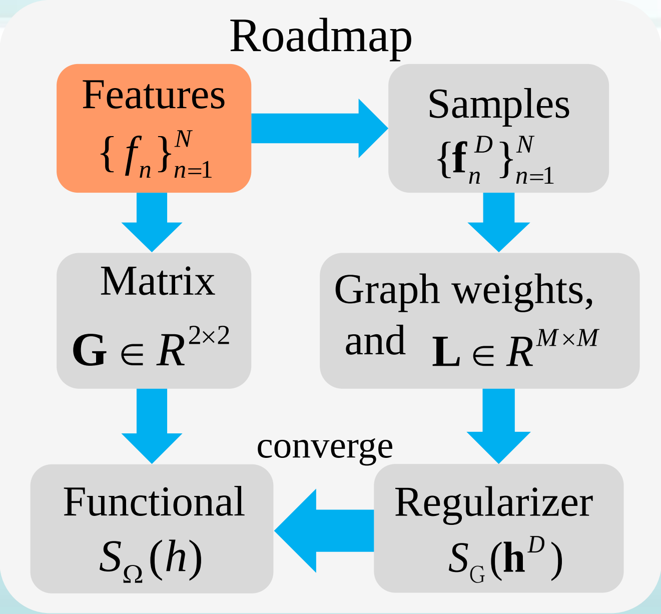
SAMPLE

CONVERGE

- Different features $\{f_n\}_{n=1}^N$ lead to different regularization behavior!

Graph Construction (I)

- First, define:
  - 2D domain $\Omega \subset R^2$
    —shape of an image patch
  - $\Gamma = \left\{ \mathbf{s}_i = [x_i \ y_i]^{\mathrm{T}} \mid \mathbf{s}_i \in \Omega, 1 \le i \le M \right\}$
    —M uniformly distributed random samples on Ω, pixel locations in our work

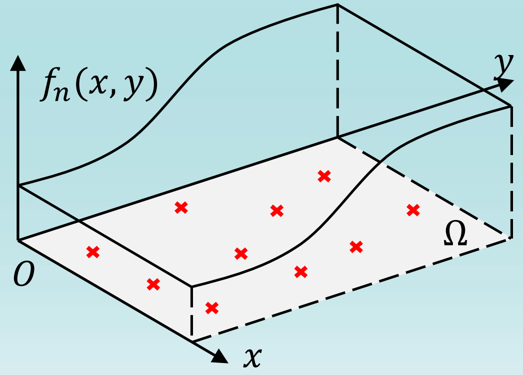

Roadmap
Features $\{f_n\}_{n=1}^N$
Samples $\{\mathbf{f}_n^D\}_{n=1}^N$
Matrix $\mathbf{G} \in R^{2\times 2}$
Graph weights, and $\mathbf{L} \in R^{M\times M}$
converge
Functional $S_\Omega(h)$
Regularizer $S_G(\mathbf{h}^D)$

- (Freely) choose N continuous functions

$$f_n(x,y): \Omega \to R, \ 1 \le n \le N$$

called feature functions, for example

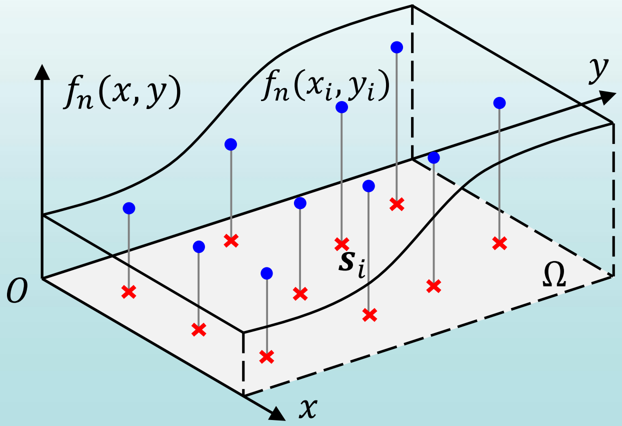
- intensity for gray-scale image $(N = 1)$
- R, G, B channels for color image $(N = 3)$

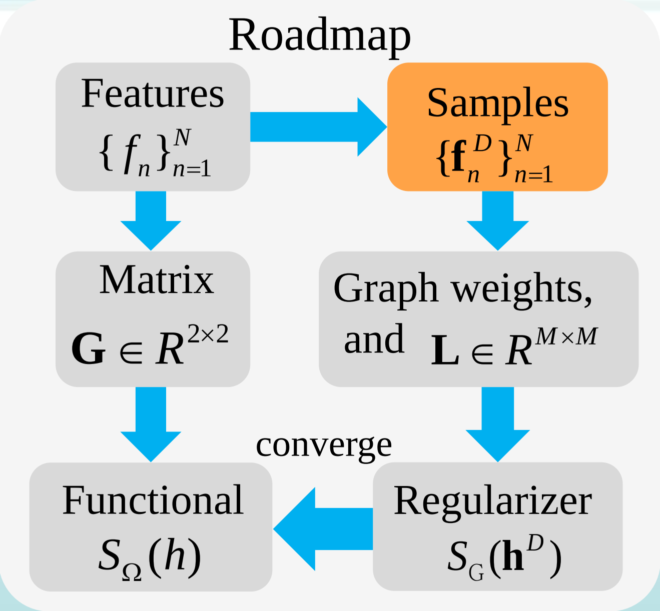


Graph Construction (II)

- Sampling f_n at positions in Γ gives N discretized feature functions

$$\mathbf{f}_n^D = [\, f_n(x_1, y_1)\ f_n(x_2, y_2) \ldots f_n(x_M, y_M)]^{\mathrm{T}}$$




Roadmap

Features $\{f_n\}_{n=1}^N$

Samples $\{\mathbf{f}_n^D\}_{n=1}^N$

Matrix $\mathbf{G} \in R^{2\times 2}$

Graph weights, and $\mathbf{L} \in R^{M\times M}$

converge

Functional $S_\Omega(h)$

Regularizer $S_{\mathrm{G}}(\mathbf{h}^D)$

- For each sample $\mathbf{s}_i \in \Gamma$, define a length N vector

$$\mathbf{v}_i = [\mathbf{f}_1^D(i)\ \mathbf{f}_2^D(i) \ldots \mathbf{f}_N^D(i)]^{\mathrm{T}}$$

- Build a graph G with M vertices; each sample $\mathbf{s}_i \in \Gamma$ has a vertex V_i

Graph Construction (III)

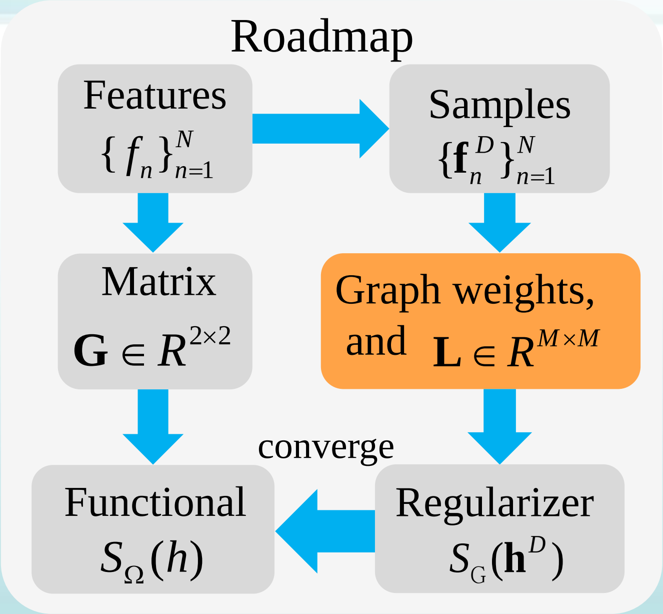
- Weight between vertices V_i and V_j

degree before normalization

$$\rho_i = \sum_{j=1}^{M} \psi(d_{ij})$$

$$w_{ij} = (\rho_i \rho_j)^{-\gamma} \psi(d_{ij})$$

normalization factor γ



Roadmap

Features $\{f_n\}_{n=1}^{N}$

Samples $\{\mathbf{f}_n^D\}_{n=1}^{N}$

Matrix $\mathbf{G} \in R^{2\times 2}$

Graph weights, and $\mathbf{L} \in R^{M\times M}$

converge

Functional $S_\Omega(h)$

Regularizer $S_G(\mathbf{h}^D)$

Clipped Gaussian kernel

$$\psi(d) = \begin{cases} \exp\left(-\dfrac{d^2}{2\varepsilon^2}\right) & |d| \le r, \\ 0 & \text{otherwise} \end{cases}$$

where $r = \varepsilon C_r$ and C_r is a constant

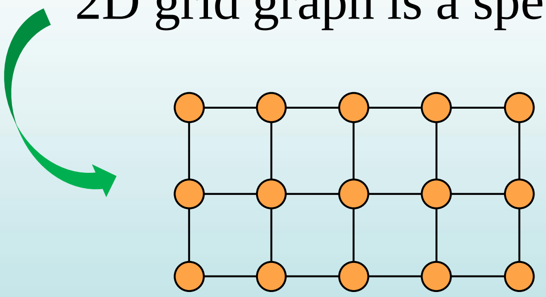
“Distance” between two features

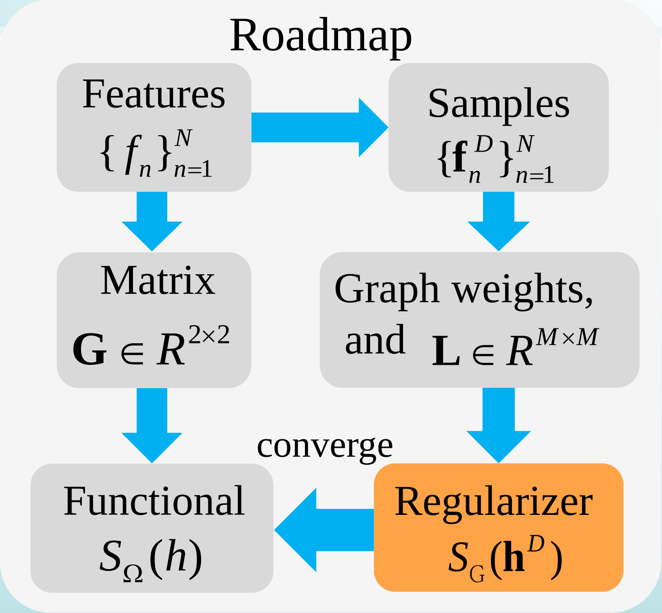
$$d_{ij}^2 = \left\| \mathbf{v}_i - \mathbf{v}_j \right\|_2^2$$

- G is an *r*-neighborhood graph, *i.e.*, no edge connecting two vertices with distance greater than r

Graph Construction (IV)

- Our graph $\mathcal{G}$ is very general
  - *e.g.*, one can derive that the popular 2D grid graph is a special case of ours





Roadmap

Features $\{f_n\}_{n=1}^{N}$

Samples $\{\mathbf{f}_n^D\}_{n=1}^{N}$

Matrix $\mathbf{G} \in R^{2\times 2}$

Graph weights, and $\mathbf{L} \in R^{M\times M}$

converge

Functional $S_\Omega(h)$

Regularizer $S_\mathcal{G}(\mathbf{h}^D)$

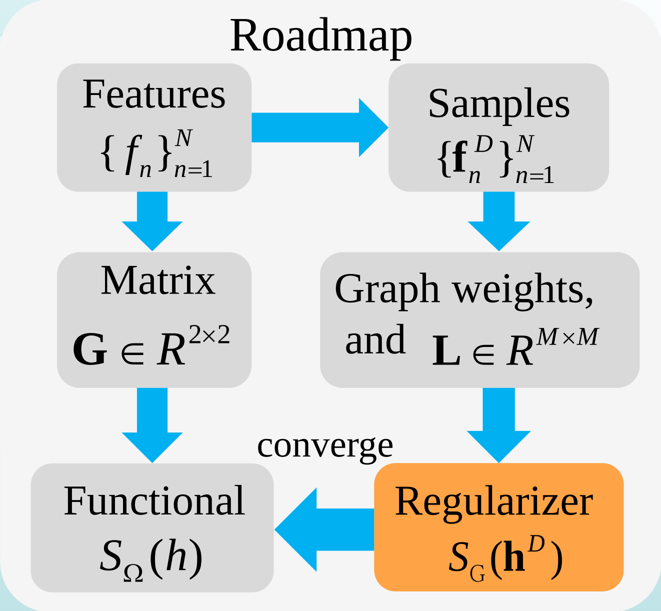
- $$\left.\begin{array}{l} \mathbf{A} \ \text{—}(i,j) \text{ -th entry is } w_{ij} \\ \mathbf{D} \ \text{— diagonal entry is} \sum_{j=1}^{m} w_{ij} \end{array}\right\}$$ unnormalized Graph Laplacian $$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

- $h(x,y):\Omega \rightarrow R$ is some continuous **candidate function**

  $\mathbf{h}^D = [h(x_1,y_1)\ h(x_2,y_2) \ldots h(x_M,y_M)]^{\mathrm{T}}$ — discrete version of $h(x,y)$

- $S_\mathcal{G}(\mathbf{h}^D) = (\mathbf{h}^D)^{\mathrm{T}}\mathbf{L}\mathbf{h}^D$ — graph Laplacian regularizer, functional in R^M

Convergence of the Graph Laplacian Regularizer (I)


Roadmap
Features $\{ f_n \}_{n=1}^N$
Samples $\{\mathbf{f}_n^D\}_{n=1}^N$
Matrix $\mathbf{G} \in R^{2\times 2}$
Graph weights, and $\mathbf{L} \in R^{M\times M}$
converge
Functional $S_\Omega(h)$
Regularizer $S_G(\mathbf{h}^D)$

- The continuous counterpart of S_G is a functional S_Ω on domain Ω

$$S_\Omega(h) = \iint_\Omega (\nabla h)^T \mathbf{G}^{-1} (\nabla h) \left(\sqrt{\det \mathbf{G}}\right)^{2\gamma-1} dxdy$$

$\nabla h = [\partial_x h \;\; \partial_y h]^T$ is the gradient of h

- $\mathbf{G}$ is a 2-by-2 matrix:

$$\mathbf{G} = \begin{bmatrix} \sum_{n=1}^N (\partial_x f_n)^2 & \sum_{n=1}^N \partial_x f_n \cdot \partial_y f_n \\ \sum_{n=1}^N \partial_x f_n \cdot \partial_y f_n & \sum_{n=1}^N (\partial_y f_n)^2 \end{bmatrix} = \sum_{n=1}^N \nabla f_n \cdot (\nabla f_n)^T$$

Structure tensor [9] of the gradients $\{\nabla f_n(x,y)\}_{n=1}^N$

- $\mathbf{G}$ is computed from $\{\nabla f_n\}_{n=1}^N$ on a point-by-point basis

[9] H. Knutsson, C.-F. Westin, and M. Andersson, "**Representing local structure using tensors ii**," in *Image Analysis*. Springer, 2011, vol. 6688, pp. 545–556.

Convergence of the Graph Laplacian Regularizer (II)

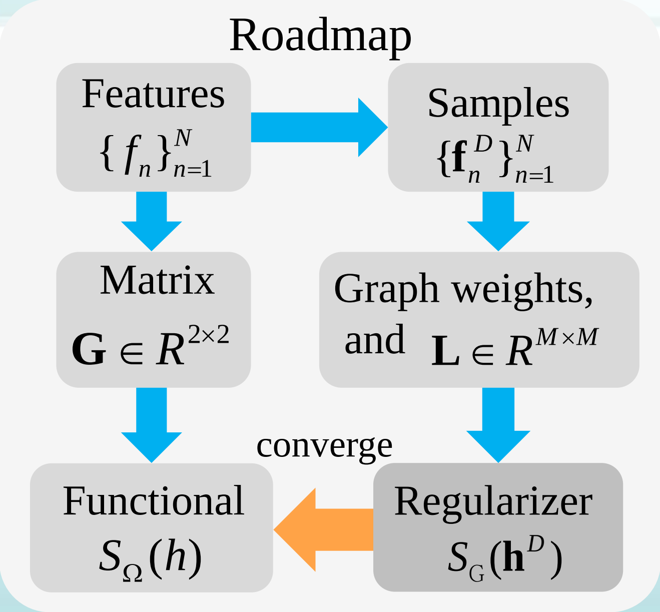
- ***Theorem :*** *convergence of* S_G *to* S_Ω

$$\lim_{\substack{M\to\infty\\ \varepsilon\to 0}} S_G\left(\mathrm{h}^D\right) \sim S_\Omega\left(h\right)$$

1) number of samples M increases

2) neighborhood $r = \varepsilon C_r$ shrinks

"~" means there exist a constant such that equality holds.



Roadmap

Features $\{f_n\}_{n=1}^N$

Samples $\{\mathbf{f}_n^D\}_{n=1}^N$

Matrix $\mathbf{G} \in R^{2\times 2}$

Graph weights, and $\mathbf{L} \in R^{M\times M}$

converge

Functional $S_\Omega(h)$

Regularizer $S_G(\mathbf{h}^D)$

- With results of [10], we proved it by viewing a graph as proxy of an N -dimensional Riemannian manifold

Vertex	Coordinate on Ω	Coordinate on N-D manifold
V_i	$\mathbf{s}_i = \left(x_i, y_i\right)$	$\mathbf{v}_i = [\mathbf{f}_1^D(i)\ \mathbf{f}_2^D(i) \ldots \mathbf{f}_N^D(i)]^T$

[10] M. Hein, "**Uniform convergence of adaptive graph-based regularization**," in *Learning Theory*. Springer, 2006, pp. 50–64.

Interpretation of Graph Laplacian Regularizer (I)

- S_{G} converges to S_{Ω} , with S_{Ω} , any new insights we gain on S_{G} ??
- Inspect the equations carefully…

$$S_{\Omega}(h) = \iint_{\Omega} (\nabla h)^{T} \mathrm{G}^{-1} (\nabla h) \left(\sqrt{\det \mathrm{G}}\right)^{2\gamma-1} dx\, dy$$

$$\mathbf{G} = \sum_{n=1}^{N} \nabla f_n \cdot (\nabla f_n)^{\mathrm{T}}$$

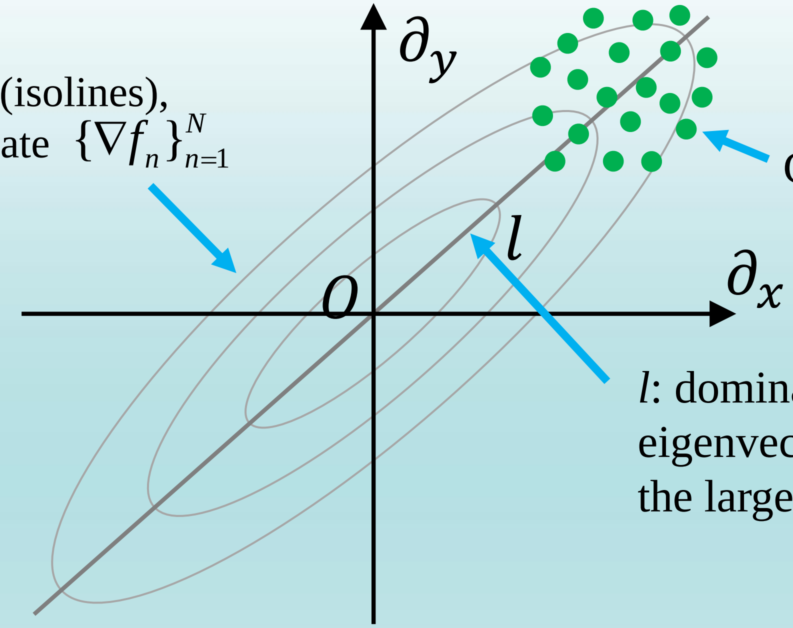
$$S_{\mathrm{G}}(\mathbf{h}^{D}) = (\mathbf{h}^{D})^{\mathrm{T}} \mathbf{L} \mathbf{h}^{D}$$

- 3 observations:
  - $(\nabla h)^{\mathrm{T}} \mathbf{G}^{-1} (\nabla h)$ measures length of ∇h in a metric space built by $\mathbf{G}$!
  - The eigen-space of $\mathbf{G}$ reflects dominant directions of $\{\nabla f_n\}_{n=1}^{N}$
  - S_{Ω} integrates the gradient norm

Justification of Graph Laplacian Regularizer (II)

- Metric space defined by $\mathbf{G}$?
  - At a certain location (x, y) on the image

Ellipses are contours (isolines), reflects how concentrate $\{\nabla f_n\}_{n=1}^{N}$



Green dots are $\{\nabla f_n\}_{n=1}^{N}$

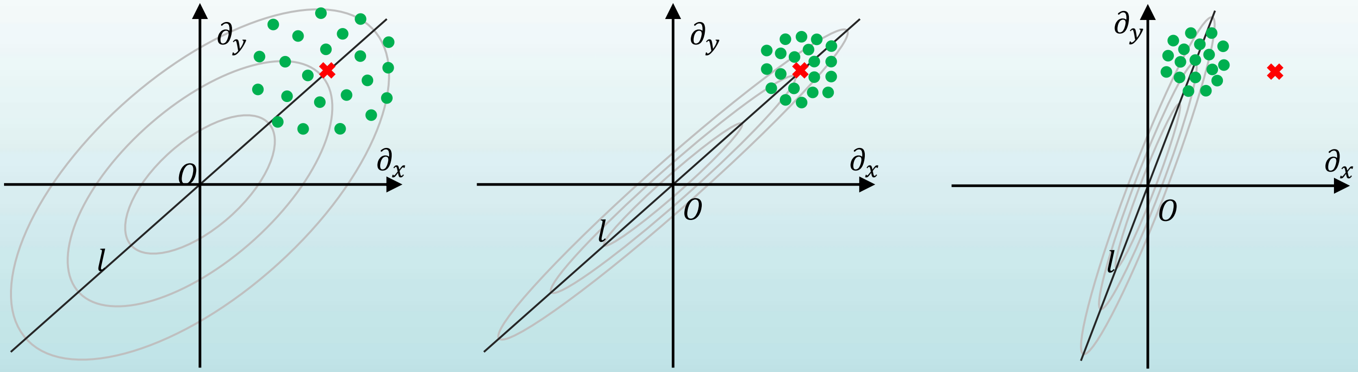
l: dominant direction, eigenvector corresponds to the largest eigenvalue of $\mathbf{G}$

$$S_\Omega(h) = \iint_\Omega (\nabla h)^{\mathrm{T}} \mathbf{G}^{-1} (\nabla h) dxdy \qquad \mathbf{G} = \sum_{n=1}^{N} \nabla f_n \cdot (\nabla f_n)^{\mathrm{T}}$$

Justification of Graph Laplacian Regularizer (III)

- The 2D metric space provides a clear picture of *what signals are being discriminated and to what extent*, on a point-by-point basis in the continuous domain.

✖ ground-truth



(a) (b) (c)

  - Both (a)(b) are correct, but (b) is more discriminant, (c) is discriminant but incorrect

- **Lesson**: when ground-truth is unknown, *one should design a discriminant metric space only to the extent that estimates of ground-truth are reliable*!

Noise Modeling in Gradient Domain

- For a $\sqrt{M} \times \sqrt{M}$ noisy patch $\mathbf{p}_0 \in R^M$, identify $K-1$ similar patches on the noisy image, the K patches $\{\mathbf{p}_k\}_{k=0}^{K-1}$ form a *cluster*

- On patch $\mathbf{p}_k$, gradient at pixel i is $\mathbf{g}_k^i$.
- Drop superscript i, model the noisy gradients $\{\mathbf{g}_k\}_{k=0}^{K-1}$ as

$$\mathbf{g}_k = \mathbf{g} + \mathbf{e}_k, 0 \le k \le K-1$$

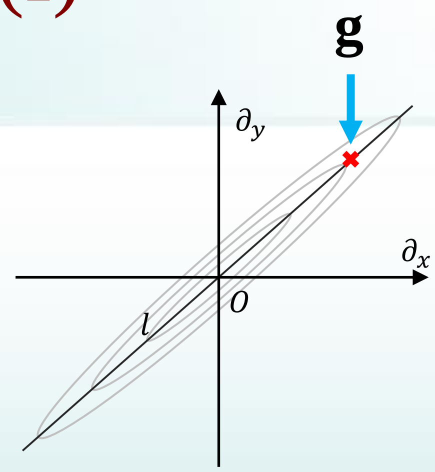
Unknown ground-truth

Noise term, follows 2D Gaussian with zero-mean and covariance $\sigma_e^2 \mathbf{I}$

- PDF of $\mathbf{g}_k$ given ground-truth $\mathbf{g}$ (likelihood) is simply

$$Pr(\mathbf{g}_k \mid \mathbf{g}) = \frac{1}{2\pi\sigma_e^2} \exp\left(-\frac{1}{2\sigma_e^2} \left\| \mathbf{g} - \mathbf{g}_k \right\|_2^2 \right)$$

Seeking for the Optimal Metric Space (I)



- We first establish an ideal metric space assuming we know ground truth: $\mathbf{g}$

$$\mathbf{G}_0(\mathbf{g}) = \mathbf{g}\mathbf{g}^{\mathrm{T}} + \alpha\mathbf{I}$$

It is discriminant to $\mathbf{g}$

$\alpha > 0$, smaller α makes the space more skewed

- With noisy gradients $\{\mathbf{g}_k\}_{k=0}^{K-1}$ seek for the optimal metric space

Δ is the whole gradient domain

posterior prob. of ground truth

$$\mathbf{G}^{\cdot} = \arg\min_{\mathbf{G}} \iint_{\Delta} \left\|\mathbf{G} - \mathbf{G}_0(\mathbf{g})\right\|_F^2 \, Pr\left(\mathbf{g} \mid \{\mathbf{g}_k\}_{k=0}^{K-1}\right) d\mathbf{g}$$

$$\Rightarrow \mathbf{G}^{\cdot} = \iint_{\Delta} \mathbf{G}_0(\mathbf{g}) \cdot Pr\left(\mathbf{g} \mid \{\mathbf{g}_k\}_{k=0}^{K-1}\right) d\mathbf{g} \qquad (1)$$

Seeking for the Optimal Metric Space (II)

- Assume the prior $Pr(\mathbf{g})$ is a 2D Gaussian with covariance $\sigma_g^2\mathbf{I}$ we derive

$$\boxed{Pr\left(\mathbf{g} \mid \{\mathbf{g}_k\}_{k=0}^{K-1}\right) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{1}{2\sigma^2}\left\|\mathbf{g}-\mathbf{g}_\mu\right\|_2^2\right)}$$

where the "ensemble" mean $\mathbf{g}_\mu$ and variance σ^2 are

noise variance of $\mathbf{g}_k$

$$\mathbf{g}_\mu = \frac{1}{K+\sigma_e^2/\sigma_g^2}\sum_{k=0}^{K-1}\mathbf{g}_k \qquad \sigma^2 = \frac{\sigma_e^2}{K+\sigma_e^2/\sigma_g^2}$$

- Carrying out the integral in (1) gives the optimal metric space

$$\mathbf{G}^{\cdot} = \mathbf{g}_\mu\mathbf{g}_\mu^{\mathrm{T}} + (\sigma^2+\alpha)\mathbf{I} \quad (2)$$

- **Intuition:** *If noise σ^2 is small, $\mathbf{g}_\mu\mathbf{g}_\mu^{\mathrm{T}}$ dominates and $\mathbf{G}^{\cdot}$ is discriminant; if σ^2 is large, $(\sigma^2+\alpha)\mathbf{I}$ dominates, $\mathbf{G}^{\cdot}$ defaults to Euclidean space!*

From Metric Space to Graph Laplacian

- The structure of $\mathbf{G}^{\cdot} = \mathbf{g}_{\mu}\mathbf{g}_{\mu}^{\mathrm{T}} + (\sigma^2 + \alpha)\mathbf{I}$ allows us to select $N = 3$ feature functions, such that they lead to the optimal metric space:

$$\mathbf{f}_1^D(i) = \sqrt{\sigma^2 + \alpha} \cdot x_i \qquad \mathbf{f}_2^D(i) = \sqrt{\sigma^2 + \alpha} \cdot y_i \qquad \text{— Spatial}$$

$$\mathbf{f}_3^D = \frac{1}{K + \sigma_e^2 / \sigma_g^2} \sum_{k=0}^{K-1} \mathbf{p}_k \qquad \text{— Intensity}$$

- $\mathbf{f}_1^D(i)$ and $\mathbf{f}_2^D(i)$ correspond to the term $(\sigma^2 + \alpha)\mathbf{I}$ in $\mathbf{G}^{\cdot}$
- $\mathbf{f}_3^D(i)$ leads to the term $\mathbf{g}_{\mu}\mathbf{g}_{\mu}^{\mathrm{T}}$ in $\mathbf{G}^{\cdot}$

- Our work is closely-related to *joint* (or *cross*) bilateral filtering, with the averaging of similar patches as guidance image.

- However, we adapt to noise, resulting in robust weight estimates.

Formulation and Algorithm

- Adopt a patch-based recovery framework, for a noisy patch $\mathbf{p}_0$
  1. Find $K-1$ patches similar to $\mathbf{p}_0$ in terms of Euclidean distance.
  2. Compute the feature functions, leading to edge weights and Laplacian.
  3. Solve the unconstrained quadratic optimization:

$$\boxed{\mathrm{q}^* = \arg\min_{\mathrm{q}} \|\mathrm{p}_0 - \mathrm{q}\|_2^2 + \lambda \mathrm{q}^T \mathrm{L}\mathrm{q} \quad \Rightarrow \quad \mathrm{q} = (\mathrm{I} + \lambda \mathrm{L})^{-1} \mathrm{p}_0}$$

  to obtain the denoised patch.

- Aggregate denoised patches to form an updated image.
- Denoise the image iteratively to gradually enhance its quality.
- Optimal Graph Laplacian Regularization for Denoising (OGLRD).

Experimentation (I)

- Test images: *Lena*, *Boats*, *Peppers* and *Airplane*
- i.i.d. Additive White Gaussian Noise (AWGN)
- Compare OGLRD to NLM and BF

1.5 dB better than NLM!

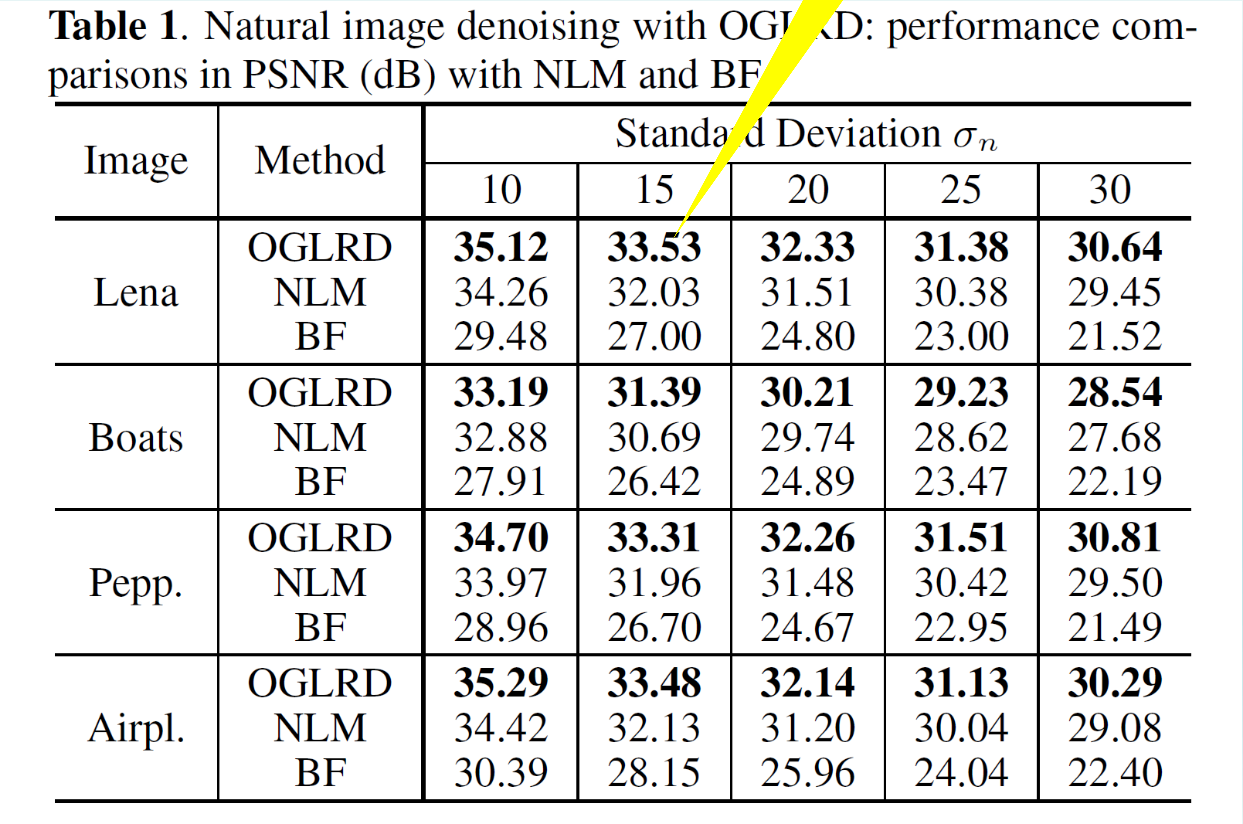
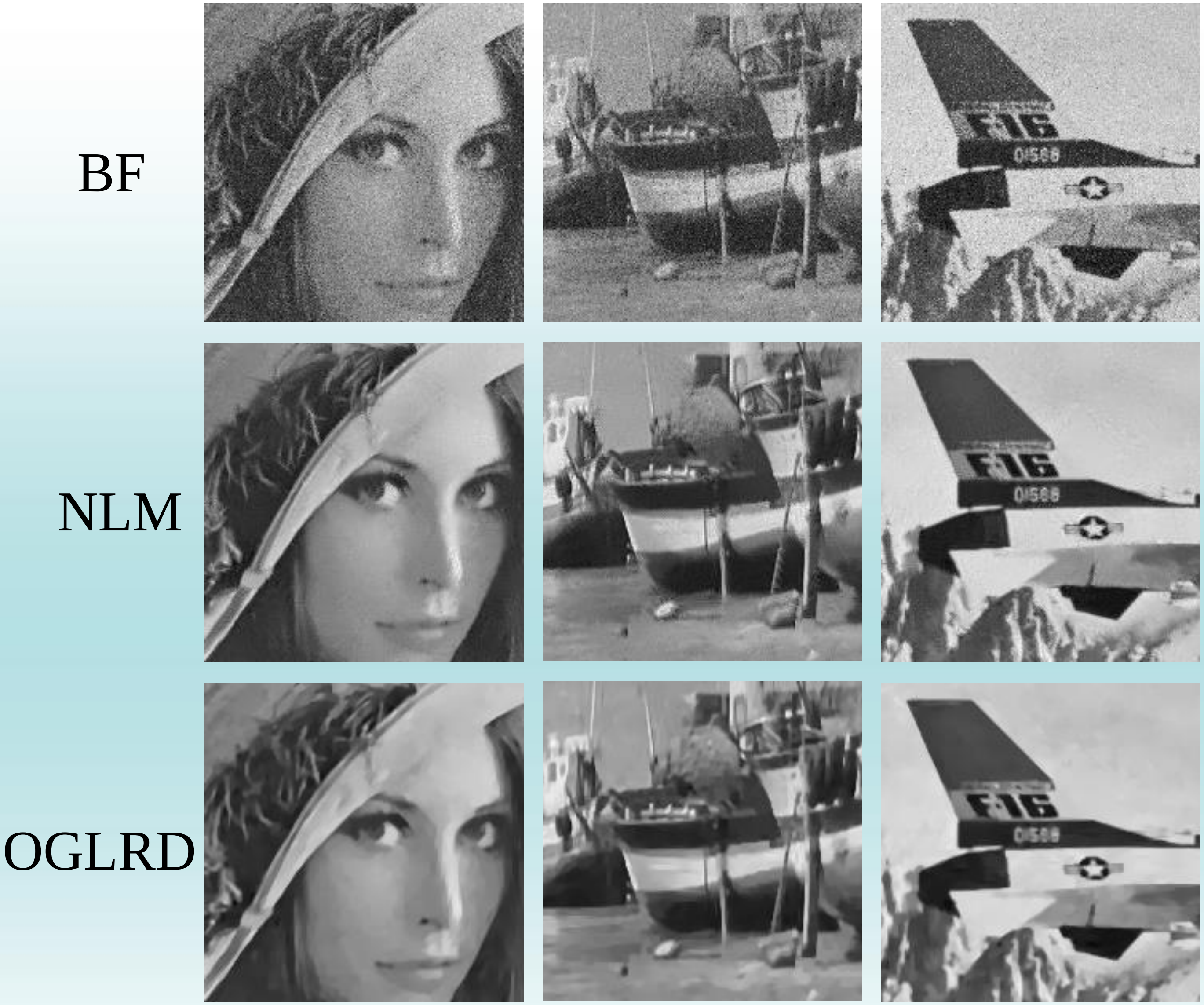


Table 1. Natural image denoising with OGLRD: performance comparisons in PSNR (dB) with NLM and BF.

Image	Method	Standard Deviation σ_n				
		10	15	20	25	30
Lena	OGLRD	**35.12**	**33.53**	**32.33**	**31.38**	**30.64**
	NLM	34.26	32.03	31.51	30.38	29.45
	BF	29.48	27.00	24.80	23.00	21.52
Boats	OGLRD	**33.19**	**31.39**	**30.21**	**29.23**	**28.54**
	NLM	32.88	30.69	29.74	28.62	27.68
	BF	27.91	26.42	24.89	23.47	22.19
Pepp.	OGLRD	**34.70**	**33.31**	**32.26**	**31.51**	**30.81**
	NLM	33.97	31.96	31.48	30.42	29.50
	BF	28.96	26.70	24.67	22.95	21.49
Airpl.	OGLRD	**35.29**	**33.48**	**32.14**	**31.13**	**30.29**
	NLM	34.42	32.13	31.20	30.04	29.08
	BF	30.39	28.15	25.96	24.04	22.40

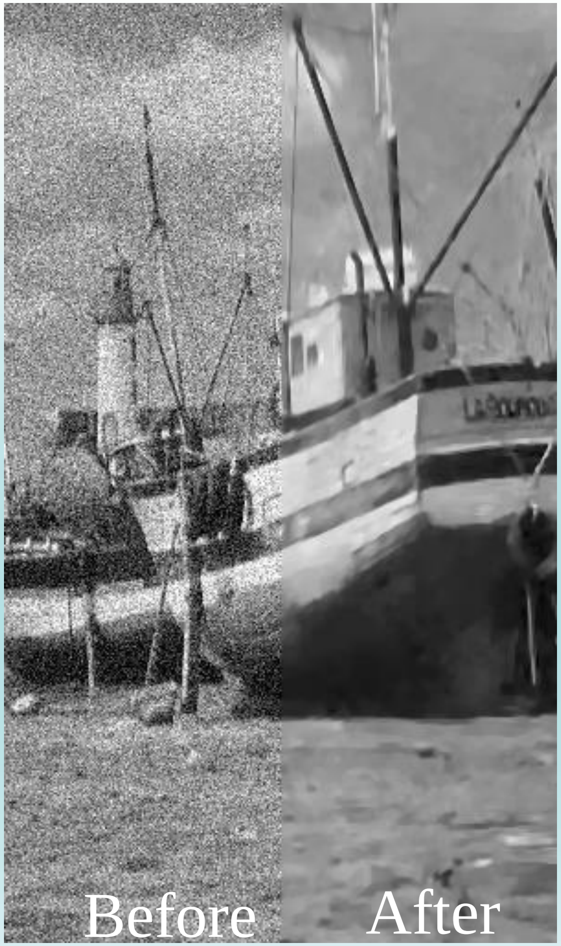
Experimental Results (II)

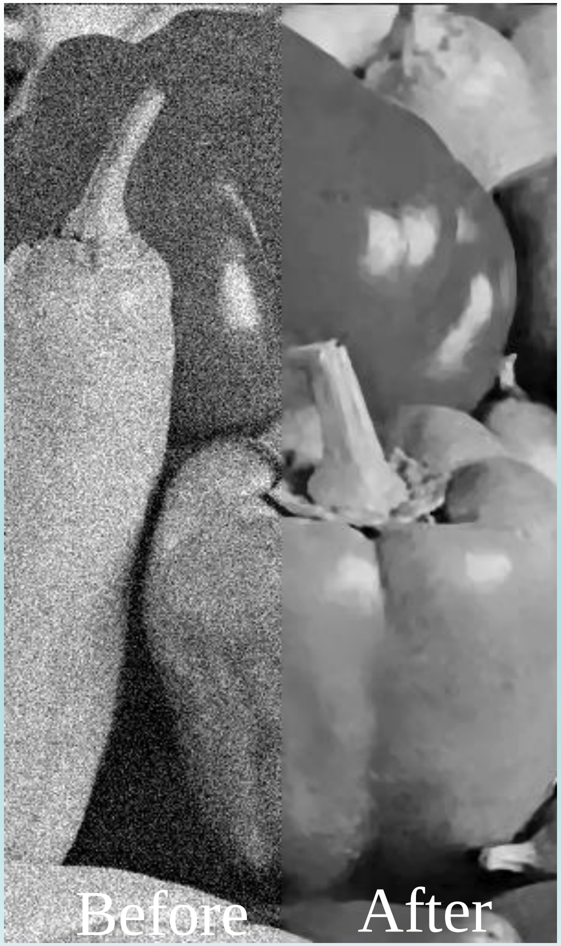
- Visual comparisons ($\sigma_n = 25$) of fragments

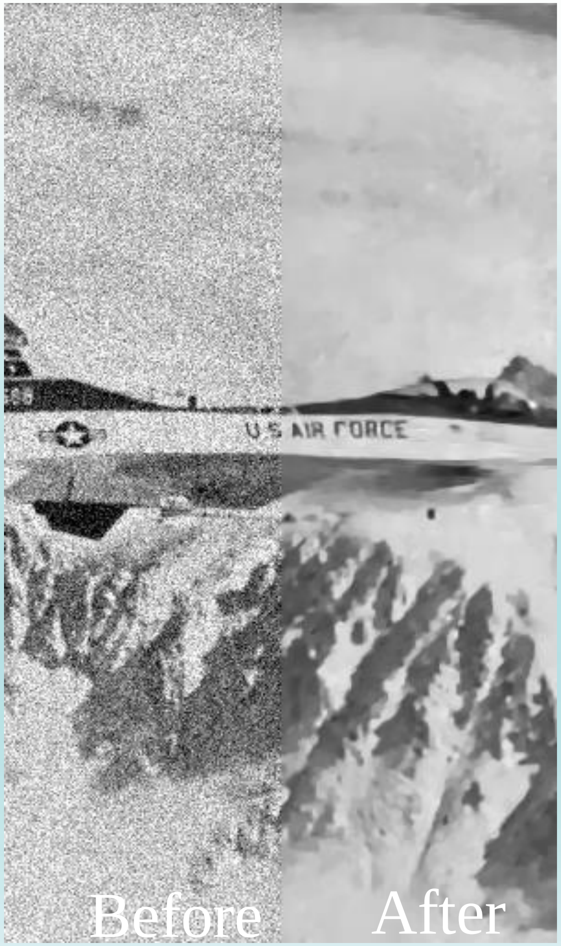


Experimental Results (III)

- Some visual results when $\sigma_n = 30$







Summary

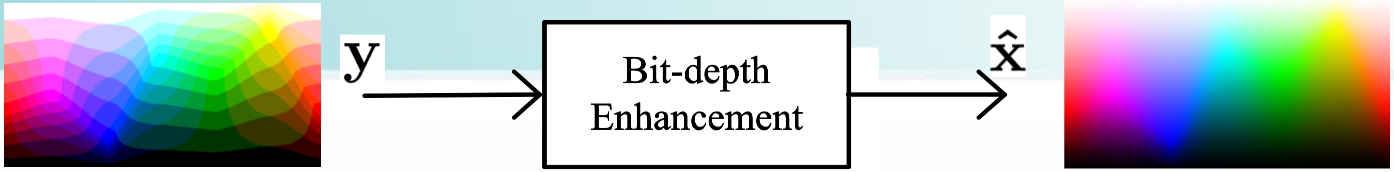
- Image denoising is an ill-posed problem; we use **graph Laplacian regularizer** as prior for regularization.
- *Graph Laplacian regularizer with Gaussian kernel weights converges to a continuous functional*.
- Analysis of the continuous functional provides theoretical justification of why and to what extent the graph Laplacian regularizer can be discriminant.
- We describe a methodology to *derive the optimal edge weights* given nonlocal noisy gradient observations.
- Our denoising algorithm with graph Laplacian regularizer and gradient-based similarity out-performs NLM by up to 1.5 dB.

Outline

- Depth Image Denoising
  - Graph Sparsity Prior
  - Graph-signal Smoothness Prior
- Bit-depth Enhancement

Image Bit-depth Enhancement

Problem:



$\mathbf{y}$

Bit-depth Enhancement

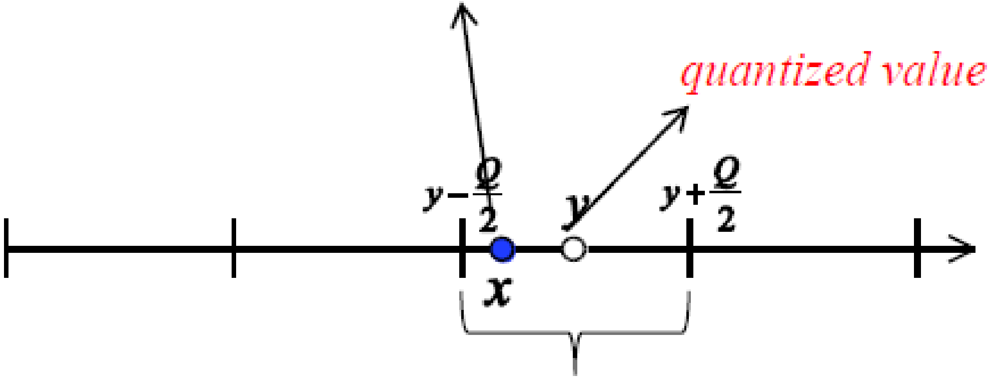
$\hat{\mathbf{x}}$

low bit-depth (LBD) image **y**—a *quantized* version of underlying HBD image $\mathbf{x}$

an estimate of the original HBD image

pre-quantized value $x \in [0,1)$

quantized value $y = \mathrm{quant}(x)$



scalar quantizer with bit-depth b and quantization step size $Q = \frac{1}{2^b}$

Quantization bin, defining the uncertainty range $x \in [y - \frac{Q}{2}, y + \frac{Q}{2})$

[11] Rudin et al, "Nonlinear total variation based noise removal algorithms", Elsevier, 1992

Image Bit-depth Enhancement

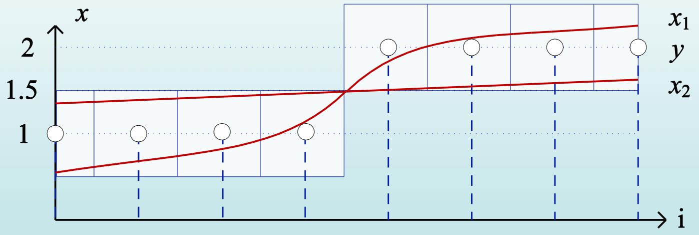
Objective: find $\hat{x}$ that minimizes mean-squared-error (MSE),

squared err

posterior prob of HBD signal x given LBD signal y

$$\hat{x}^{MMSE} = \arg\min_{\hat{x}} \int \left\| \hat{x} - x \right\|_2^2 \, f(x \mid y)\, dx$$

Posterior: $f(\mathbf{x}|\mathbf{y}) \propto f(\mathbf{y}|\mathbf{x})\; f(\mathbf{x})$



Likelihood: equals to 1 iff x_i quantizes to y_i

$$f(\mathbf{y}|\mathbf{x}) = \begin{cases} 1, & \text{if } \mathrm{quant}(x_i) = y_i, \ \forall i \\ 0, & \text{otherwise} \end{cases}$$

Smoothness prior: HBD signal is smooth

Conventional smoothness (*e.g.*, total Variation) are signal-independent → ***over-smoothing***

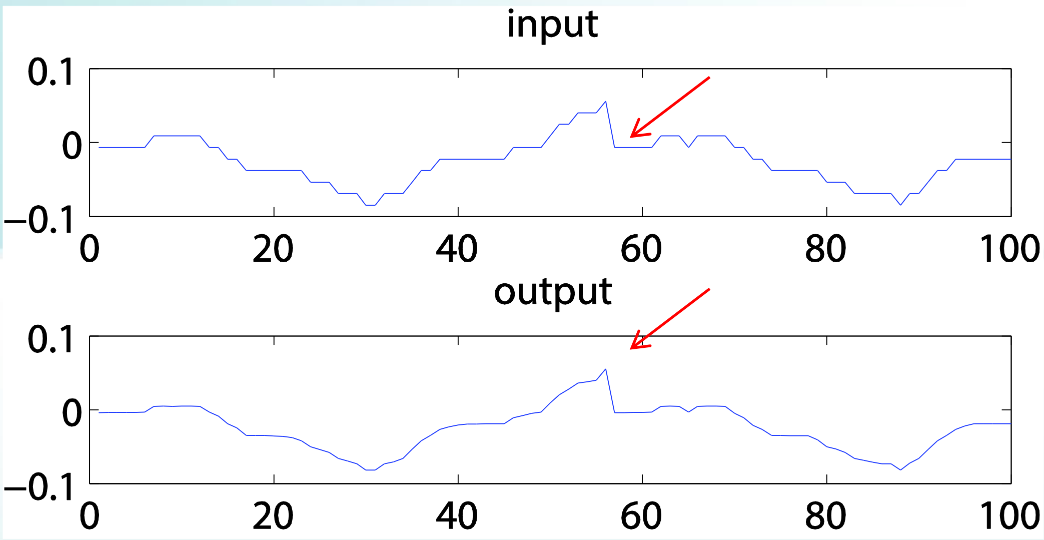
Question: what's a good signal smoothness prior?

Image Bit-depth Enhancement

Graph-signal smoothness prior

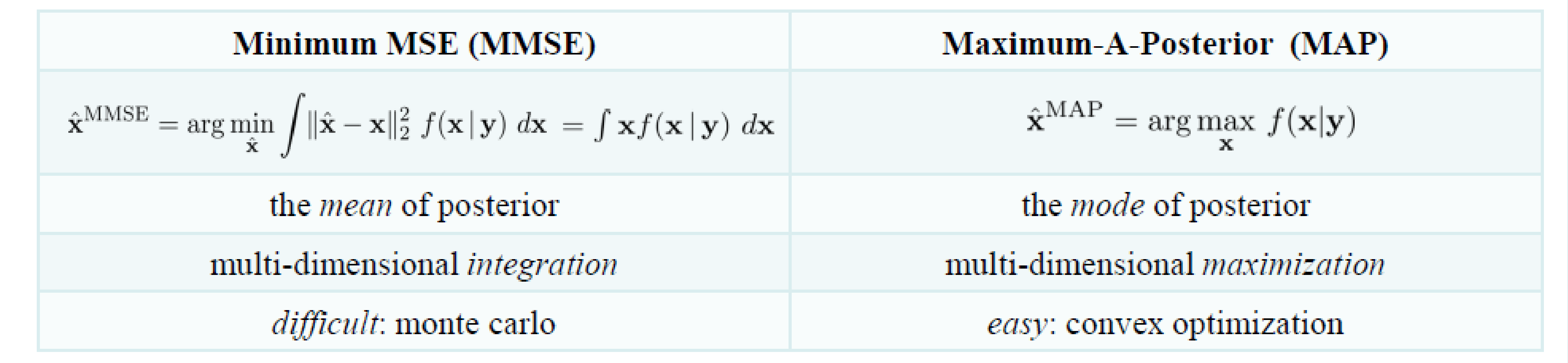
$$f(\mathbf{x}) = \frac{1}{K}\exp\{-\sigma\mathbf{x}^{\intercal}\mathbf{L}\mathbf{x}\}$$

L is the graph Laplacian matrix describing inter-pixel similarities*



Reconstruct smooth signal without blurring edges

- MMSE problem is now well posed, but difficult to solve.

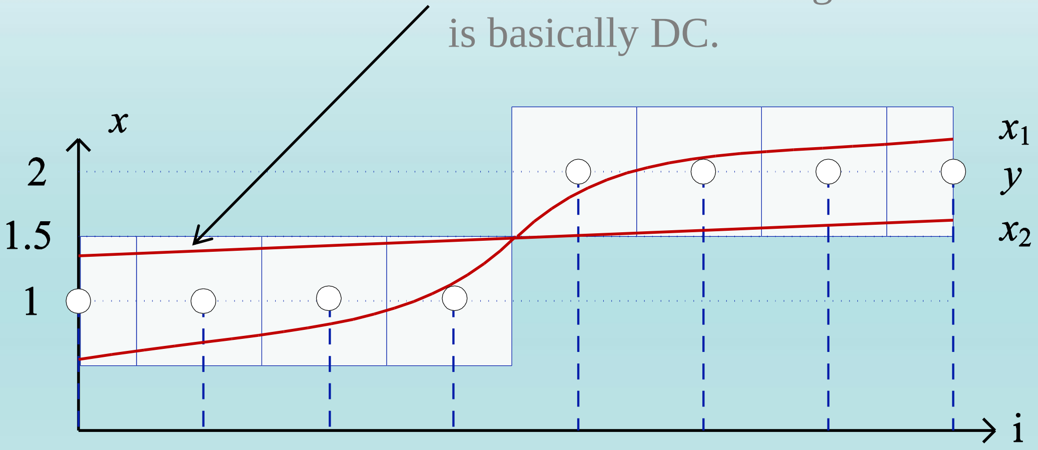


Minimum MSE (MMSE)	Maximum-A-Posterior (MAP)
$\hat{\mathbf{x}}^{\mathrm{MMSE}} = \arg\min_{\hat{\mathbf{x}}} \int \lVert\hat{\mathbf{x}} - \mathbf{x}\rVert_2^2\, f(\mathbf{x}\,\vert\,\mathbf{y})\; d\mathbf{x} \;=\; \int \mathbf{x} f(\mathbf{x}\,\vert\,\mathbf{y})\; d\mathbf{x}$	$\hat{\mathbf{x}}^{\mathrm{MAP}} = \arg\max_{\mathbf{x}}\; f(\mathbf{x}\vert\mathbf{y})$
the *mean* of posterior	the *mode* of posterior
multi-dimensional *integration*	multi-dimensional *maximization*
difficult: monte carlo	*easy*: convex optimization

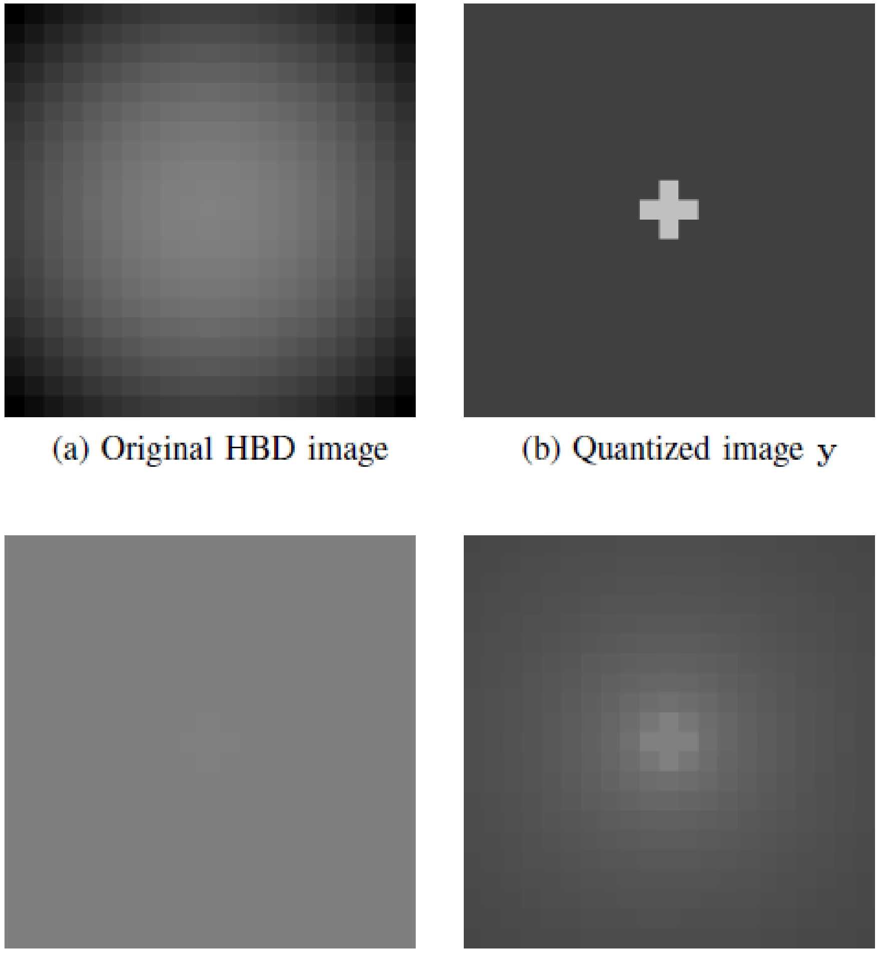
[12] P. Wan, G. Cheung, D. Florencio, C. Zhang, O. Au, "**Image Bit-depth Enhancement via Maximum-a-Posteriori Estimation of Graph AC Component,"** *IEEE International Conference on Image Processing*, Paris, France, October, 2014. (Top 10% accepted paper recognition)

Image Bit-depth Enhancement

- MAP finds smoothest solution in feasible space.
  - Can have arbitrarily large MSE!



Smoothest feasible signal
is basically DC.



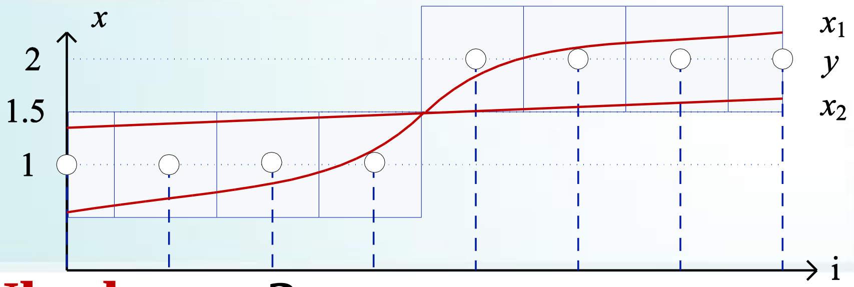
(a) Original HBD image

(b) Quantized image $\mathbf{y}$

(c) Output by (8)

(d) Output by proposed method

Image Bit-depth Enhancement



Proposed ACDC Algorithm:

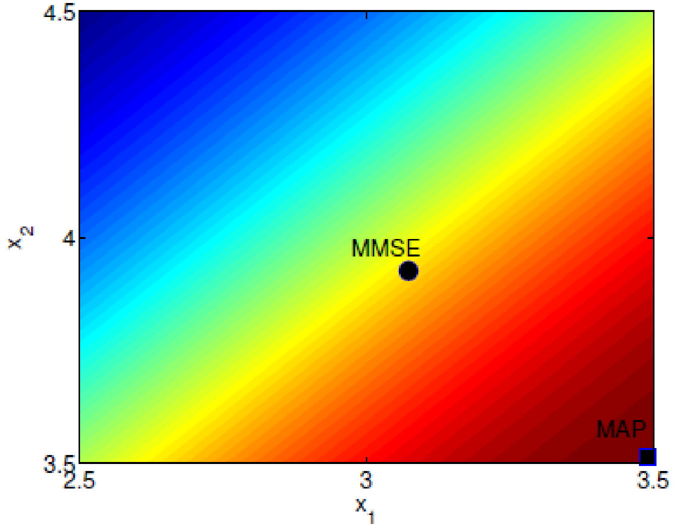
- Compute edge weights from quantized signal.
- Compute MAP solution of AC signal.
- Compute MMSE solution of DC signal given AC signal.

Why better?

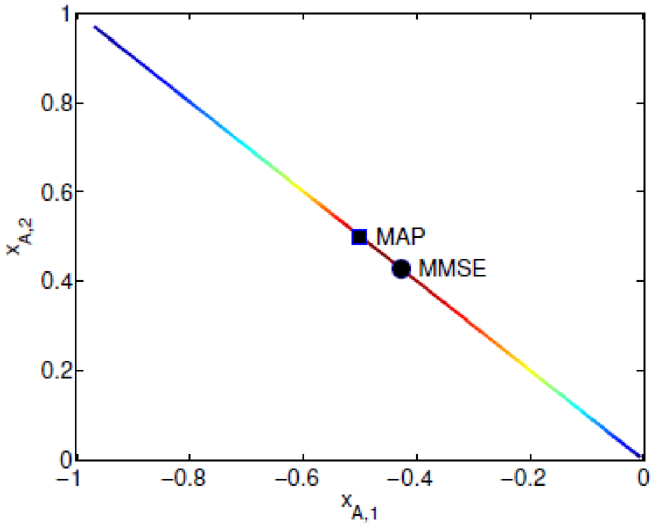
$$\mathrm{x} = \mathrm{x}_A + x_D$$

- Posterior ≈ likelihood * prior
- Likelihood of AC $f(\mathrm{y}\,|\mathrm{x}_A)$ is less skewed

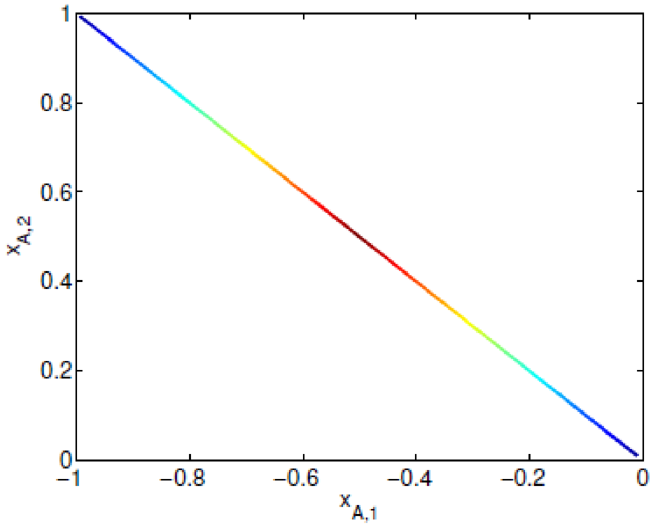
(integrating over possible DC values).



(a) Posterior PDF $f(\mathrm{x}|\mathrm{y})$ in direct MAP (8). Likelihood function (3) defines a square feasible space where the MAP solution is at the corner.



(b) Posterior PDF of AC signal $f(\mathrm{x}_A|\mathrm{y})$ in MAP AC estimation (22). Note MAP solution is close to MMSE solution.



(c) Likelihood function of AC signal (19) in MAP AC estimation (22). Feasible space of AC signal (20) is a line segment.

Experiments

Numerical comparison:

c1 (720x480) c2 (912x688) c3 (1024x768) c4 (922x901)

c5 (853x520) g1 (640x480) g2 (800x480) g3 (440x320)

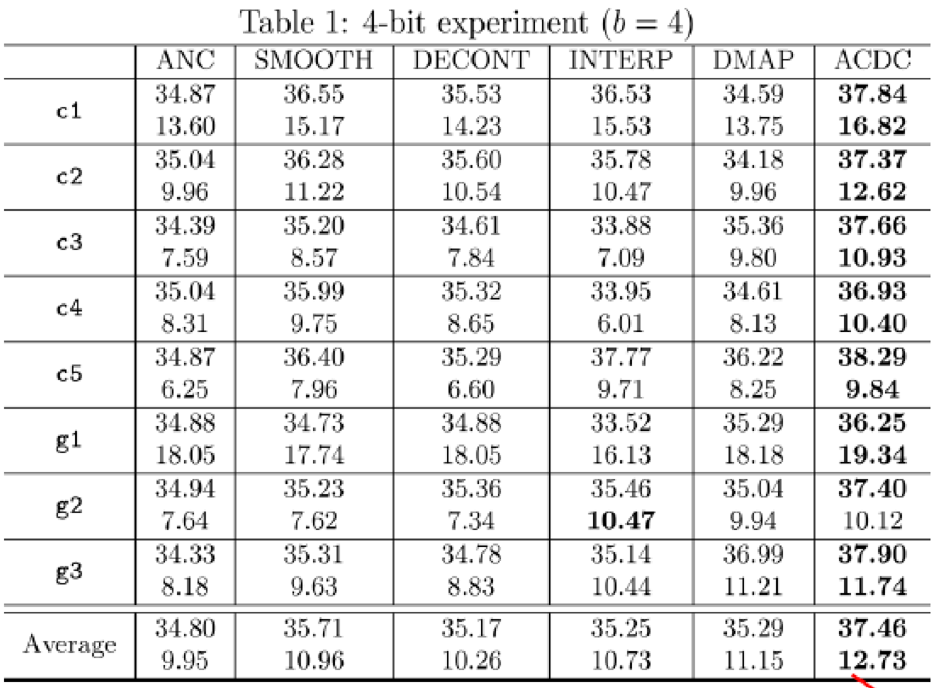


Table 1: 4-bit experiment ($b = 4$)

	ANC	SMOOTH	DECONT	INTERP	DMAP	ACDC
c1	34.87 13.60	36.55 15.17	35.53 14.23	36.53 15.53	34.59 13.75	**37.84** **16.82**
c2	35.04 9.96	36.28 11.22	35.60 10.54	35.78 10.47	34.18 9.96	**37.37** **12.62**
c3	34.39 7.59	35.20 8.57	34.61 7.84	33.88 7.09	35.36 9.80	**37.66** **10.93**
c4	35.04 8.31	35.99 9.75	35.32 8.65	33.95 6.01	34.61 8.13	**36.93** **10.40**
c5	34.87 6.25	36.40 7.96	35.29 6.60	37.77 9.71	36.22 8.25	**38.29** **9.84**
g1	34.88 18.05	34.73 17.74	34.88 18.05	33.52 16.13	35.29 18.18	**36.25** **19.34**
g2	34.94 7.64	35.23 7.62	35.36 7.34	35.46 **10.47**	35.04 9.94	**37.40** 10.12
g3	34.33 8.18	35.31 9.63	34.78 8.83	35.14 10.44	36.99 11.21	**37.90** **11.74**
Average	34.80 9.95	35.71 10.96	35.17 10.26	35.25 10.73	35.29 11.15	**37.46** **12.73**

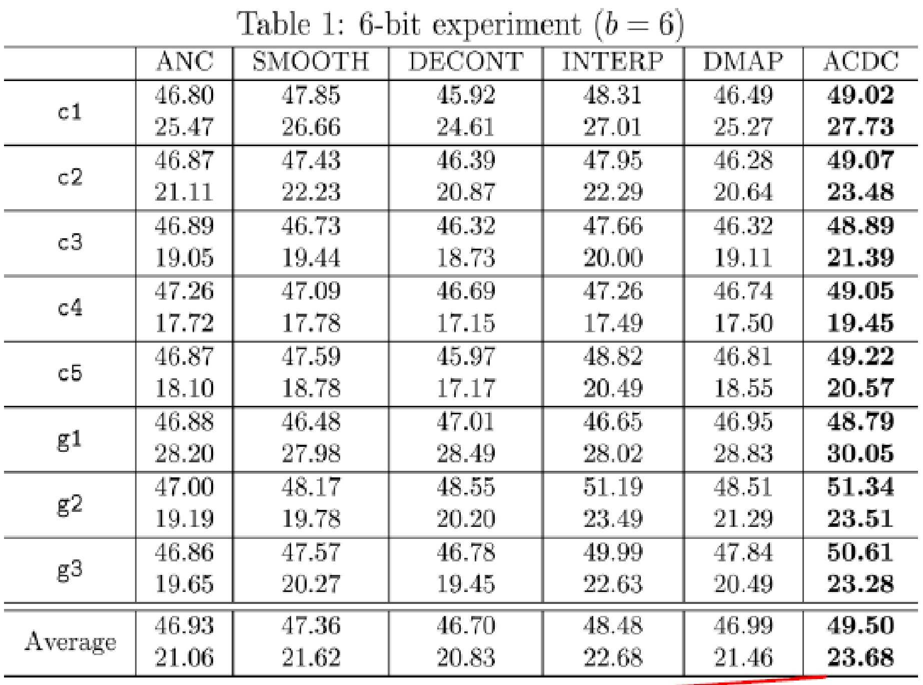


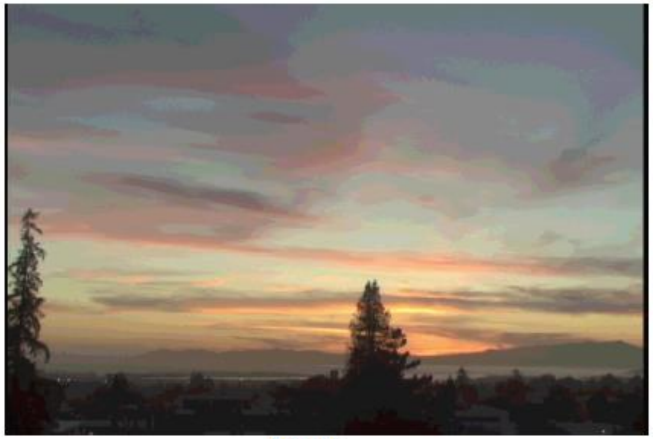
Table 1: 6-bit experiment ($b = 6$)

	ANC	SMOOTH	DECONT	INTERP	DMAP	ACDC
c1	46.80 25.47	47.85 26.66	45.92 24.61	48.31 27.01	46.49 25.27	**49.02** **27.73**
c2	46.87 21.11	47.43 22.23	46.39 20.87	47.95 22.29	46.28 20.64	**49.07** **23.48**
c3	46.89 19.05	46.73 19.44	46.32 18.73	47.66 20.00	46.32 19.11	**48.89** **21.39**
c4	47.26 17.72	47.09 17.78	46.69 17.15	47.26 17.49	46.74 17.50	**49.05** **19.45**
c5	46.87 18.10	47.59 18.78	45.97 17.17	48.82 20.49	46.81 18.55	**49.22** **20.57**
g1	46.88 28.20	46.48 27.98	47.01 28.49	46.65 28.02	46.95 28.83	**48.79** **30.05**
g2	47.00 19.19	48.17 19.78	48.55 20.20	51.19 23.49	48.51 21.29	**51.34** **23.51**
g3	46.86 19.65	47.57 20.27	46.78 19.45	49.99 22.63	47.84 20.49	**50.61** **23.28**
Average	46.93 21.06	47.36 21.62	46.70 20.83	48.48 22.68	46.99 21.46	**49.50** **23.68**

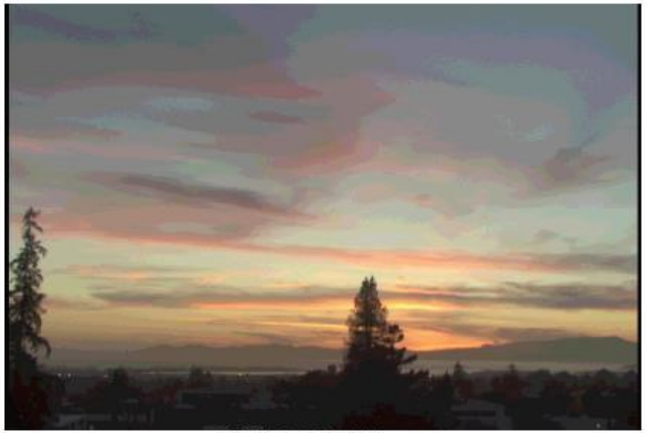
On average, gains over 2.5dB in PSNR over the traditional method

Experiments

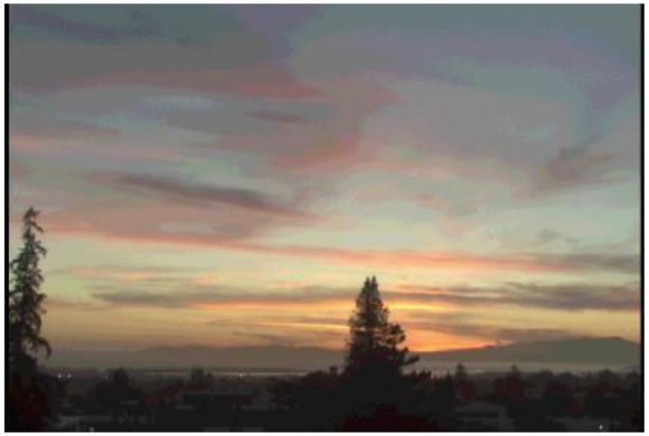
Visual comparison:



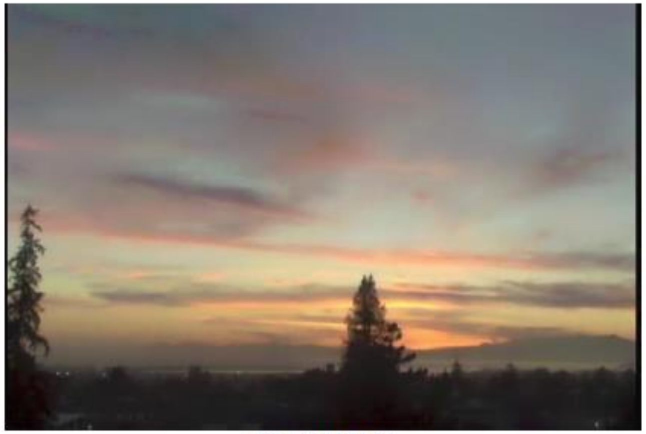
ANC



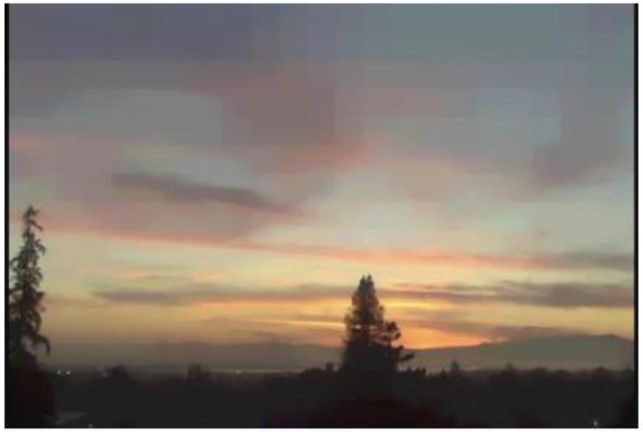
SMOOTH



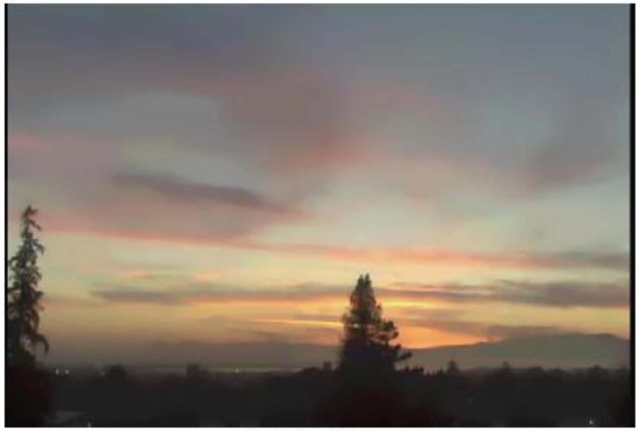
DECONT



INTERP



DMAP



ACDC

Summary

- Inverse imaging requires good signal priors.
- Depth Image Denoising
  - Graph Sparsity Prior (probabilistic interpretation)
  - Graph-signal Smoothness Prior (deterministic interpretation)
- Bit-depth Enhancement
  - Instead of fidelity term, restricted feasible space due to quantization bin constraints (as likelihood term).

Conclusion

Depth Image Coding & Processing

- Coding: graph Fourier Transform (GFT), generalized graph Fourier Transform (GGFT)
- Denoising: graph sparsity prior, graph-signal smoothness prior

Future Work

- Natural image coding using graph-based transforms.
- Depth image denoising / interpolation for non-AWGN noise.
- **Apps**: Given depth images, foreground / background segmentation, tracking, face modeling, etc.