Gene Cheung, Xianming Liu
National Institute of Informatics
11th July, 2016

大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

Graph Signal Processing for Image Compression & Restoration (Part II)

.

Outline (Part II)

- Image Restoration using GSP Tools
  - Image Denoising
  - Soft Decoding of JPEG Compressed Images

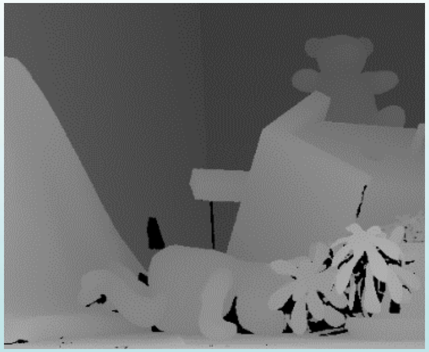
Outline (Part II)

- Image Restoration using GSP Tools
  - Image Denoising
    - Sparsity Prior
    - Smoothness Prior
  - Soft Decoding of JPEG Compressed Images

Introduction to PWS Image Denoising

- Limitations of current sensing technologies

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



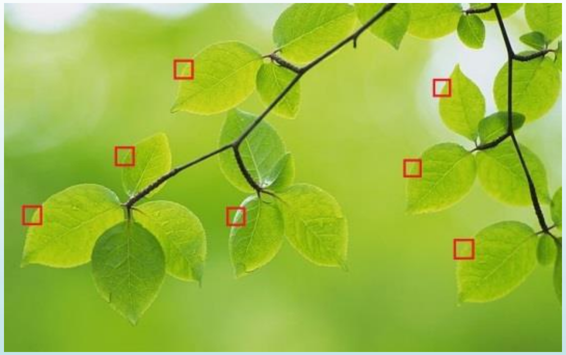
- Denoising is an inverse imaging problem.

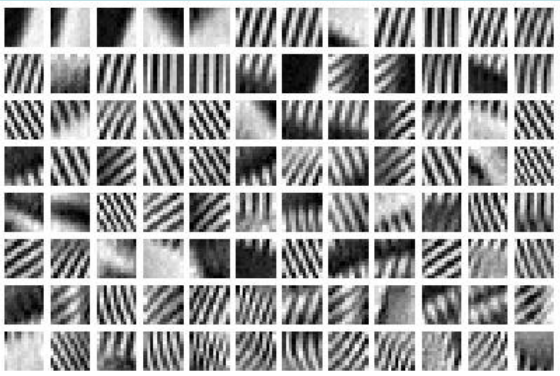
desired signal

observation → $y = x + v$ ← 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

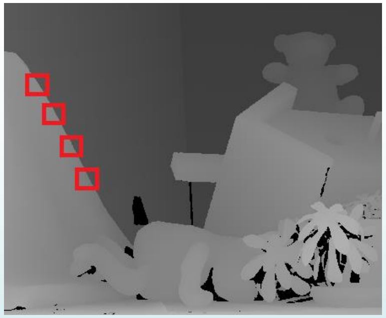
  Elad et al, "Image denoising via sparse and redundant representation over learned dictionaries," *TIP 2006.*

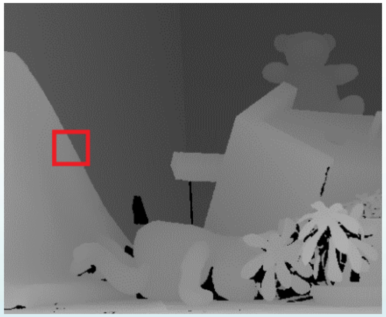
  - represent a signal by the linear combination of a few atoms out of a dictionary

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

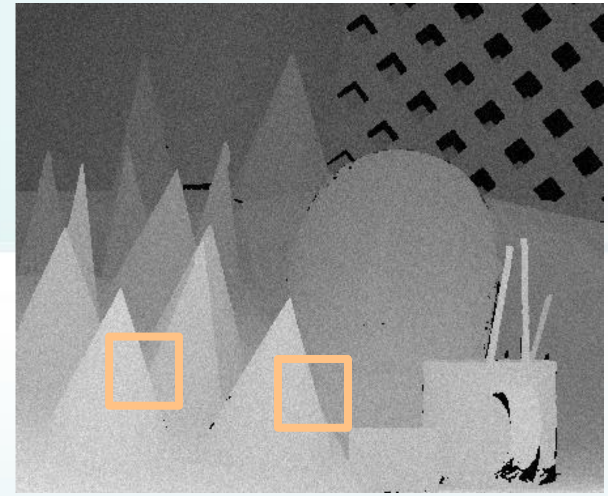
unify in GFT domain

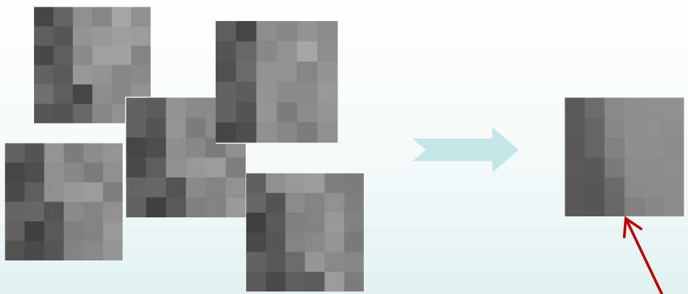
Challenges Our method

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



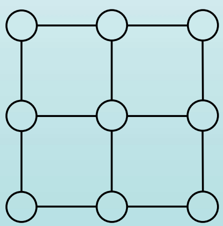


common GFT from avg. patch

observation i

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

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)

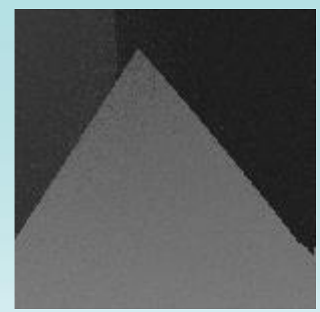
- Setup:
  - Test Middlebury depth maps: *Cones, Teddy, Sawtooth*
  - Add Additive White Gaussian Noise
  - Compare against Bilateral Filtering (BF), Non-Local Means Denoising (NLM) and Block-Matching 3D (BM3D)
- Results
  - Up to 2.28dB improvement over BM3D.

		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 Middleburry depth maps: *Cones*, *Teddy, Sawtooth*

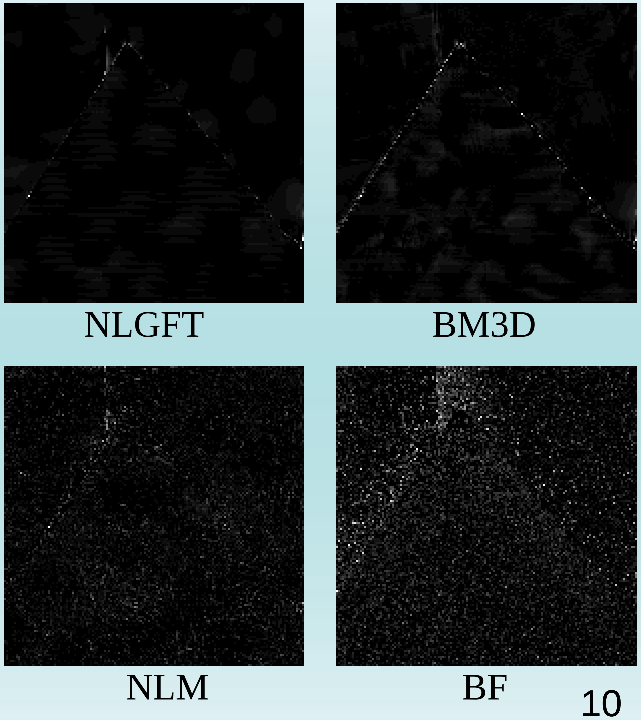
  - Add Additive White Gaussian Noise

  - 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



Outline (Part II)

- Image Restoration using GSP Tools
  - Image Denoising
    - Sparsity Prior
    - Smoothness Prior
  - Soft Decoding of JPEG Compressed Images
  - Joint Denoising / Contrast Enhancement

Motivation (I)

- Image denoising—a basic restoration problem:

$$\mathbf{z} = \mathbf{u} + \mathbf{e}$$

observation → $\mathbf{z}$; noise → $\mathbf{e}$; desired signal → $\mathbf{u}$

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

$$\min_{\mathbf{u}} \|\mathbf{z} - \mathbf{u}\|_2^2 + \tau \ \text{prior}(\mathbf{u})$$

fidelity term → $\|\mathbf{z} - \mathbf{u}\|_2^2$; prior term → $\text{prior}(\mathbf{u})$

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

$$S_G(\mathbf{u}) = \mathbf{u}^{\mathrm{T}} \mathbf{L} \mathbf{u} \qquad \mathbf{L} = \mathbf{D} - \mathbf{A}$$

graph Laplacian matrix → $\mathbf{L}$

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

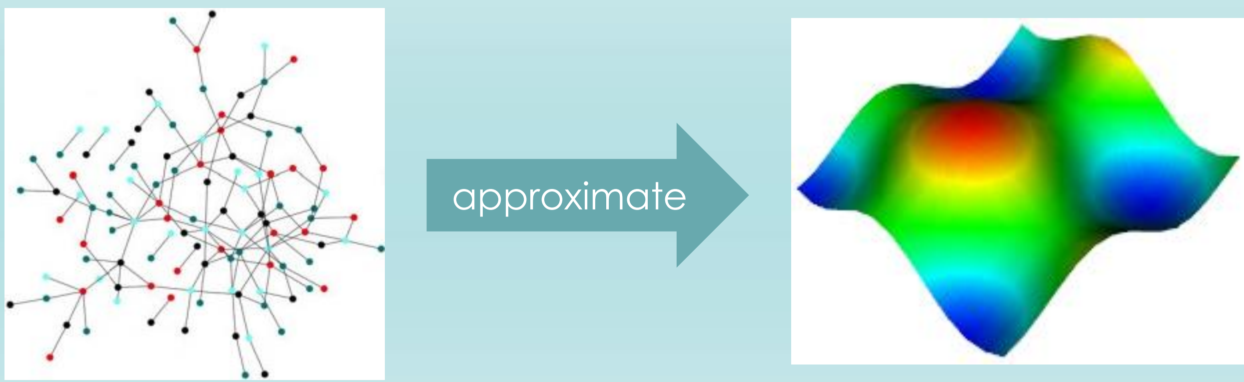
$$w_{ij} = \exp\left(\frac{-dist(i, j)^2}{\sigma^2} \right)$$

$\text{dist}(i, j)$ 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)

- However…
  a. Why is $S_G(\mathbf{x}) = \mathbf{u}^T\mathbf{L}\mathbf{u}$ a good prior?
  b. How to design the optimal $\mathbf{u}^T\mathbf{L}\mathbf{u}$ for restoration?
  c. $\mathbf{u}^T\mathbf{L}\mathbf{u}$ performs particularly well on PWS images, why?
- We answer these basic questions by viewing:
  - discrete graph as **samples** of high-dimensional manifold.



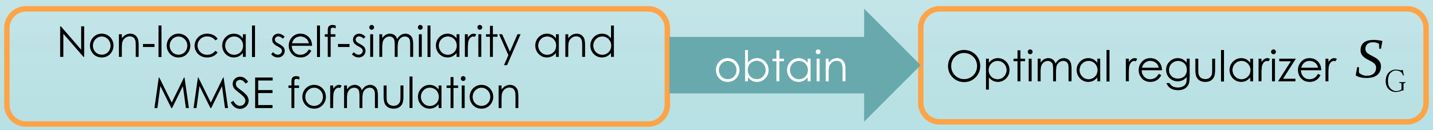
discrete graph continuous manifold

[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 and Gene Cheung, "**Graph Laplacian Regularization for Inverse Imaging: Analysis in the Continuous Domain**," arXiv preprint, arXiv:1604.07948, 2016.

Our Contributions

1. We show $S_{\mathrm{G}}(\mathbf{u}) = \mathbf{u}^{\mathrm{T}}\mathbf{L}\mathbf{u}$ converges to a continuous functional S_{Ω}, analysis of S_{Ω} explains the mechanism of $\mathbf{u}^{\mathrm{T}}\mathbf{L}\mathbf{u}$ for inverse imaging

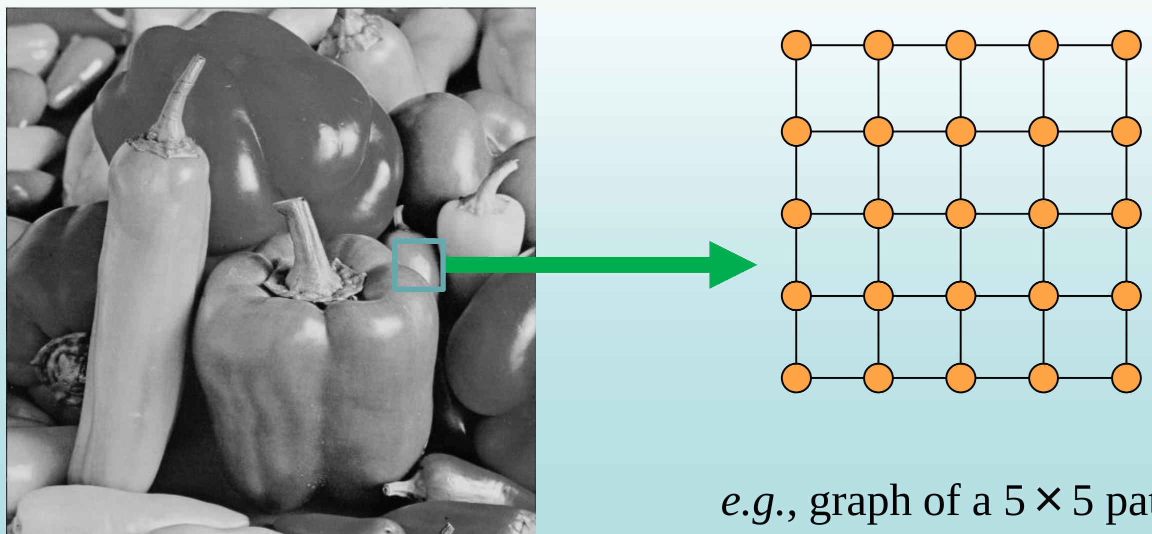
2. We derive the optimal graph Laplacian regularizer for denoising, which is discriminant for small noise and robust when very noisy.



3. We interpret graph Laplacian regularization as anisotropic diffusion, show that it not only smooths but may also sharpens the image, promote piecewise smooth images

Graph-Based Image Processing

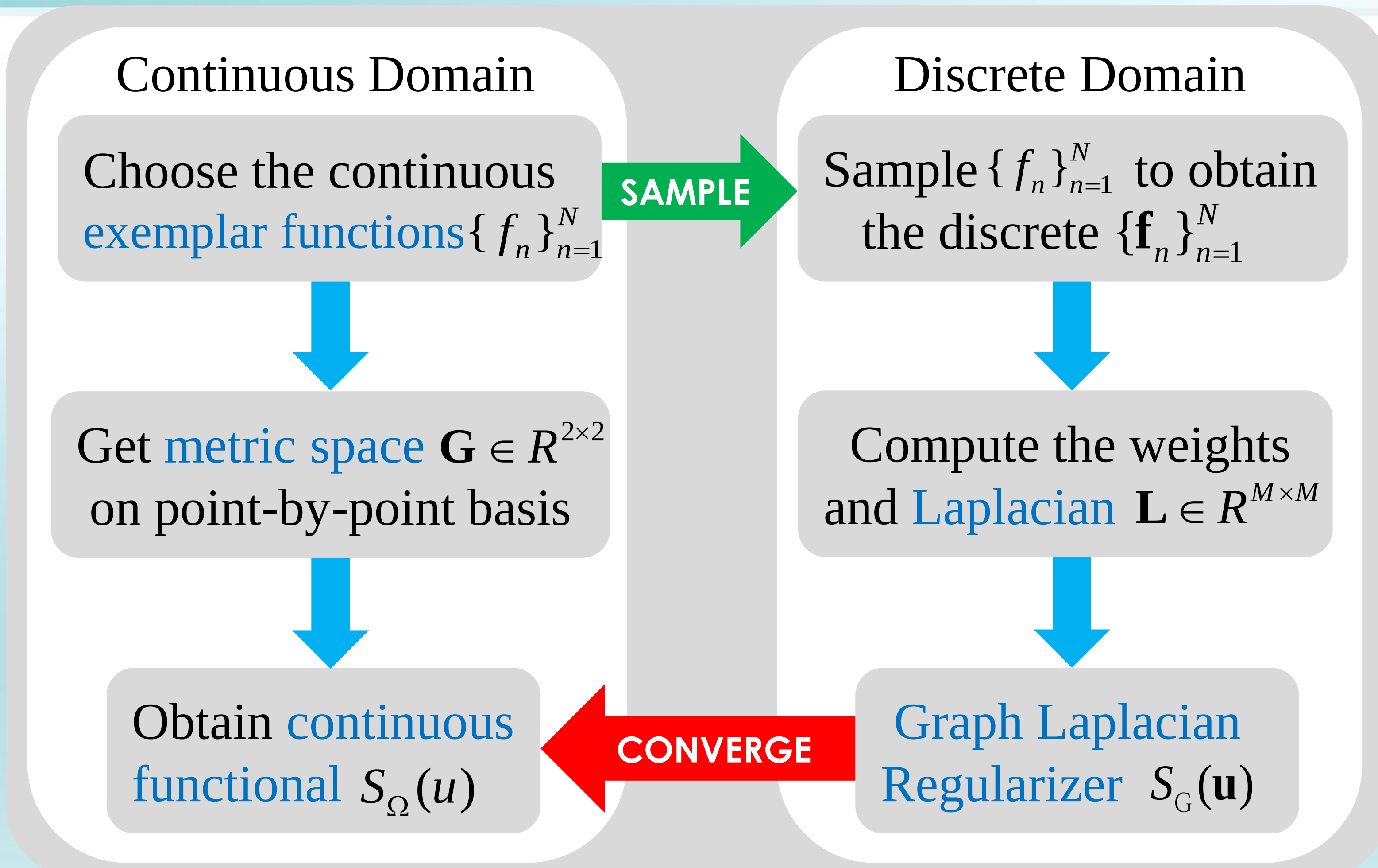
- Graph for image restoration
  - Each pixel corresponds to a vertex in a graph (denote # of pixels as M).



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

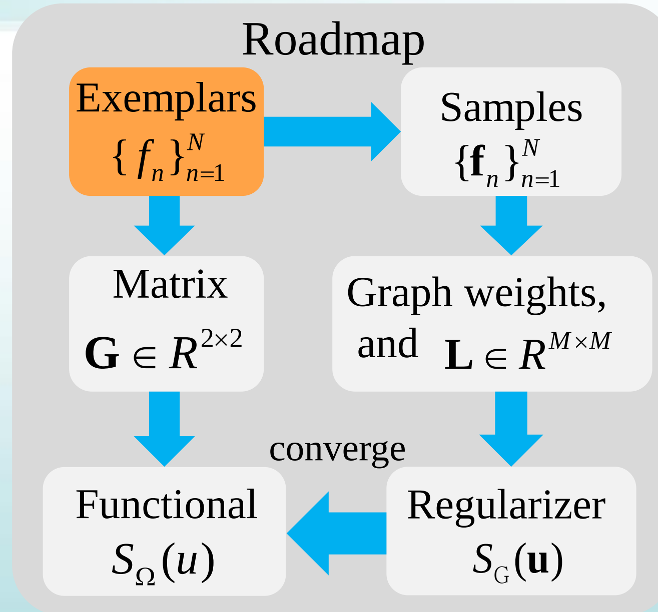
  - Regard the image as a signal defined on a weighted graph.
  - With proper graph configuration, construct filter for image (graph signal) using prior knowledge (i.e., smooth on the graph).

Road Map



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

Graph Construction (I)

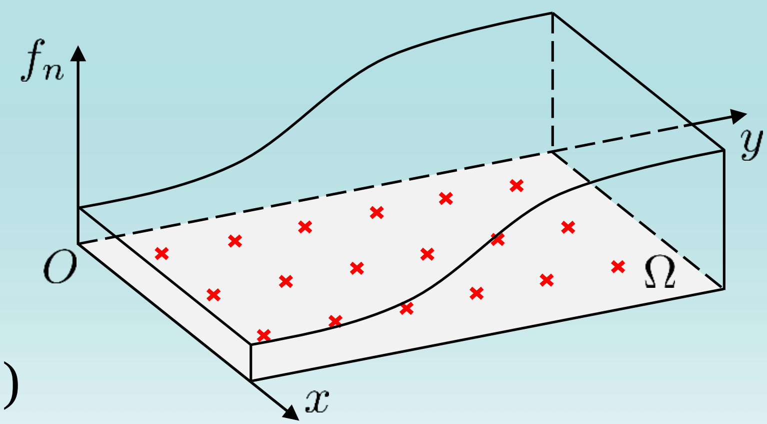


- First, define:
  - 2D domain $\Omega \subset R^2$
    —shape of an image patch
  - $\Gamma = \left\{ \mathbf{s}_i = [x_i \ y_i]^\mathrm{T} \mid \mathbf{s}_i \in \Omega, 1 \le i \le M \right\}$
    —M uniformly distributed random samples on Ω, pixel locations in our work

- (Freely) choose N continuous functions

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

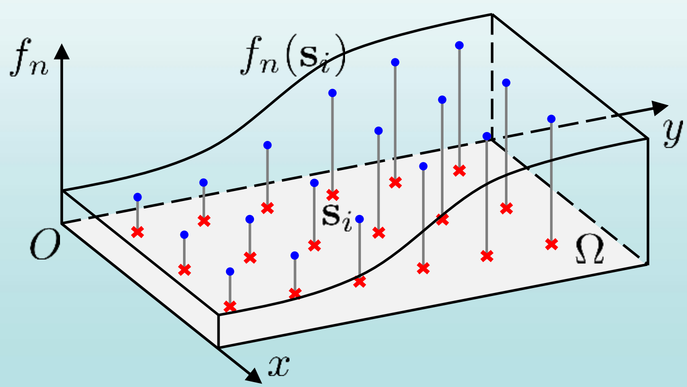
called exemplar functions, for example
  - intensity for gray-scale image ($N = 1$)
  - R, G, B channels for color image ($N = 3$)

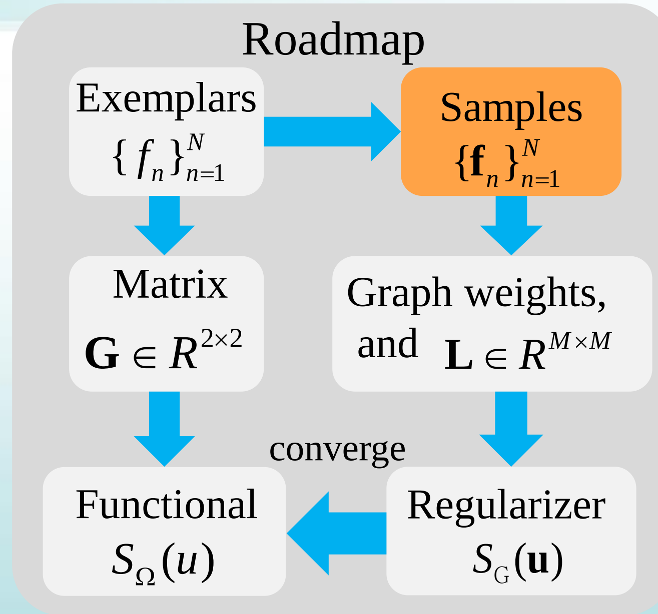


Graph Construction (II)

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

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





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

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

- 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

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

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

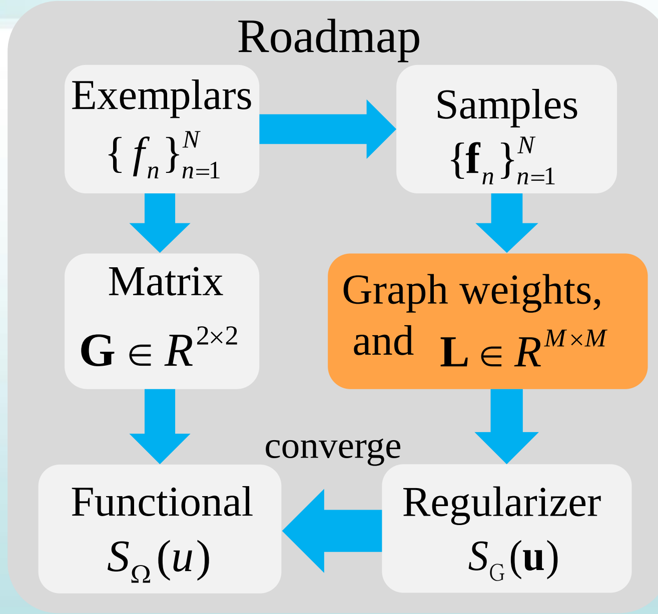
normalization factor γ

Thresholded Gaussian kernel

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

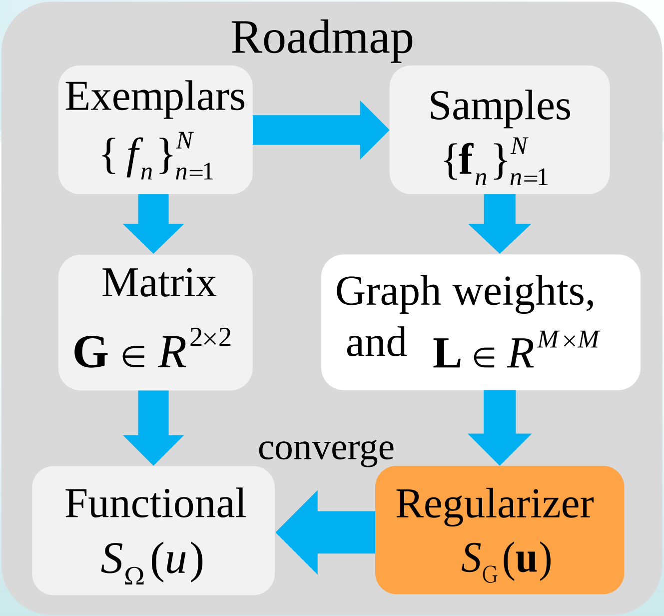
"Distance" between two vertices

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

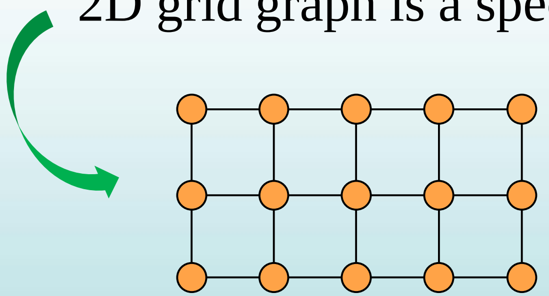


- G is an *r*-neighborhood graph:
  no edge connecting two vertices with distance greater than r

Graph Construction (IV)



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



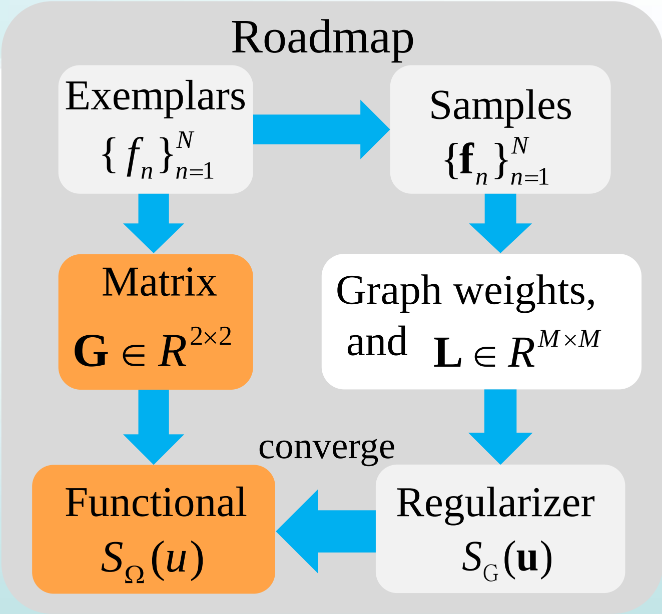
- $\mathbf{A}$ — (i, j)-th entry is w_{ij}
  $\mathbf{D}$ — diagonal entry is $\sum_{j=1}^{m} w_{ij}$
  } unnormalized Graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$

- $u(x, y): \Omega \to R$ is a continuous image

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

- $S_G(\mathbf{u}) = \mathbf{u}^{\mathrm{T}}\mathbf{L}\mathbf{u}$ — graph Laplacian regularizer

Convergence of the Graph Laplacian Regularizer (I)



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

$$S_\Omega(u) = \int_\Omega \nabla u^T \mathbf{G}^{-1} \nabla u \left(\sqrt{\det \mathbf{G}}\right)^{2\gamma-1} d\mathbf{s}$$

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

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

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

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)

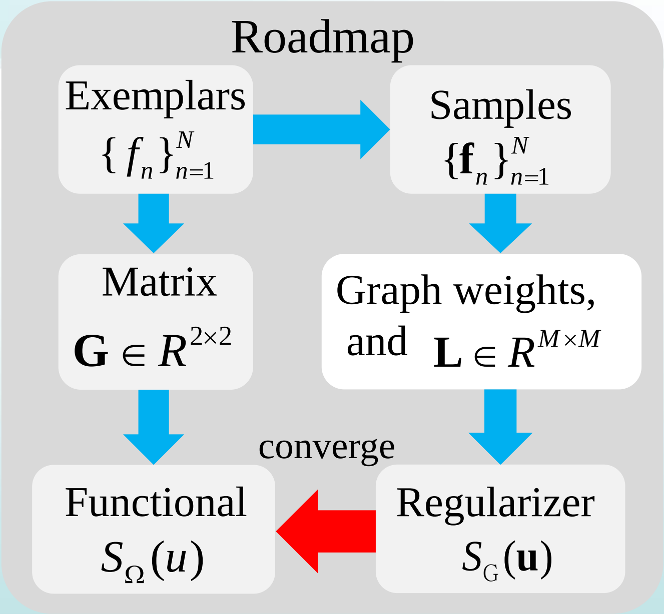
- ***Theorem :*** *convergence of* S_{G} *to* S_{Ω}

$$\lim_{\substack{M\to\infty \\ r\to 0}} S_{\mathrm{G}}(\mathbf{u}) \sim S_{\Omega}(u)$$

1) number of samples M increases

2) neighborhood r shrinks

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



- 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	$\mathbf{s}_i = (x_i, y_i)$	$\mathbf{v}_i = [\mathbf{f}_1(i)\ \mathbf{f}_2(i) \ldots \mathbf{f}_N(i)]^{\mathrm{T}}$

[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…

$$S_{\Omega}(u) = \int_{\Omega} \nabla u^{\mathrm{T}} \mathbf{G}^{-1} \nabla u \left(\sqrt{\det \mathbf{G}} \right)^{2\gamma - 1} ds$$

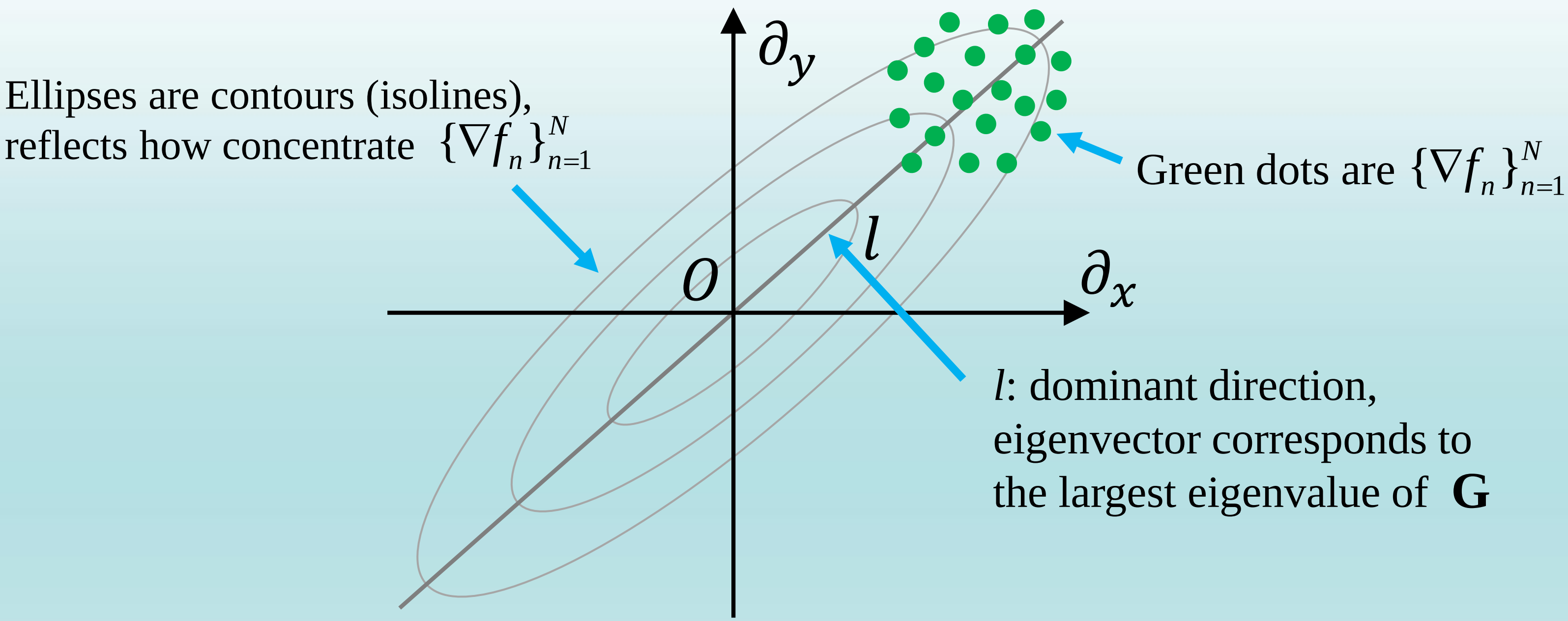
$$\mathbf{G} = \sum_{n=1}^{N} \nabla f_n \cdot \nabla f_n^{\mathrm{T}}$$

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

- 3 observations:
  - $\nabla u^{\mathrm{T}} \mathbf{G}^{-1} \nabla u$ measures length of ∇u 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 norm of gradient

Justification of Graph Laplacian Regularizer (II)

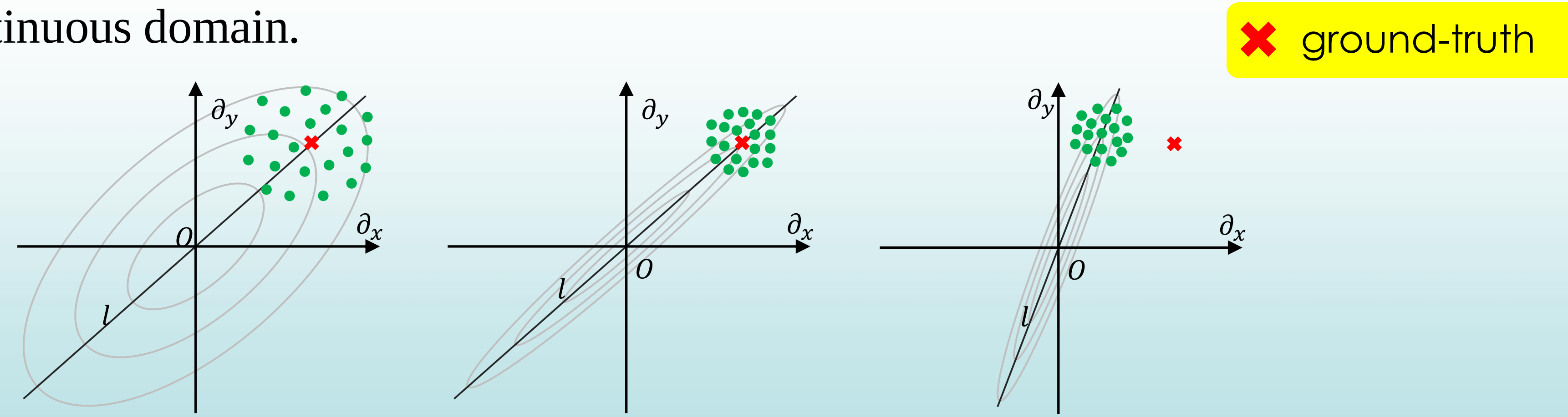
- Metric space defined by $\mathbf{G}$?
  - At a certain location (x, y) on the image



$$S_\Omega(u) = \int_\Omega \nabla u^{\mathrm{T}} \mathbf{G}^{-1} \nabla u \left(\sqrt{\det \mathbf{G}} \right)^{2\gamma - 1} ds \qquad \mathbf{G} = \sum_{n=1}^{N} \nabla f_n \cdot \nabla f_n^{\mathrm{T}}$$

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.



(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 $\mathbf{z}_0 \in R^M$, identify $K-1$ similar patches on the noisy image, the K patches $\{\mathbf{z}_k\}_{k=0}^{K-1}$ form a *cluster*
- On patch $\mathbf{z}_k$, gradient at pixel i is $\mathbf{g}_k^{(i)}$.
- Drop superscript i, model the noisy gradients $\{\mathbf{g}_k\}_{k=0}^{K-1}$ as

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

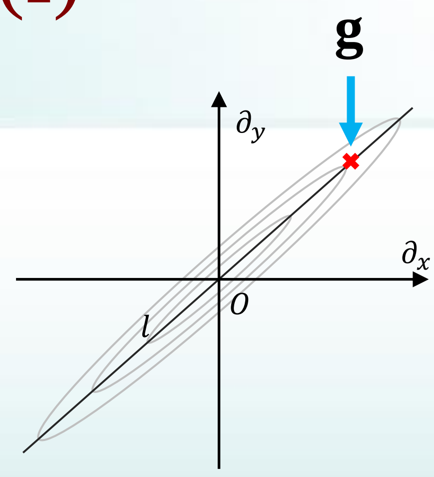
Unknown ground-truth

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

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

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

Seeking for the Optimal Metric Space (I)



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

$$\mathbf{G}_I(\mathbf{g}) = \mathbf{g}\mathbf{g}^{\mathrm{T}} + \beta\mathbf{I}$$

It is discriminant to $\mathbf{g}$

$\beta > 0$, smaller β makes the space more skewed

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

$$\mathrm{G}^* = \arg\min_{\mathrm{G}} \int_{\mathrm{R}^2} \left\| \mathrm{G} - \mathrm{G}_I(\mathrm{g}) \right\|_F^2 \cdot \Pr\left(\mathrm{g} \mid \{\mathrm{g}_k\}_{k=0}^{K-1}\right) d\mathrm{g}$$

$$\mathrm{G}^* = \int_{\mathrm{R}^2} \mathrm{G}_I(\mathrm{g}) \cdot \Pr\left(\mathrm{g} \mid \{\mathrm{g}_k\}_{k=0}^{K-1}\right) d\mathrm{g} \qquad (1)$$

Seeking for the Optimal Metric Space (II)

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

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

where the "ensemble" mean $\mathbf{g}$ and variance $\tilde{\sigma}^2$ are

noise variance of $\mathbf{g}_k$

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

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

$$\mathbf{G}^{\cdot} = \mathbf{g}\mathbf{g}^{\mathrm{T}} + (\tilde{\sigma}^2+\beta)\mathbf{I} \qquad (2)$$

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

From Metric Space to Graph Laplacian

- The structure of $\mathbf{G}^{\bullet} = \mathbf{g}\mathbf{g}^{\mathrm{T}} + (\tilde{\sigma}^2 + \beta)\mathbf{I}$ allows us to assign $N = 3$ exemplar functions, such that they lead to the optimal metric space:

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

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

- $\mathbf{f}_1^{\bullet}(i)$ and $\mathbf{f}_2^{\bullet}(i)$ correspond to the term $(\tilde{\sigma}^2 + \beta)\mathbf{I}$ in $\mathbf{G}^{\bullet}$
- $\mathbf{f}_3^{\bullet}(i)$ leads to the term $\mathbf{g}\mathbf{g}^{\mathrm{T}}$ in $\mathbf{G}^{\bullet}$
- With $\left\{\mathbf{f}_i^{\bullet}\right\}_{i=1}^{3}$, compute the optimal graph Laplacian

Graph Laplacian Regularization as Anisotropic Diffusion

- Our denoising problem is $\mathbf{u}^{\cdot} = \arg\min_{\mathbf{u}} \|\mathbf{z} - \mathbf{u}\|_2^2 + \tau \cdot \mathbf{u}^{\mathrm{T}} \mathbf{L} \mathbf{u}$
- Continuous counterpart is $u^{\cdot} = \arg\min_{u} \|u - z_0\|_{\Omega}^2 + \tau \cdot \int_{\Omega} \nabla u^{\mathrm{T}} \mathbf{D} \nabla u\, ds$

  where $\mathbf{D} = \mathbf{G}^{-1} \left(\sqrt{\det \mathbf{G}} \right)^{2\gamma - 1}$
- Differentiate with respect to u and equate it to zero

$$u^* = z_0 + \tau \operatorname{div}\left(\mathbf{D} \nabla u^*\right)$$

- Equivalent to marching z_0 forward in time using tensor diffusion

$$\partial_t u = \operatorname{div}\left(\mathbf{D} \nabla u\right), \; u(\mathbf{s}, t = 0) = z_0(\mathbf{s})$$

  with step size τ and tensor diffusivity $\mathbf{D}$

Graph Laplacian Regularization as P-M Diffusion (I)

- Assuming small noise, with $\mathbf{G}^{\cdot} = \mathbf{g}\mathbf{g}^{\mathrm{T}} + (\tilde{\sigma}^2 + \beta)\mathbf{I}$, tensor diffusion is simplified as

$$\partial_t u = \beta_{\mathrm{G}}^2 \mathrm{div}\left(\lambda(\|\nabla u\|_2)\nabla u\right)$$

where diffusivity is a scalar and constant

$$\lambda(\|\nabla u\|_2) = \left(1 + \frac{\|\nabla u\|_2^2}{\beta_{\mathrm{G}}^2}\right)^{\gamma-1.5} \qquad \beta_{\mathrm{G}} = \tilde{\sigma}^2 + \beta$$

It is called the Perona-Malik diffusion

- Denote $J(\|\nabla u\|_2) = \lambda(\|\nabla u\|_2)\|\nabla u\|_2$, diffusion scheme rewritten as

$$\partial_t u = \beta_{\mathrm{G}}^2\left(\lambda(\|\nabla u\|_2)\partial_{\zeta\zeta} u + J'(\|\nabla u\|_2)\partial_{\eta\eta} u\right)$$

decomposed as two independent diffusion processes.

Graph Laplacian Regularization as P-M Diffusion (II)

- A closer look..

ζ : direction perpendicular to ∇u η : direction parallel to ∇u

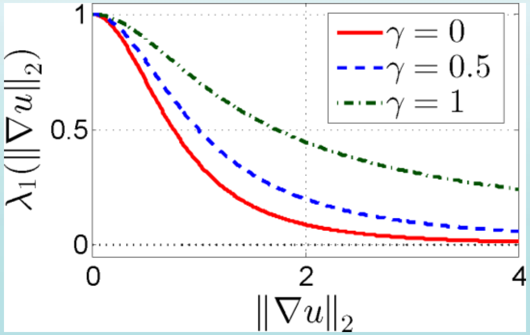
Denote a diffusion process along edges

Denote a diffusion process across edges

$$\partial_t u = \beta_G^2 \left(\lambda(\|\nabla u\|_2)\partial_{\zeta\zeta} u + J'(\|\nabla u\|_2)\partial_{\eta\eta} u \right)$$

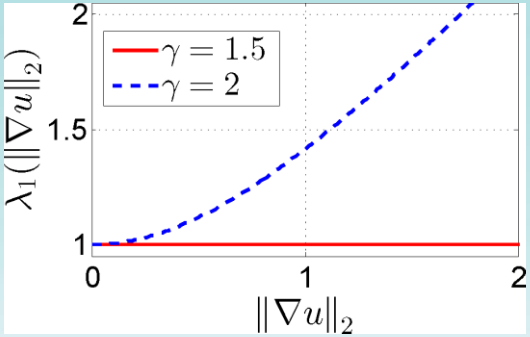
Diffusivity along edges

Diffusivity across edges



- Along edges: forward diffusion, *i.e.*, smoothing, as

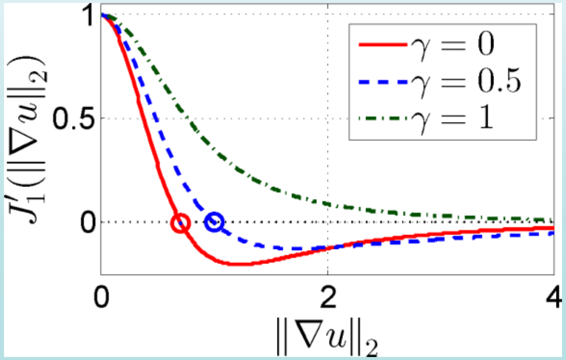
$$\lambda(\|\nabla u\|_2) = \left(1 + \frac{\|\nabla u\|_2^2}{\beta_G^2}\right)^{\gamma-1.5} > 0$$



Graph Laplacian Regularization as P-M Diffusion (III)

- Across edges: forward-backward diffusion, define $T = \beta_G / \sqrt{2(1-\gamma)}$

$$\begin{cases} \text{If } \gamma < 1 \begin{cases} \|\nabla u\|_2 < T \Rightarrow J' > 0\text{, forward diffusion} \\ \|\nabla u\|_2 = T \Rightarrow J' = 0\text{, no diffusion} \\ \|\nabla u\|_2 > T \Rightarrow J' < 0\text{, backward diffusion (sharpening)} \end{cases} \\ \text{If } \gamma = 1\text{, becomes a discretization of TV} \\ \text{If } \gamma > 1\text{, forward diffusion to smooth the image} \end{cases}$$

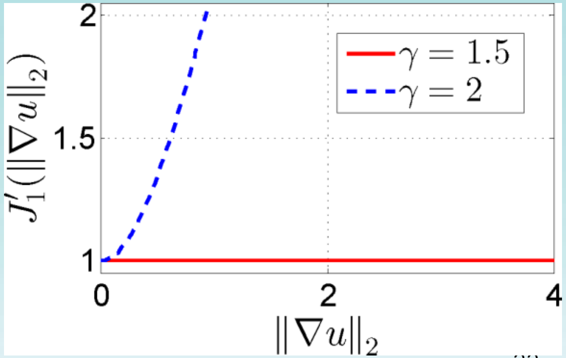


- Not only smooth but also sharpen images→PWS

Diffusivity across edges $J_1'(\|\nabla u\|_2) = \left(1 + \frac{\|\nabla u\|_2^2}{\beta_G^2}\right)^{\gamma-2.5} \cdot \left(1 + \frac{2\|\nabla u\|_2^2}{\beta_G^2}(\gamma - 1)\right)$

$$\partial_t u = \beta_G^2 \left(\lambda(\|\nabla u\|_2)\partial_{\zeta\zeta} u + J'(\|\nabla u\|_2)\partial_{\eta\eta} u\right)$$

Denote a diffusion process across edges



Experimentation (I)

- We develop an iterative patch-based algorithm
  - Optimal Graph Laplacian Regularization (OGLR) for denoising
    - Step 1: Search for similar patches
    - Step 2: Compute the optimal graph Laplacian
    - Step 3: Patch-based denoising
    - Step 4: Denoised image aggregation
- Corruption model: i.i.d. Additive White Gaussian Noise (AWGN)
- Measurements: PSNR (in dB), SSIM
- Natural images..
  - Test images: *Lena*, *Barbara*, *Peppers*, *Mandrill*
  - Compared with state-of-the-arts: K-SVD, BM3D, PLOW

Experimentation (II)

- Objective results

K-SVD: PSNR	BM3D: PSNR
K-SVD: SSIM	BM3D: SSIM
PLOW: PSNR	OGLR: PSNR
PLOW: SSIM	OGLR: SSIM

Image	Standard Deviation $\sigma_{\mathcal{I}}$													
	10		20		30		40		50		60		70	
Lena	35.55	**35.89**	32.40	**33.02**	30.42	**31.23**	28.96	29.82	27.80	**29.00**	26.87	**28.20**	26.11	**27.50**
	0.910	**0.915**	0.862	**0.876**	0.823	**0.843**	0.790	0.813	0.759	**0.796**	0.732	**0.776**	0.707	**0.756**
	35.28	35.62	32.70	32.93	31.11	31.22	29.79	**30.06**	28.72	28.86	27.92	28.19	27.09	27.46
	0.906	0.912	0.871	0.874	0.842	0.842	0.809	**0.821**	0.776	0.785	0.752	0.768	0.719	0.742
Barbara	34.54	**34.96**	30.89	**31.75**	28.56	**29.79**	26.87	28.00	25.45	27.23	24.23	26.30	23.32	25.51
	0.936	**0.942**	0.881	**0.905**	0.821	**0.867**	0.767	0.822	0.714	0.794	0.662	0.759	0.617	0.727
	33.79	34.46	30.97	31.45	29.41	29.63	28.11	**28.31**	26.98	**27.36**	26.06	**26.42**	25.25	**25.62**
	0.928	0.937	0.892	0.902	0.860	**0.867**	0.823	**0.838**	0.783	**0.801**	0.746	**0.768**	0.710	**0.734**
Peppers	34.83	**35.02**	32.31	**32.75**	30.64	**31.23**	29.31	29.93	28.09	**29.09**	27.03	**28.26**	26.14	**27.54**
	0.879	**0.879**	0.839	**0.845**	0.811	**0.820**	0.786	0.795	0.762	**0.782**	0.738	**0.763**	0.715	**0.746**
	34.40	34.91	32.40	32.67	31.01	**31.23**	29.80	**30.10**	28.76	28.83	27.86	28.20	27.17	27.42
	0.870	**0.879**	0.840	0.842	0.815	0.818	0.789	**0.798**	0.760	0.762	0.732	0.751	0.713	0.729
Mandrill	30.39	**30.58**	26.36	**26.60**	24.30	**24.56**	22.92	23.09	21.92	22.35	21.20	21.74	20.71	21.28
	0.895	**0.897**	0.778	**0.792**	0.675	0.702	0.582	0.617	0.503	0.549	0.443	0.498	0.401	0.459
	29.58	29.84	26.10	26.35	24.33	**24.56**	23.18	**23.40**	22.41	**22.59**	21.81	**21.99**	21.33	**21.47**
	0.853	0.883	0.761	0.786	0.681	**0.706**	0.612	**0.650**	0.559	**0.595**	0.510	**0.546**	0.468	**0.500**

0.3 dB better than BM3D!

Experimentation (III)

- Subjective comparisons ($\sigma_I = 40$)

Original

Noisy, 16.48 dB

K-SVD, 26.84 dB

BM3D, 27.99 dB

PLOW, 28.11 dB

OGLR, 28.35 dB

Experimentation (IV)

- Piecewise smooth images..
  - Test images: *Cones*, *Teddy*, *Art*, *Moebius*, *Aloe*
  - Compared with state-of-the-arts: BM3D, NLGBT
- Objective results

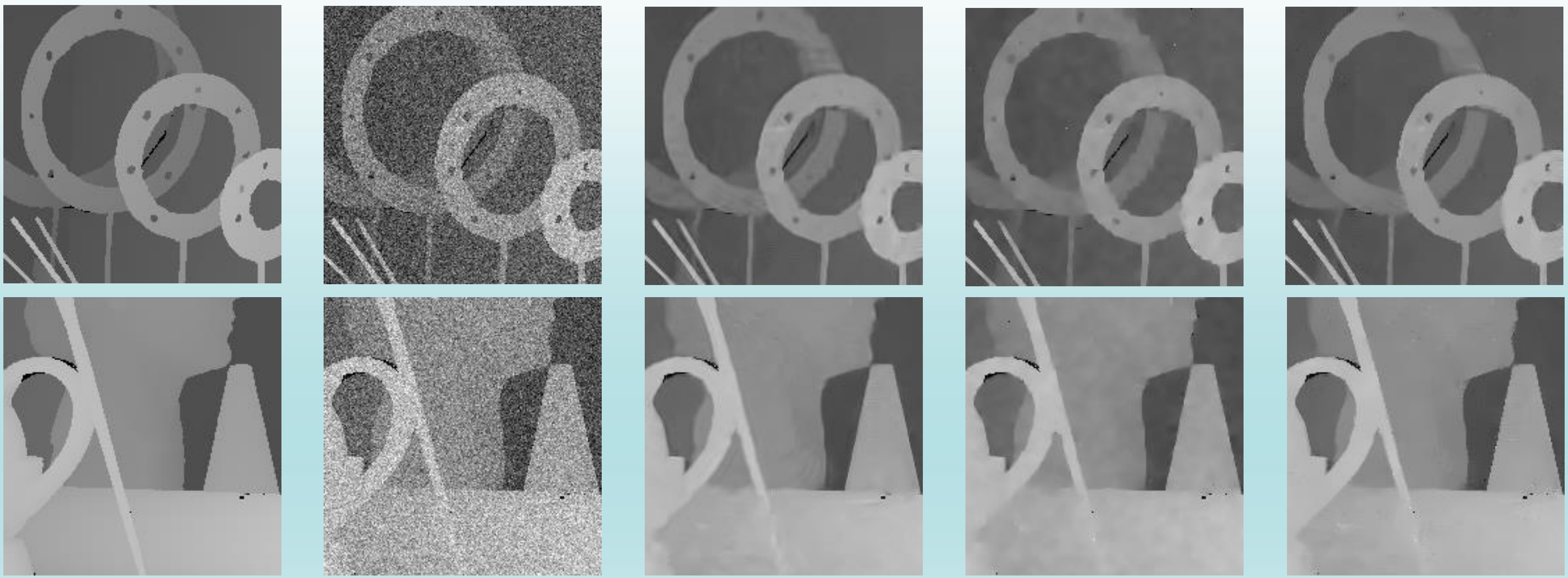
BM3D: PSNR	NLGBT: PSNR	OGLR: PSNR
BM3D: SSIM	NLGBT: SSIM	OGLR: SSIM

Image	Standard Deviation $\sigma_{\mathcal{I}}$														
	10			20			30			40			50		
Cones	40.40 0.983	42.19 **0.987**	**42.93** **0.987**	35.17 0.960	36.63 0.966	**37.39** **0.968**	32.57 0.935	33.45 0.942	**34.08** **0.944**	31.01 0.912	31.36 **0.926**	**31.78** 0.922	29.62 0.898	30.01 **0.913**	**30.36** 0.900
Teddy	41.17 0.985	41.80 0.985	**42.80** **0.986**	35.94 0.967	36.84 **0.968**	**37.73** **0.968**	33.16 0.948	33.85 **0.949**	**34.52** 0.947	31.32 0.927	31.65 **0.937**	**32.20** 0.929	29.73 0.919	30.26 **0.928**	**30.70** 0.910
Art	40.04 0.983	41.34 0.986	**42.98** **0.988**	35.47 0.959	36.13 0.963	**37.33** **0.967**	33.21 0.934	33.36 0.937	**34.27** **0.944**	31.60 0.907	31.61 0.920	**32.15** **0.922**	30.36 0.891	30.45 **0.906**	**30.82** 0.898
Moebius	42.03 0.983	42.58 0.984	**43.31** **0.985**	37.15 **0.962**	37.63 **0.962**	**38.36** **0.962**	34.70 **0.940**	34.89 **0.940**	**35.35** 0.938	33.09 0.918	33.13 **0.929**	**33.19** 0.917	31.75 0.911	**31.98** **0.922**	31.94 0.898
Aloe	40.30 0.984	41.37 0.986	**42.86** **0.988**	35.66 0.962	36.25 0.965	**37.47** **0.968**	33.31 0.938	33.45 0.941	**34.53** **0.946**	31.73 0.913	31.68 0.925	**32.56** **0.928**	30.58 0.899	30.62 **0.913**	**31.18** 0.907

1.6 dB better than NLGBT!

Experimentation (V)

- Subjective comparisons ($\sigma_I = 30$)



Original | Noisy, 18.66 dB | BM3D, 33.26 dB | NLGBT, 33.41dB | OGLR, 34.32 dB

Summary of Image Denoising via Graph Smoothness Prior

- Inverse imaging problems are ill-posed; we use graph Laplacian regularizer as image prior
- Graph Laplacian regularizer converges to a continuous functional, analysis of the functional explains the mechanisms and implications of graph Laplacian regularizer
- We describe a methodology to derive the optimal edge weights given nonlocal noisy gradient observations
- By interpreting graph Laplacian regularization as anisotropic diffusion, we show that it not only smooth images but may also sharpen images, promoting piecewise smooth results
- Our algorithm performs competitively with state-of-the-art methods for natural images, and out-perform them for piecewise smooth images

Outline (Part II)

- Image Restoration using GSP Tools
  - Image Denoising
    - Sparsity Prior
    - Smoothness Prior
  - Soft Decoding of JPEG Compressed Images