Gene Cheung, Xianming Liu
National Institute of Informatics
16th December, 2015

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
  - Joint Denoising / Contrast Enhancement

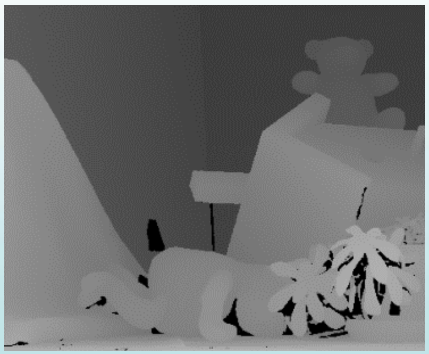
Outline (Part II)

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

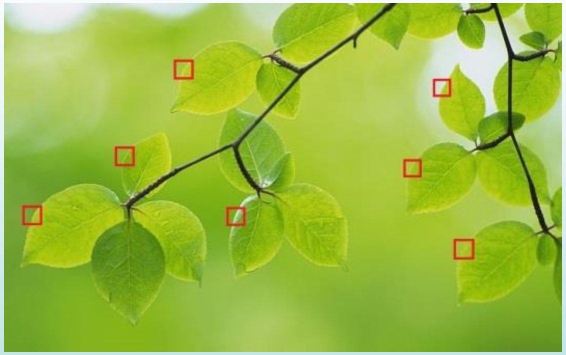
desired signal

observation $y = x + v$ noise

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

Existing Image Denoising Methods

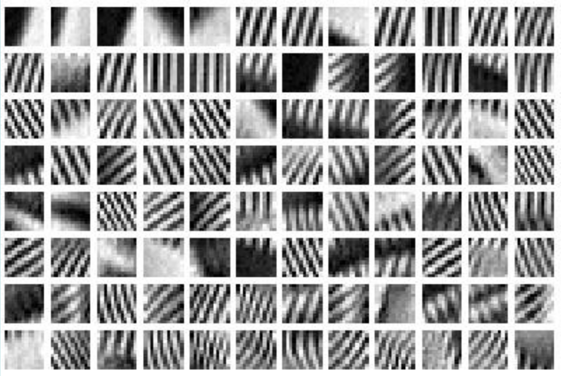
- Local methods (e.g., bilateral filtering)



- Nonlocal image denoising

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

  - Assumption: nonlocal self-similarity



- Dictionary learning based

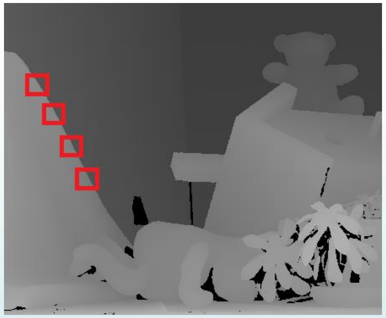
  Elad et al, "Image denoising via sparse and redundant representation over learned dictionaries," *TIP 2006.*

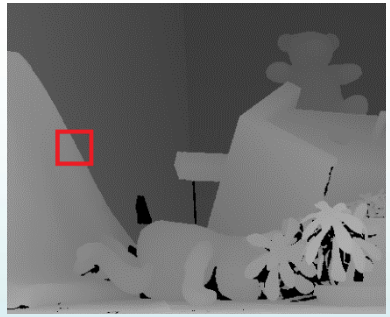
  - represent a signal by the linear combination of a few atoms out of a dictionary

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

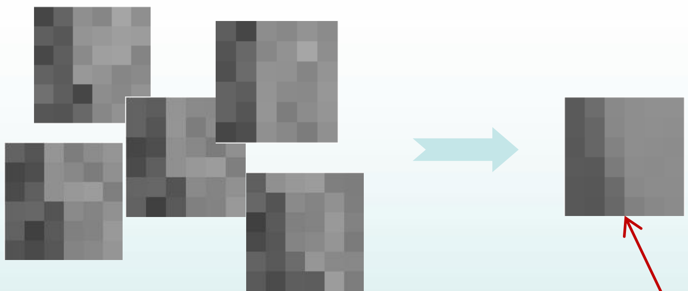
unify in GFT domain

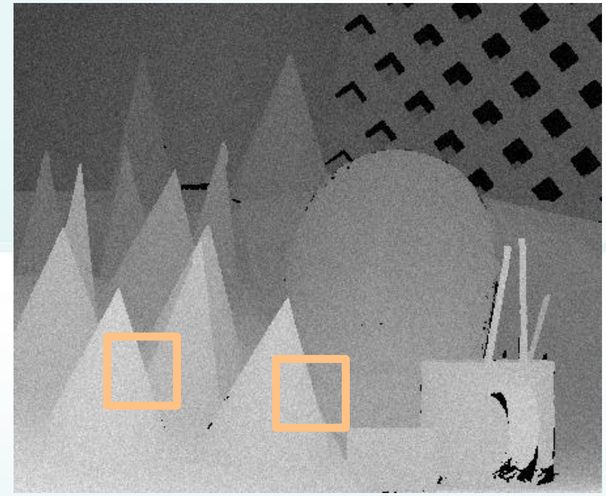
Challenges Our method

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



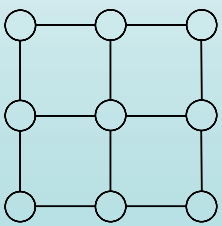


common GFT from avg. patch

observation i

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

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.

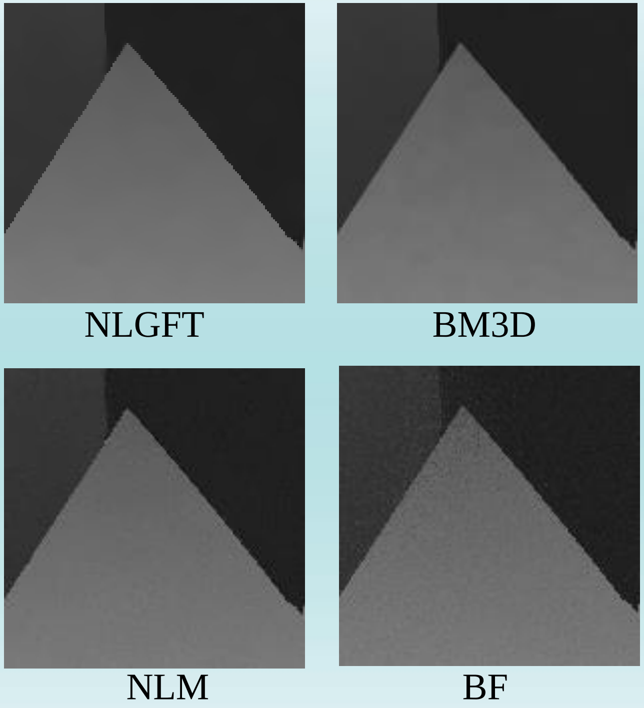
[4] G. Shen, W.-S. Kim, S.K. Narang, A. Ortega, J. Lee, and H. Wey, "**Edge-adaptive transforms for efficient depth map coding**," in *IEEE Picture Coding Symposium*, Nagoya, Japan, December 2010.

[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



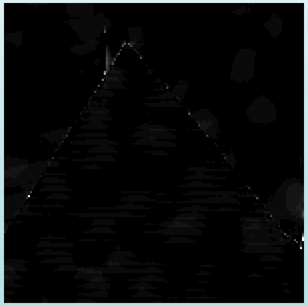
Experimental Results (2)

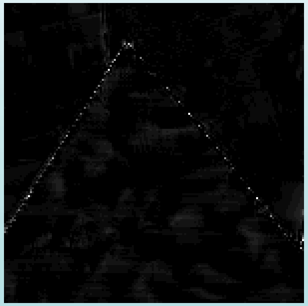
- Setup:

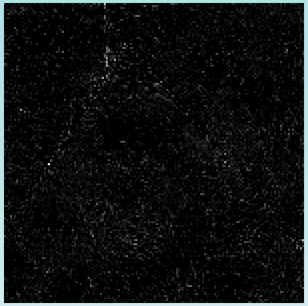
  - Test Middleburry depth maps: *Cones*, *Teddy, Sawtooth*
  - Add Additive White Gaussian Noise
  - Compare agaist Bilateral Filtering (BF), Non-Local Means Denoising (NLM) and Block-Matching 3D (BM3D)

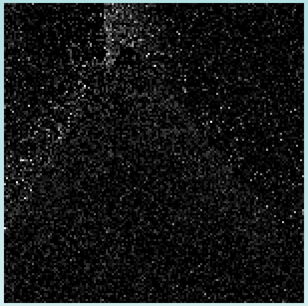
- Results
  - Up to 2.28dB improvement over BM3D.

		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


NLGFT


BM3D


NLM


BF

10

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:

$$\underbrace{\mathbf{y}}_{\text{observation}} = \underbrace{\mathbf{x}}_{\text{desired signal}} + \underbrace{\mathbf{e}}_{\text{noise}}$$

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

$$\min_{\mathbf{x}} \underbrace{\|\mathbf{y}-\mathbf{x}\|_2^2}_{\text{fidelity term}} + \lambda \underbrace{\text{prior}(\mathbf{x})}_{\text{prior term}}$$

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

$$S_G(\mathbf{x}) = \mathbf{x}^T \mathbf{L} \mathbf{x} \qquad \underbrace{\mathbf{L}}_{\text{graph Laplacian matrix}} = \mathbf{D} - \mathbf{A}$$

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

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

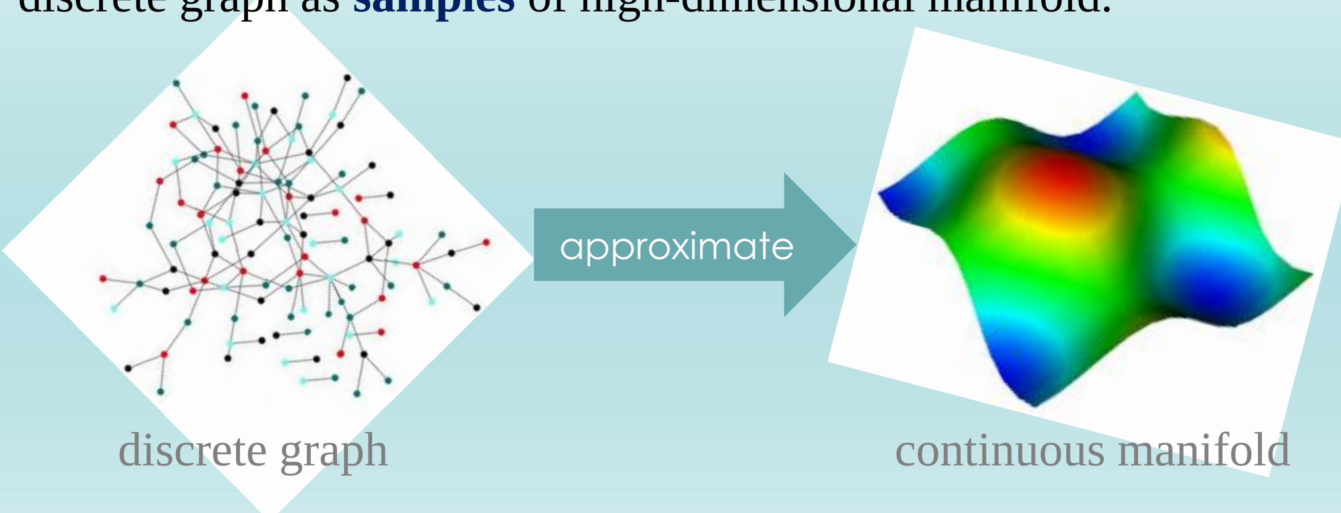
$\text{dist}(i, j)$ is some distance metric between pixels *i* and *j*

[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)

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

- However…
  a. Why is $S_G(\mathbf{x}) = \mathbf{x}^T \mathbf{L} \mathbf{x}$ a good prior?
  b. Why using Gaussian kernel for edge weights?
  c. How to design a discriminant $\mathbf{x}^T \mathbf{L} \mathbf{x}$ 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

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

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

   Graph Laplacian regularizer S_G $\xrightarrow{\text{converge}}$ A continuous functional S_Ω for regularization

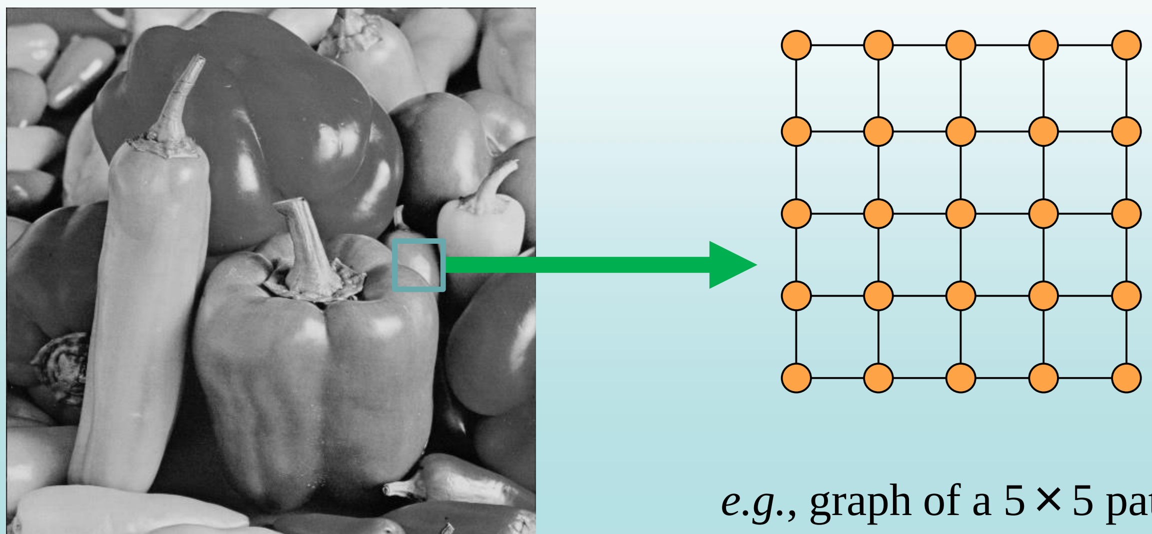
2. Analysis of functional S_Ω provides understanding of how signals are being discriminated and to what extent; careful graph construction leads to ***discriminant*** signal prior.

   Select edge weights via "feature functions" $\xrightarrow{\text{obtain}}$ 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

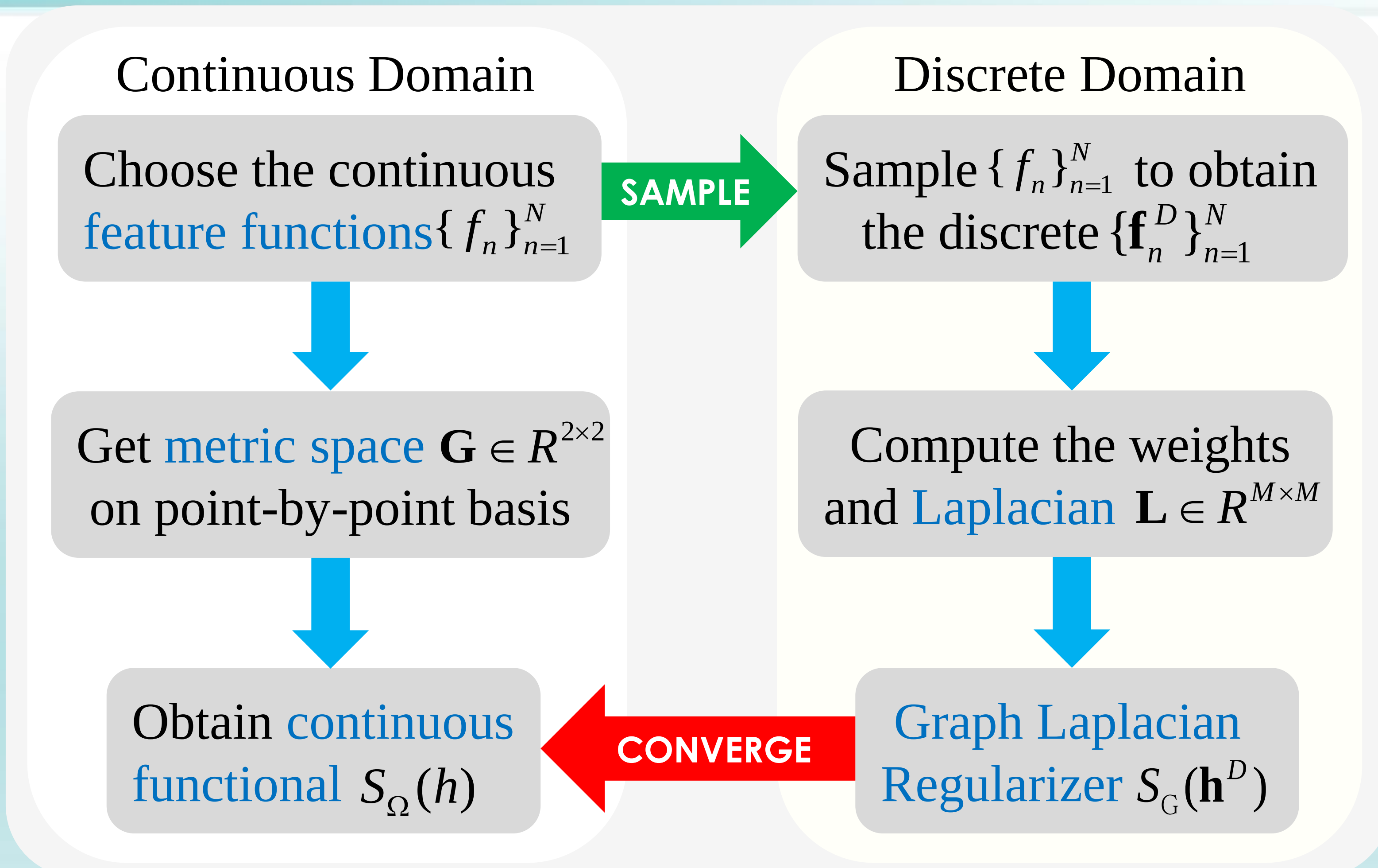
- Graph for image restoration
  - Each pixel corresponds to a vertex in a graph (denote # of pixels as M).



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).

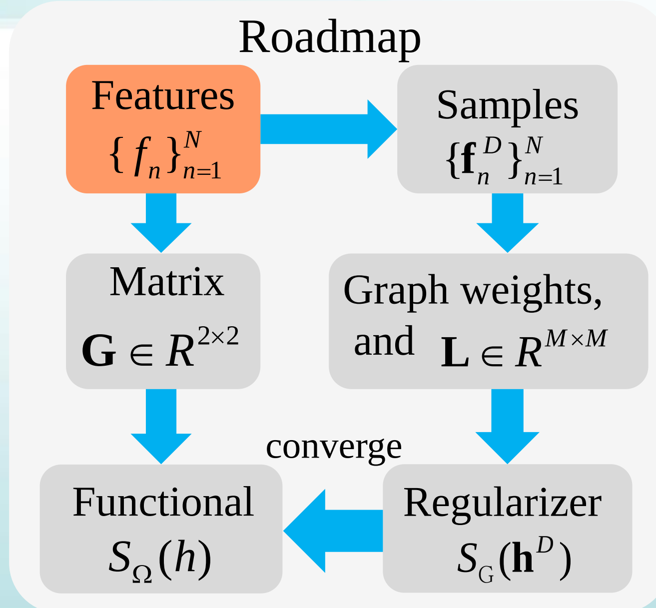
Road Map



- 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

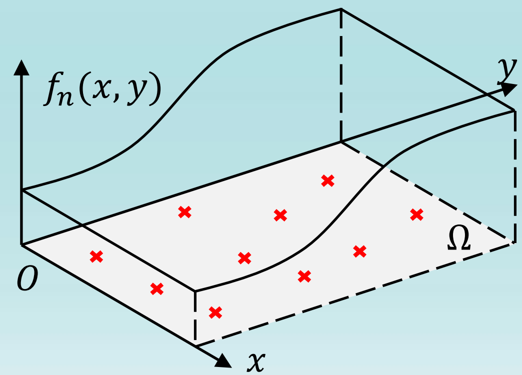


- (Freely) choose N continuous functions

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

called feature functions, for example

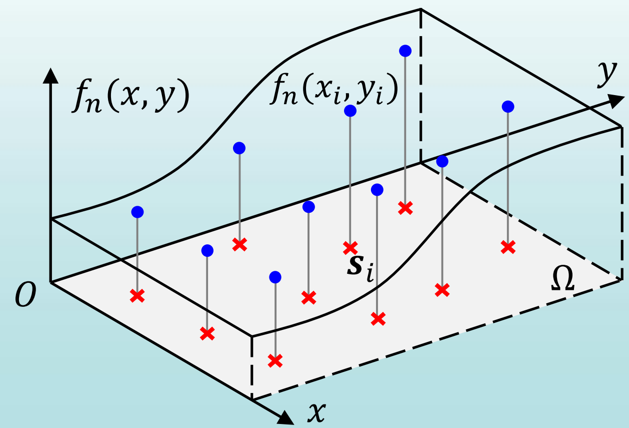
  - intensity for gray-scale image ($N = 1$)
  - R, G, B channels for color image ($N = 3$)

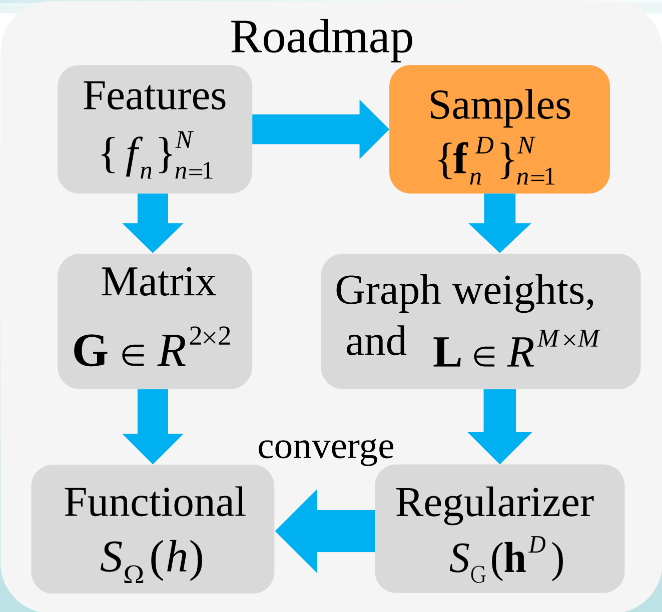


Graph Construction (II)

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

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





- 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

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

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

Roadmap

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

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

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

Matrix ↓ Functional $S_\Omega(h)$

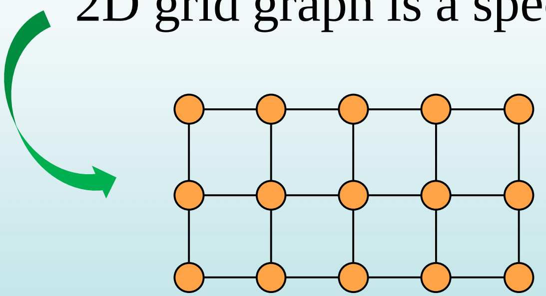
Graph weights ↓ Regularizer $S_G(\mathbf{h}^D)$

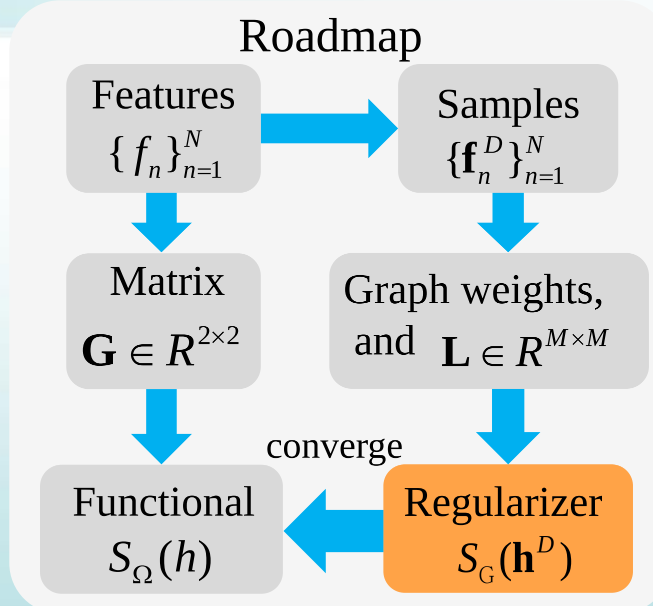
Regularizer → Functional: converge

- G is an *r*-neighborhood graph, *i.e.*, 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





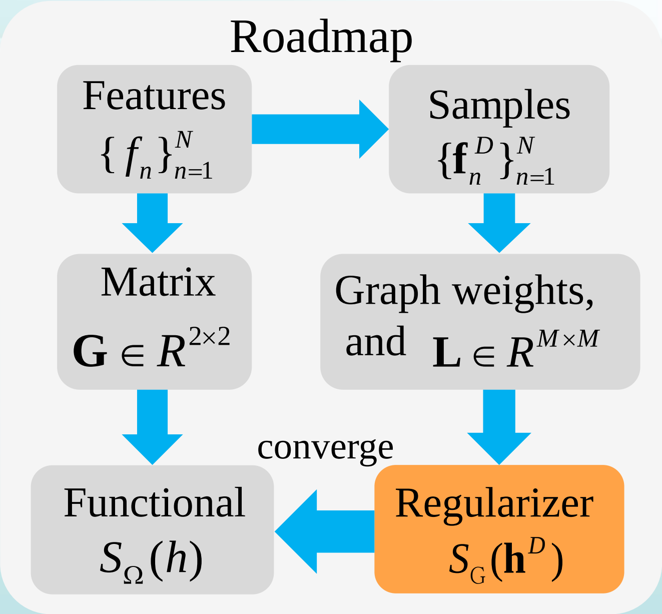
- $\mathbf{A}$ —(i, j) -th entry is w_{ij}
  $\mathbf{D}$ — diagonal entry is $\sum_{j=1}^{m} w_{ij}$
  } unnormalized Graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$

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

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

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

Convergence of the Graph Laplacian Regularizer (I)



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

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

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

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

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

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)

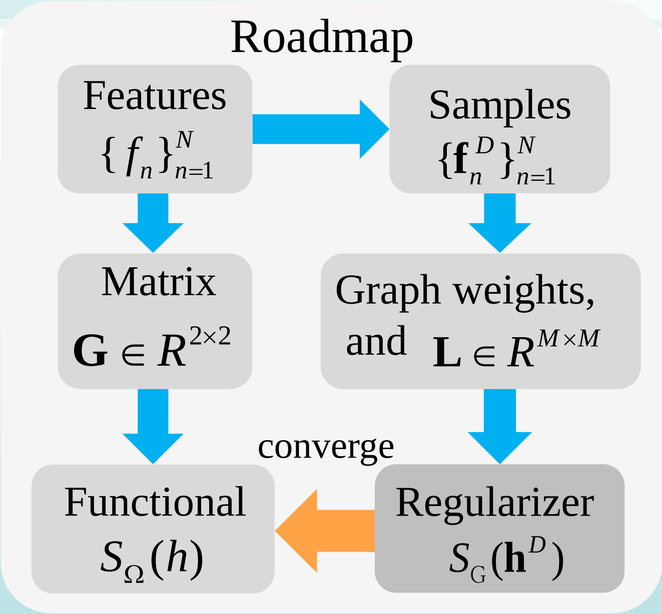
- ***Theorem :*** *convergence of* S_{G} *to* S_{Ω}

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

1) number of samples M increases

2) neighborhood $r = \varepsilon C_r$ shrinks

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



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

[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…

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

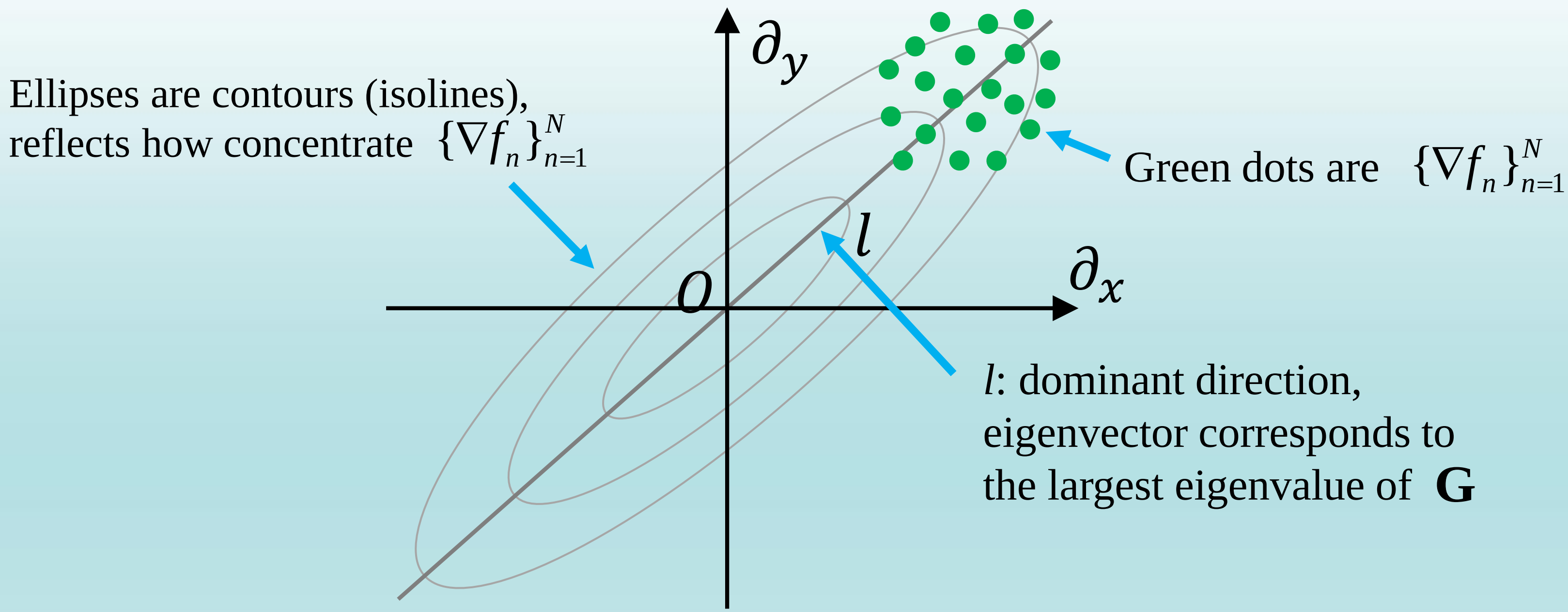
$$\mathbf{G} = \sum_{n=1}^{N} \nabla f_n \cdot (\nabla f_n)^{\mathrm{T}}$$

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

- 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)

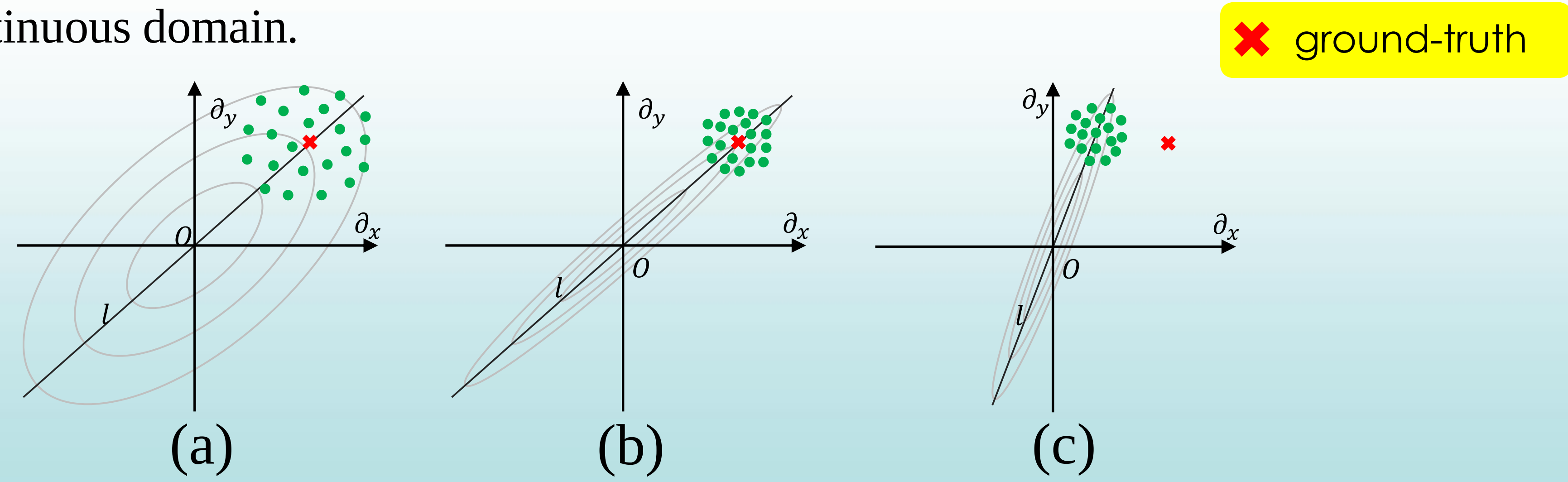
- Metric space defined by $\mathbf{G}$?
  - At a certain location (x, y) on the image



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

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.



  - 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

$$\underbrace{\mathbf{g}}_{\text{Unknown ground-truth}}\!\!{}_k = \mathbf{g} + \mathbf{e}_k, 0 \le k \le K-1$$

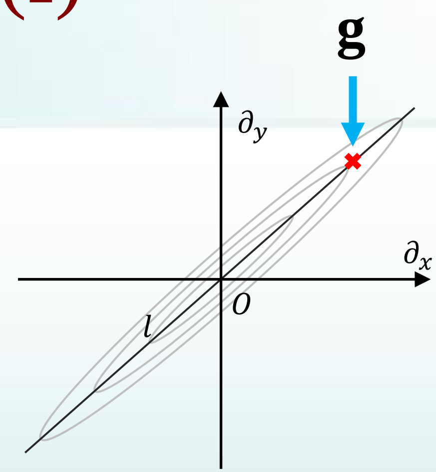
Unknown ground-truth: $\mathbf{g}$

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

- 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:

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

  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

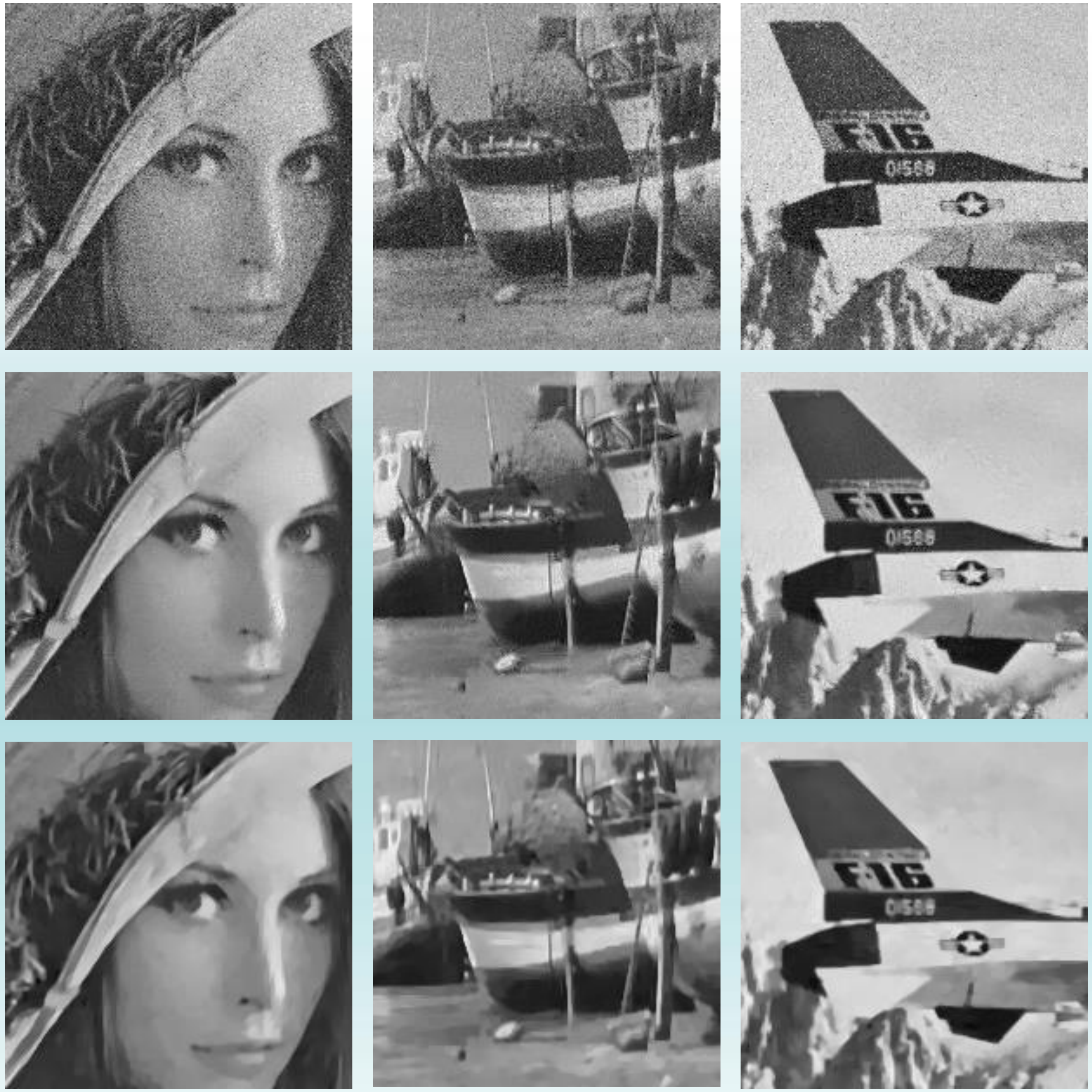
Experimental Results (II)

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

BF

NLM

OGLRD



Experimental Results (III)

- Some visual results when $\sigma_n = 30$

Summary of Image Denoising via Graph Smoothness Prior

- 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 (Part II)

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

Outline

- Compressed Image Restoration using GSP
  - Background
  - Related Work
  - The Proposed Algorithm
- Joint Image Denoising and Contrast Enhancement using GSP
  - Background
  - Related Work
  - The Proposed Algorithm

Outline

- Compressed Image Restoration using GSP
  - Background
  - Related Work
  - The Proposed Algorithm
- Joint Image Denoising and Contrast Enhancement using GSP
  - Backgroud
  - Related Work
  - The Proposed Algorithm

Outline

- Compressed Image Restoration using GSP
  - Background
  - Related Work
  - The Proposed Algorithm
- Joint Image Denoising and Contrast Enhancement using GSP
  - Backgroud
  - Related Work
  - The Proposed Algorithm

Background

- **Compressed image restoration: important and practical problem:**
  - Compression is the most common cause of image degradation.
  - Compression is indispensable in almost all visual communication systems.
- **Compressed image restoration is a non-trivial problem:**
  - Compression noises are signal-dependent.
  - Far from being white and independent.
  - Composite noises: blocking and ringing effects.

Background

- **DCT-based image/video compression**
  - JPEG, H. 264, HEVC

- **We consider JPEG-Compressed image restoration**
  - Most images from Internet are .JPG format

- **JPEG standard consists three steps:**
  - Perform DCT on each non-overlapping 8x8 blocks;
  - Perform quantization on DCT coefficients; → introduce compression noise
  - Lossless entropy coding.

Restoration of JPEG-compressed Images

➢ Problem Formulation

- Encoder:

DCT

$$q_i = round(\mathrm{Y}_i / \mathrm{Q}_i), \quad \mathbf{Y} = \mathbf{T}\mathbf{y}$$

8x8 pixel block

quantization parameter

DCT Coefficients

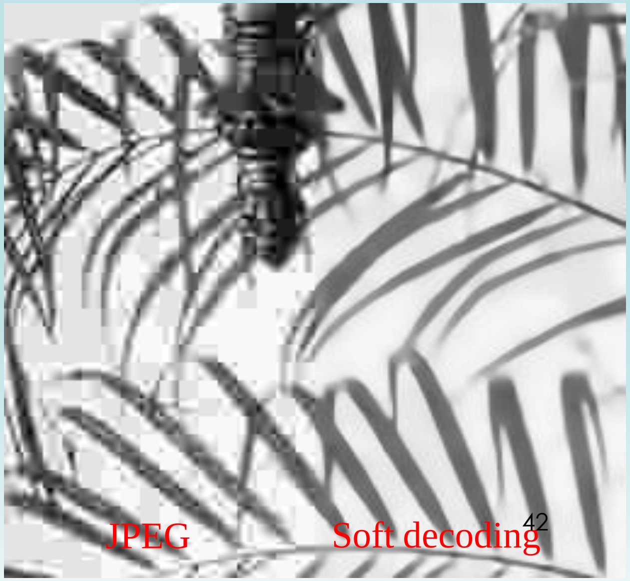
- Decoder:

$$\underbrace{q_i \mathrm{Q}_i \le \mathrm{Y}_i \le (q_i + 1)\mathrm{Q}_i, i = 1, 2, \cdots, 64.}$$

This is the strongest side information used for JPEG-image restoration.

Hard decoding vs. Soft decoding

- **Hard decoding**
  - Reconstruct DCT coefficients using the centers of assigned quantization bins.
- **Soft decoding**
  - Find the most probable signal WITHIN the set of quantization bin constraints.
  - Signal priors is used for aid
    - Local/non-local similarity
    - Total variation
    - Sparsity
    - Our method: combining sparsity and graph-signal smoothness prior



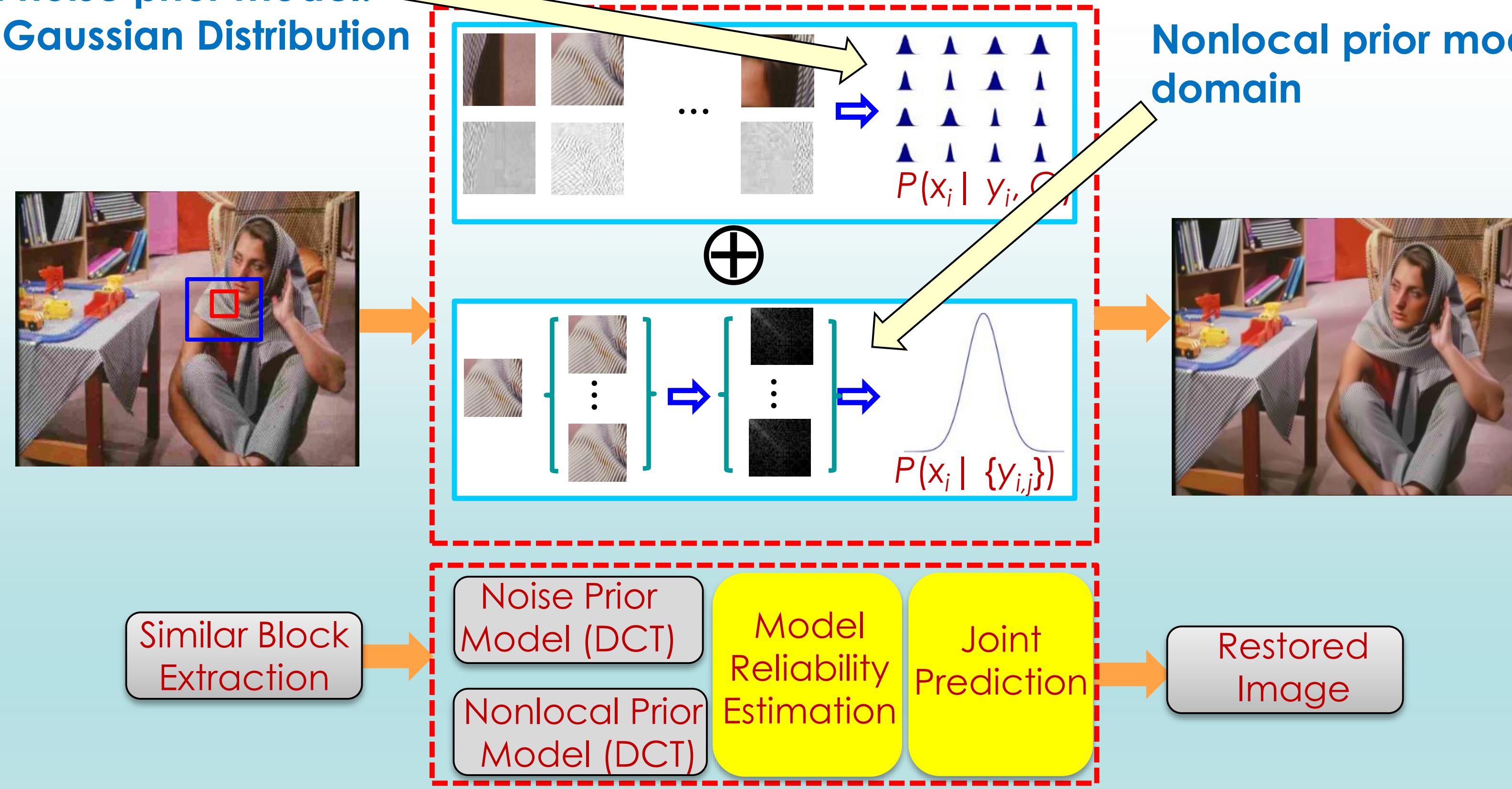
Outline

- Compressed Image Restoration using GSP
  - Background
  - Related Work
  - The Proposed Algorithm
- Joint Image Denoising and Contrast Enhancement using GSP
  - Background
  - Related Work
  - The Proposed Algorithm

Related Work (1): Non-local Similarity in DCT Domain

Compression noise prior model: Generalized Gaussian Distribution

Nonlocal prior model in DCT domain



$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \left(\|X_B - Y_B\|_{W_Q}^2 + \|X_B - N\{Y_B\}\|_{W_B}^2 \right)$$

- Xinfeng Zhang, et al. Compression Artifact Reduction by Overlapped-Block Transform Coefficient Estimation with Block Similarity", IEEE Trans. on Image Processing, vol. 22, no. 12, pp. 4613-4626, Dec 2013.

Related Work (2): Total-variation

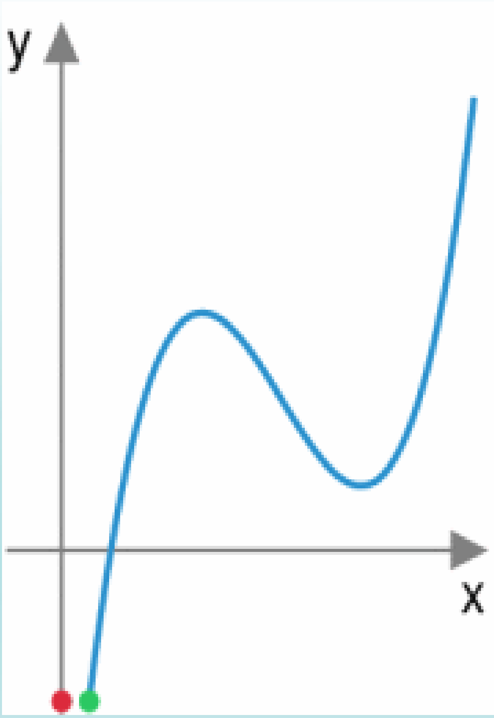
- **Assumption:** natural images are approximately piecewise constant.

$$TV(\mathbf{x}) := \int_{\Omega} |\nabla \mathbf{x}|$$

  - $\mathbf{x}$ is an image defined on the bounded, open and convex region Ω of $\mathbb{R}^2$ such that $\mathbf{x} \in L^1(\Omega)$

- **Discrete Form:**

$$TV(\mathbf{x}) := \sum_{k=1}^{N} \sqrt{([\nabla \mathbf{x}^{(x)}])_k^2 + ([\nabla \mathbf{x}^{(y)}])_k^2}$$



[from wiki]

- K. Bredies and M. Holler, "A Total-Variation-based JPEG decompression model," SIAM Journal on Imaging Sciences, vol. 5, no. 1, pp.366-393, 2012.

Related Work (2): Total-variation

➢ **TV-based JPEG-image restoration:**

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \left\{ \mathrm{TV}(\mathbf{x}) + \lambda F_U(\mathbf{x}) \right\}$$

- where

$$F_U(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in U \\ +\infty & \text{otherwise} \end{cases}$$

- U is the given image data set associated with the JPEG object.

- K. Bredies and M. Holler, "A Total-Variation-based JPEG decompres-sion model," SIAM Journal on Imaging Sciences, vol. 5, no. 1, pp.366-393.

Related Work (3): Sparsity-based Soft Decoding

- **Sparse signal representation prior:**

$$\mathbf{x} = \boldsymbol{\Phi}\boldsymbol{\alpha} + \varepsilon$$

  - The dictionary $\boldsymbol{\Phi}$ could be learned from the input JPEG image or offline training set.

- **Sparsity-based soft decoding:**

$$\min_{\{\mathbf{x},\boldsymbol{\alpha}\}} \|\mathbf{x} - \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2 + \lambda_1 \|\boldsymbol{\alpha}\|_1$$

$$s.t., \quad q_i \mathrm{Q}_i \le \mathrm{Y}_i \le (q_i + 1)\mathrm{Q}_i$$

  - Both quantities are unknown.
  - When quantization is coarse, any sparse solution within the large q-bins is equally good. → more prior information is required!

- C. Jung , L. Jiao, H. Qi, and T. Sun, "Image deblocking via sparse representation," Signal Processing: Image Communication, vol. 27, no. 6, pp. 663-677, 2012.

Joint Sparse Coding in Dual Domains

- **Restoration in pixel domain alone:**
  - Advantage : directly reconstruct visually significant features, such as edges, textures;
  - Disadvantage : IDCT will propagate the quantization error for single coefficient to the whole 8x8 patch.
- **Restoration in DCT domain alone:**
  - Advantage : avoid error propagation;
  - Disadvantage : the loss of high-frequency information make it impossible to recover edges and fine textures.
- **Strategy**
  - Complement the advantages of dual domains;
  - Joint sparse representation in dual domains.

- Xianming Liu, Xiaolin Wu, Jiantao Zhou, Debin Zhao, Data-driven Sparsity-based Restoration of JPEG-compressed Images in Dual Transform-Pixel Domain, CVPR 2015, Boston, June 2015.

Joint Sparse Coding in Dual Domains

- **Joint Sparse Coding in Dual Domains :**

The processing DCT patch

Dictionary in DCT domain

Inverse DCT

Dictionary in pixel domain

$$\arg\min_{\{\boldsymbol{\alpha},\boldsymbol{\beta}\}} \left\{ \left\| \mathbf{y}_0 - \boldsymbol{\Phi}\boldsymbol{\alpha} \right\|_2^2 + \lambda_1 \left\| \boldsymbol{\alpha} \right\|_1 + \lambda_2 \left\| \mathbf{T}^{-1}\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{\Psi}\boldsymbol{\beta} \right\|_2^2 + \lambda_3 \left\| \boldsymbol{\beta} \right\|_1 \right\}$$

$$s.t., \mathbf{q}^L \prec \boldsymbol{\Phi}\boldsymbol{\alpha} \prec \mathbf{q}^U.$$

- **Optimization**

$$\arg\min_{\boldsymbol{\theta}} \left\| \mathbf{y}_0 - \mathbf{D}\boldsymbol{\theta} \right\|_2^2 + \lambda \left\| \boldsymbol{\theta} \right\|_1$$

$$s.t., [\mathbf{q}^L\ \mathbf{0}] \prec \mathbf{D}\boldsymbol{\theta} \prec [\mathbf{q}^U\ \mathbf{0}].$$

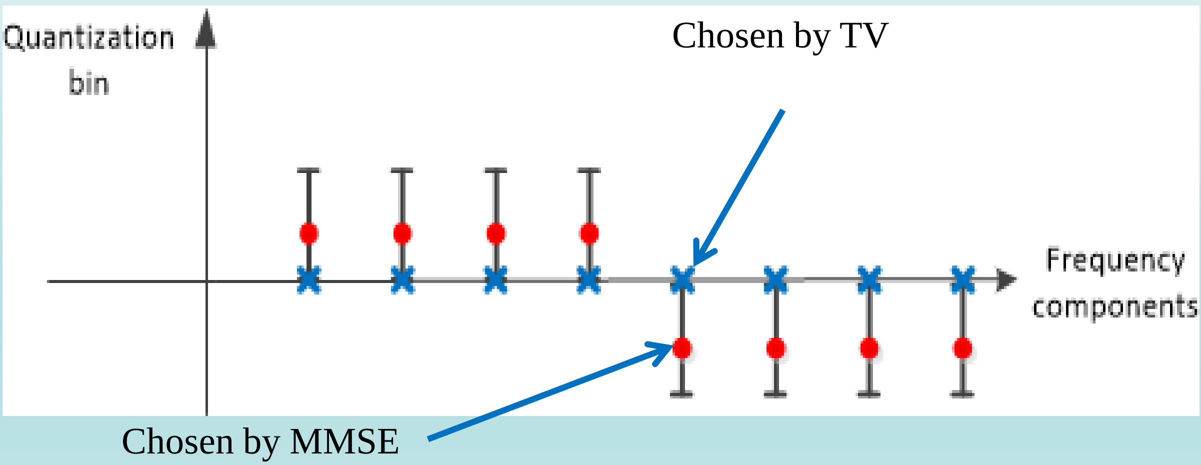
- Xianming Liu, Xiaolin Wu, Jiantao Zhou, Debin Zhao, Data-driven Sparsity-based Restoration of JPEG-compressed Images in Dual Transform-Pixel Domain, CVPR 2015, Boston, June 2015.

Sparsity + Total Variation

➢ Total-variation as a regularizer

- TV norm would promote reconstructed coefficients closest to the zero q-bin boundaries, resulting in an almost DC signal.

→ over-smoothing!



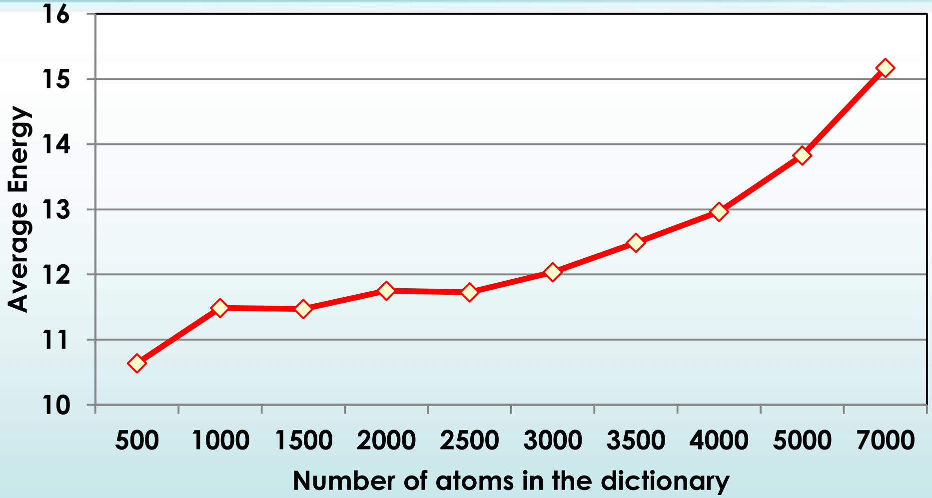
$$\min_{\{\mathbf{x},\boldsymbol{\alpha}\}}\left\{\left\|\mathbf{x}-\boldsymbol{\Phi}\boldsymbol{\alpha}\right\|_2^2+\lambda\left\|\boldsymbol{\alpha}\right\|_1\right\}+\gamma\mathrm{TV}(\mathbf{x})+F_U(\mathbf{x})$$

- Huibin Chang, M.K. Ng, and Tieyong Zeng, "Reducing artifacts in jpeg decompression via a learned dictionary," IEEE Transactions on Signal Processing, vol. 62, no. 3, pp. 718–728, Feb 2014.

Outline

- Compressed Image Restoration using GSP
  - Background
  - Related Work
  - The Proposed Algorithm
- Joint Image Denoising and Contrast Enhancement using GSP
  - Background
  - Related Work
  - The Proposed Algorithm

Key Observation



- **AE of AC components of atoms:**

$$\mathrm{AE} = \sum_{i=2}^{n} \lambda_i \alpha_i^2$$

α_i : DCT coefficients of an atom

λ_i : The corresponding DCT basis

- **Average Energy (AE):** a measurement of high-frequency information preserved by atoms in the offline learned dictionary.

- **Our observation:** AE almost increases progressively with the size of KSVD-trained dictionary.

- Xianming Liu, Gene Chueng, Xiaolin Wu, Debin Zhao, Soft Decoding of JPEG Compressed Images with Sparsity and Graph-Signal Smoothness Priors, submitted to IEEE Transactions on Image Processing.

Key Motivation

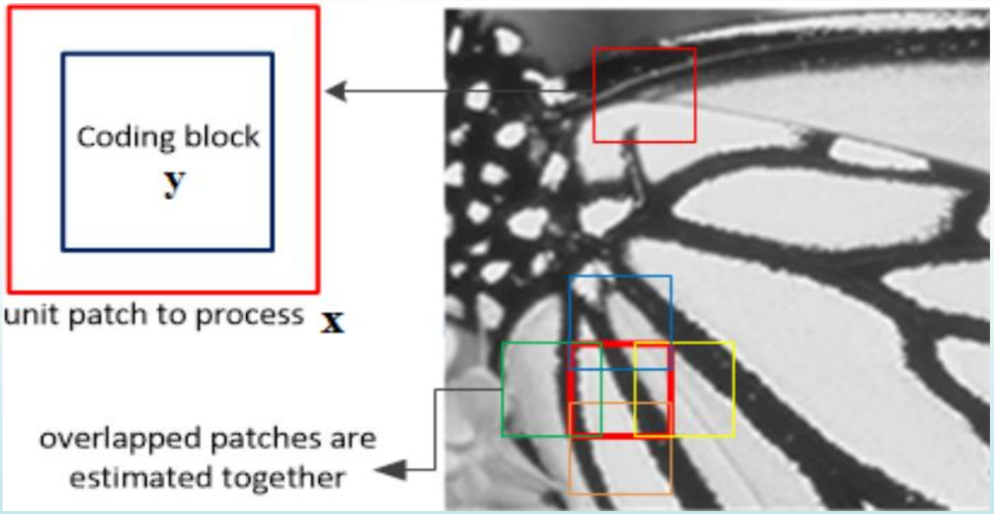
- **Limitation:** In practice, a just reasonable large dictionary is selected to reduce computation complexity.

> • Atoms do not have enough high-frequency information!
> • Their linear combination also will not produce high-frequency information!

- **Solution:** We need a new prior to assist sparsity prior to produce high-frequency features:
  - Graph-signal smoothness prior!
  - It mines the knowledge of input JPEG image itself.

- Xianming Liu, Gene Chueng, Xiaolin Wu, Debin Zhao, Soft Decoding of JPEG Compressed Images with Sparsity and Graph-Signal Smoothness Priors, submitted to IEEE Transactions on Image Processing.

Setting



- Define a pixel patch **x** enclosing a coded block as the unit patch

  → **x** is extracted in an overlapping fashion.

- Neighboring overlapped blocks are estimated jointly

  → inter-block consistency

Graph Signal Smoothness Prior

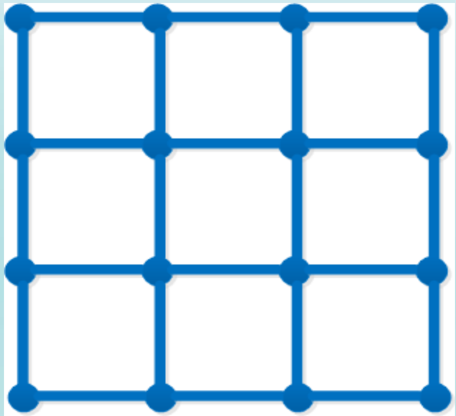
➢ Graph signal smoothness prior

- Promote appropriate smoothing!

How？→ capture underlying structure by graph!

It helps to restore sharp discontinuities (structure)!

➢ Graph Construction



- Each pixel is a node of the graph;
- 4-connected graph;
- Edge weights:

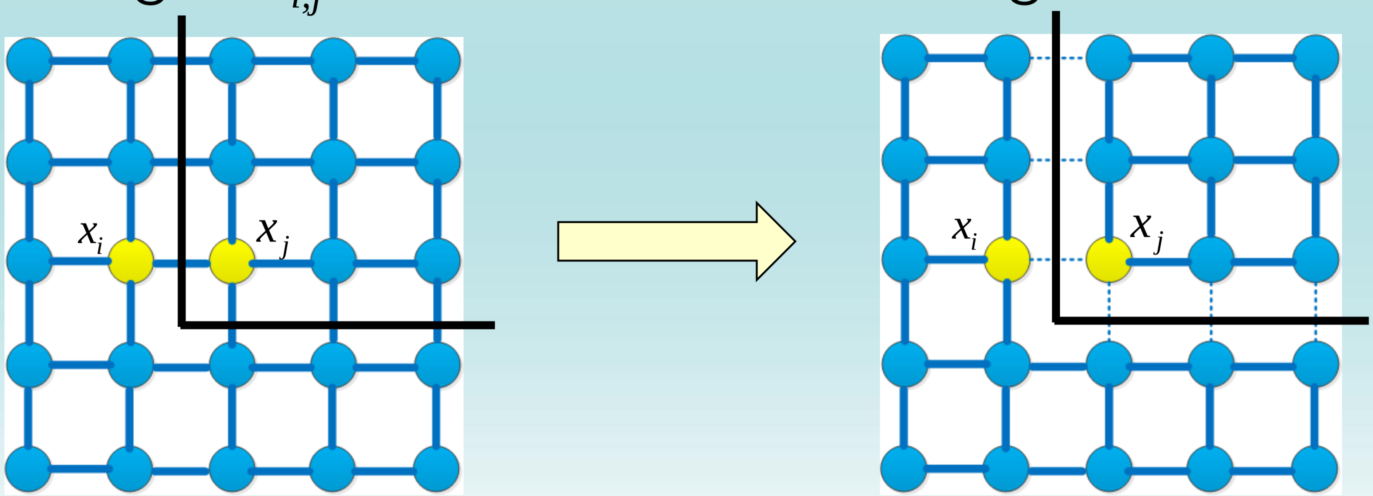
$$W_{i,j} = \exp\left\{ -\left\| x_i - x_j \right\|_2^2 / \sigma^2 \right\}.$$

Graph Signal Smoothness Prior

➢ **Graph Laplacian regularizer as smoothness prior**

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \frac{1}{2} \sum_{i,j} (x_i - x_j)^2 W_{i,j}$$

- It is small if the variation of (x_i, x_j) is small, or the weight $W_{i,j}$ is small.
- Smoothing along edges!
- Example: if a discontinuity is expected at (x_i, x_j) , one can pre-set a small edge weight $W_{i,j}$ to avoid over-smoothing.



Initial Estimation of **x**

- A good initial estimation of **x** is needed to construct the graph.
- Given q-bin constraints, we compute a minimum mean square error (MMSE) solution $\mathbf{x}^0$ as the initial estimate of **x**:
  - each MMSE coefficient Y_i^o is computed as:

$$Y_i^o = \arg\min_{Y_i^o} \int_{q_i Q_i}^{(q_i+1)Q_i} (Y_i^o - Y_i)^2 \, \underline{Pr(Y_i)} dY_i,$$

Laplacian probability distribution

  - we obtain a closed-form solution:

$$Y_i^o = \frac{(q_i Q_i + \mu) e^{\left\{-\frac{q_i Q_i}{\mu}\right\}} - ((q_i+1)Q_i + \mu) e^{\left\{-\frac{(q_i+1)Q_i}{\mu}\right\}}}{e^{\left\{-\frac{q_i Q_i}{\mu}\right\}} - e^{\left\{-\frac{(q_i+1)Q_i}{\mu}\right\}}}.$$

  - $\mathbf{x}^0$ can be finally obtained by performing inverse DCT on $\{Y_i^o\}$.

Objective Function

- Final objective function, combining two priors and inter-block consistency:

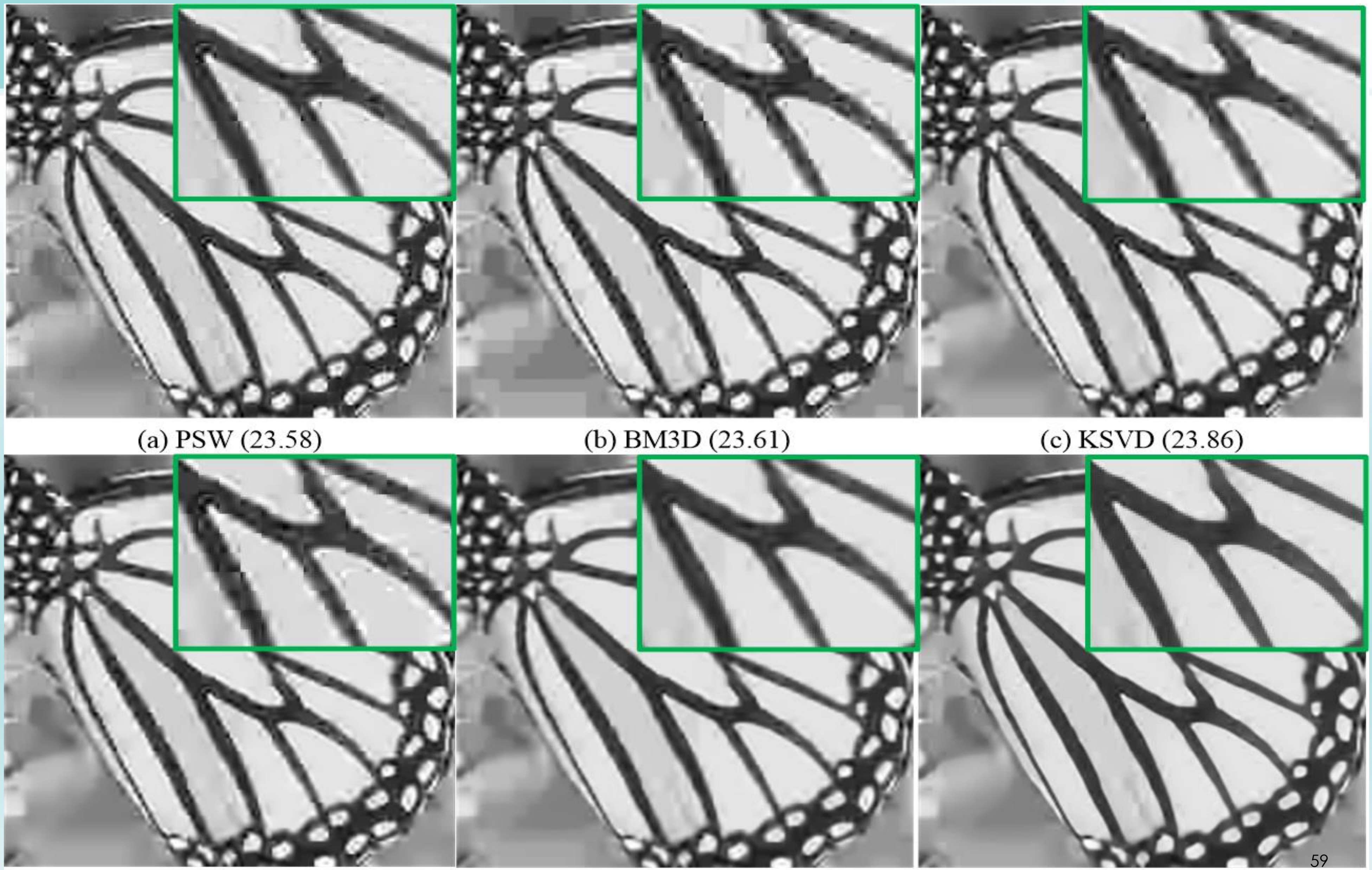
$$\arg\min_{\{\mathbf{x}_i,\ \boldsymbol{\alpha}_i\}} \sum_i \left\| \mathbf{x}_i - \boldsymbol{\Phi}\,\boldsymbol{\alpha}_i \right\|_2^2 + \lambda_1 \| \boldsymbol{\alpha}_i \|_1 + \lambda_2 \mathbf{x}_i^T \mathbf{L}_i \mathbf{x}_i$$

$$\text{s.t.}, q_{k,i} Q_k \le \mathbf{1}(k)^T \mathbf{TM}\mathbf{x}_i < (q_{k,i}+1) Q_k, \forall k \longleftarrow \text{Quantization constraint}$$

$$\sum_{j \in \mathrm{N}(i)} \left\| R_{i,j}\mathbf{x}_i - R_{j,i}\mathbf{x}_j \right\|_2^2 \le \tau \qquad \forall i. \longleftarrow \text{Inter-block consistency}$$

- Iterative optimization
  - fix $\mathbf{x}$ and estimate $\boldsymbol{\alpha}$;
  - fix $\boldsymbol{\alpha}$ and estimate $\mathbf{x}$;
  - quantization bin constraints.

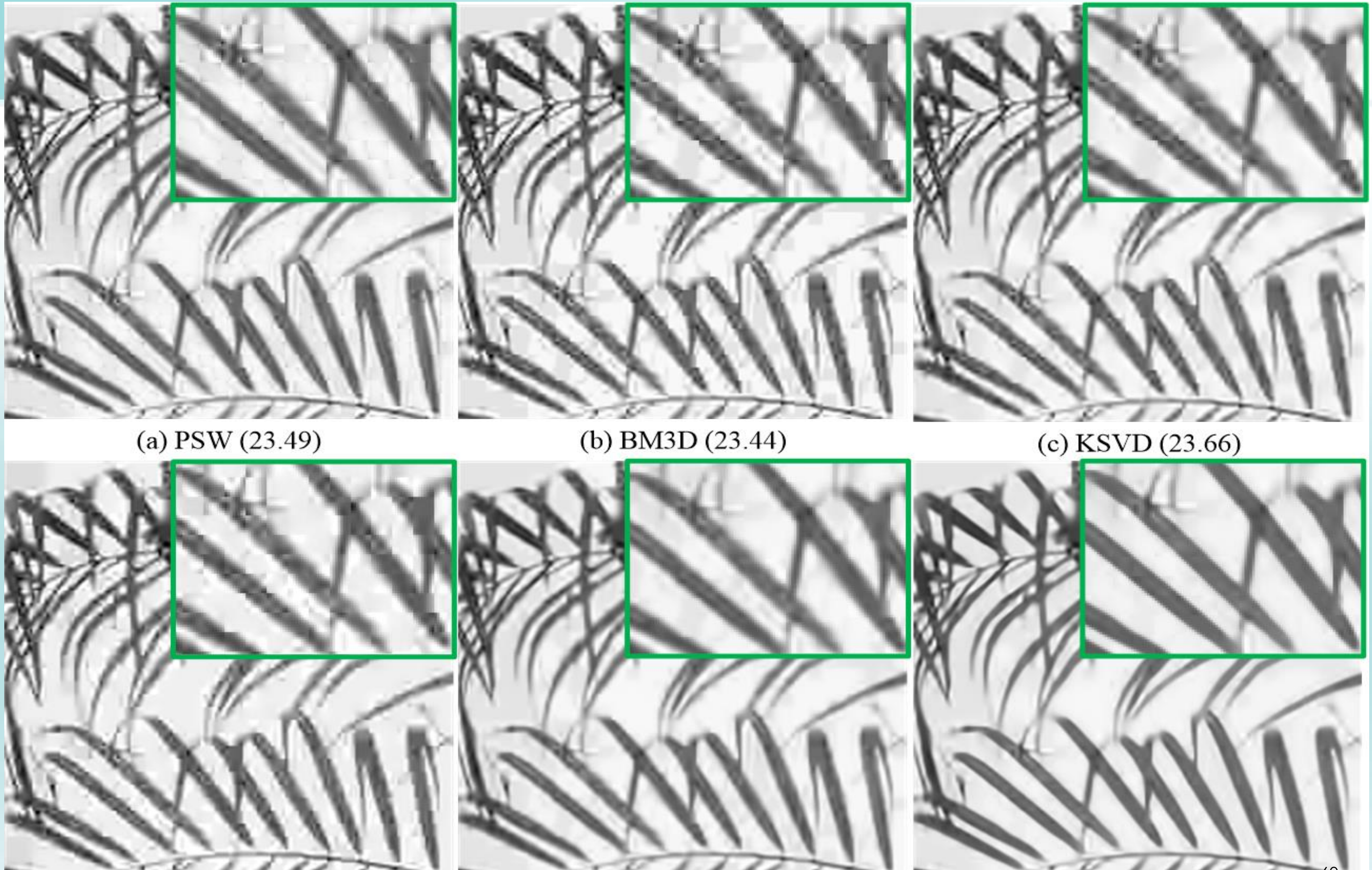
Experimentation



(a) PSW (23.58) (b) BM3D (23.61) (c) KSVD (23.86)

(d) DicTV (23.54) (e) ANCE (24.34) (f) Our result (25.71)

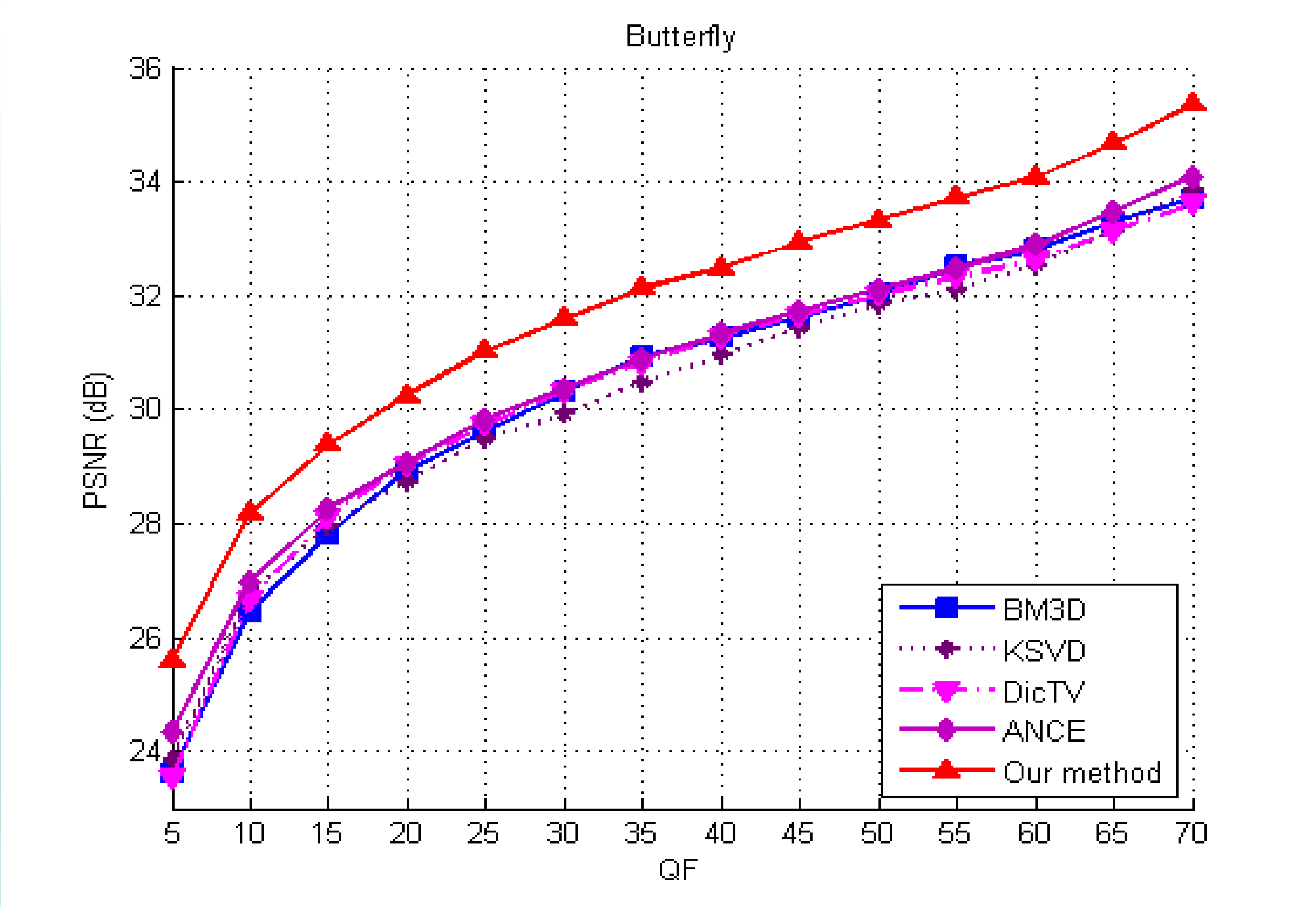
Experimentation



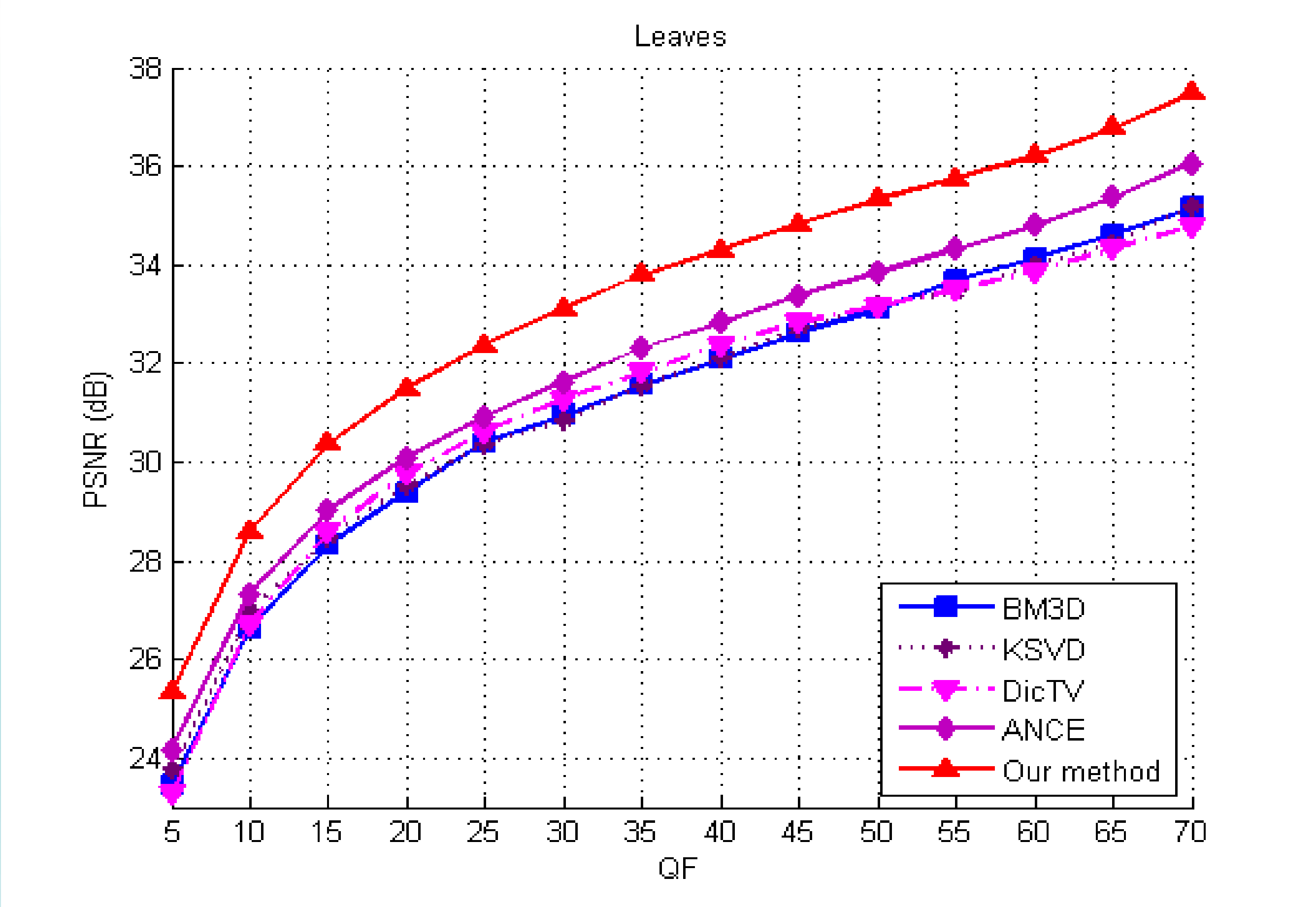
(a) PSW (23.49) (b) BM3D (23.44) (c) KSVD (23.66)

(d) DicTV (23.27) (e) ANCE (24.18) (f) Our result (25.34)

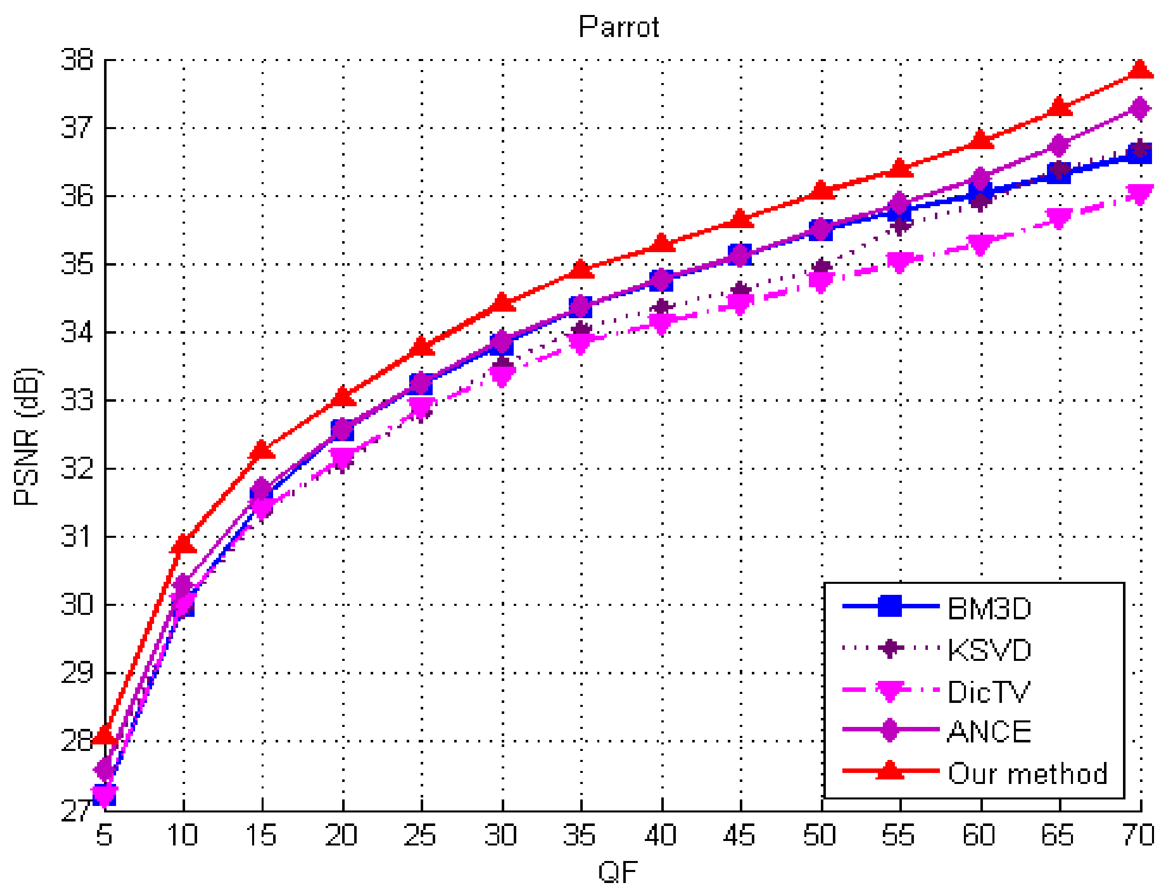
Experimentation



Experimentation



Experimentation



Outline

- Compressed Image Restoration using GSP
  - Background
  - Related Work
  - The Proposed Algorithm
- Joint Image Denoising and Contrast Enhancement using GSP
  - Background
  - Related Work
  - The Proposed Algorithm

Outline

- Compressed Image Restoration using GSP
  - Background
  - Related Work
  - The Proposed Algorithm
- Joint Image Denoising and Contrast Enhancement using GSP
  - Background
  - Related Work
  - The Proposed Algorithm

Background

- Images are often captured in poor light conditions, resulting in low-contrast images that are corrupted by acquisition noise.
  - Example: highway surveillance cameras, outdoor nighttime animal watching

- **Difficulty**: Denoising and contrast enhancement are two conflict tasks.
  - Denoising tries to remove high frequency components due to additive noise;
  - Contrast enhancement tries to enhance high-frequency features.

Background

- Sequentially performing denoising / enhancement is sub-optimal：
  - Denoising first: Useful high-frequency information may also be removed.

    How to decide a 'right' amount of noise to remove in one shot？

  - Enhancement first：Noises may also be enhanced.

    How to decide a 'right' amount of high-frequency to enhance in one shot？

Outline

- Compressed Image Restoration using GSP
  - Background
  - Related Work
  - The Proposed Algorithm
- Joint Image Denoising and Contrast Enhancement using GSP
  - Background
  - Related Work
  - The Proposed Algorithm

Related Work (1): Histogram Equalization

➢ Classical contrast enhancement approach

- Heuristic approach;

No rigorous mathematical basis exists!

- Influenced heavily by noise.

➢ Details

- Probability of level i

$$p_x(i) = p(x=i) = \frac{n_i}{n}, 0 \le i < L$$

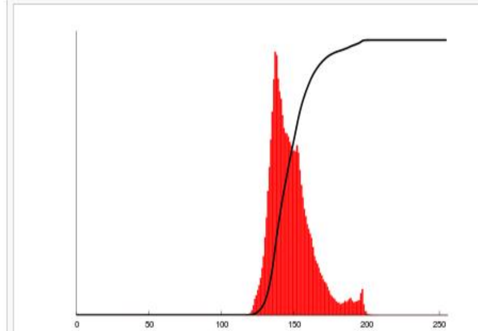
- Cumulative distribution function

$$cdf_x(i) = \sum_{j=0}^{i} p_x(j)$$

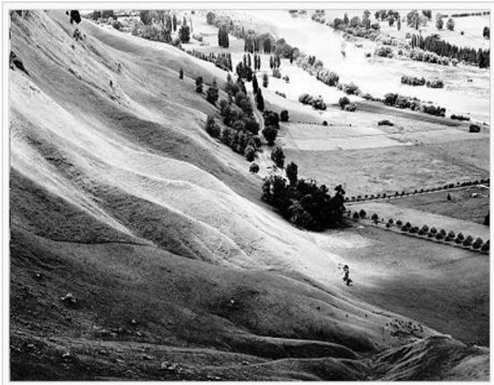
- Remapping

$$h(v) = round\left(\frac{cdf(v) - cdf_{\min}}{Size - cdf_{\min}} \times (L-1)\right)$$

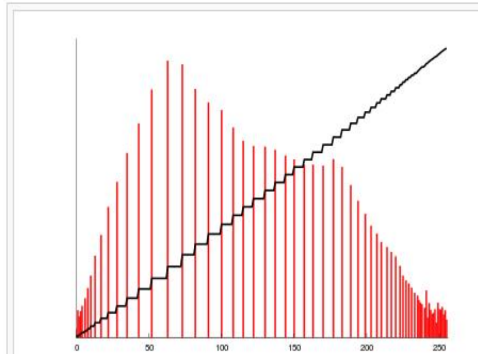
An unequalized image



Corresponding histogram (red) and cumulative histogram (black)



The same image after histogram equalization



Corresponding histogram (red) and cumulative histogram (black)

[from wiki]

Related Work (2): OCTM

- **Expected contrast gain:**

$$C(T)=\sum_{i=1}^{L-1} p_i s_i = \sum_{i=1}^{L-1} p_i \cdot [T(i)-T(j)] \qquad T(i)=\left\lfloor \sum_{t=1}^{i} s_t \right\rfloor$$

  - s_i is called the contrast at graylevel i, which is the unit rate of change from level i to level j in the output image.

- **The objective function:**

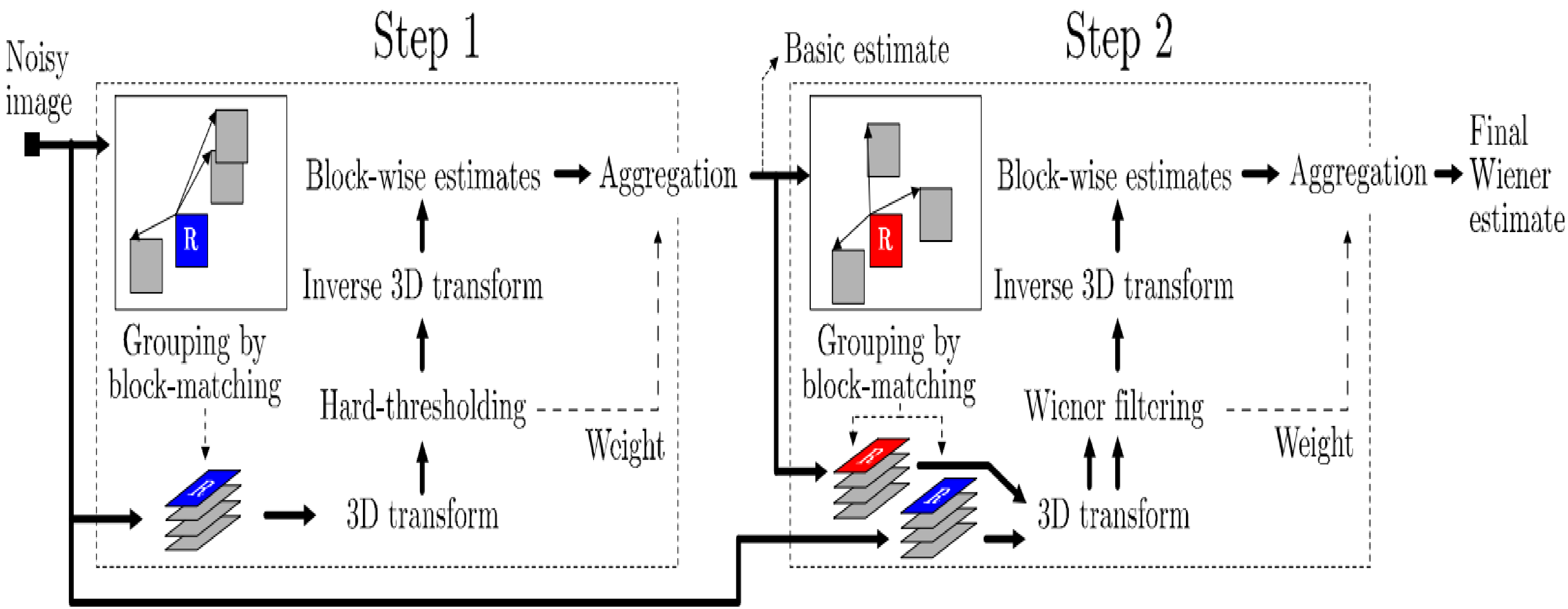
$$\max_T \sum_{i=1}^{L-1} p_i \cdot [T(i)-T(j)]$$

$$s.t., \quad \mathbf{1}^T \mathbf{s} \le L, \quad 0 \le s < u, \quad \sum_{j=0}^{d-1} s_{i+j} \ge 1, i=1,\cdots,L-d$$

- X. Wu, “ A Linear Programming Approach for Optimal Contrast-Tone Mapping,” IEEE Transactions on Image Processing. 20(5): 1262-1272 (2011)

Related Work (3): BM3D

- **One of the best image denoising algorithms so far.**



- K. Dabov, A Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering,” IEEE Transactions on Image Processing, vol. 16, no. 8, pp. 2080–2095, Aug 2007.

Related Work (4): Denoising / Contrast Enhancement

➢ [Malm et al.] maybe the first one in the literature that considers denoising / contrast enhancement together.

- **Denoising:** spatio-temporal bilateral filtering
- **Contrast enhancement:** histogram equalization
- **Perform two tasks separately**.

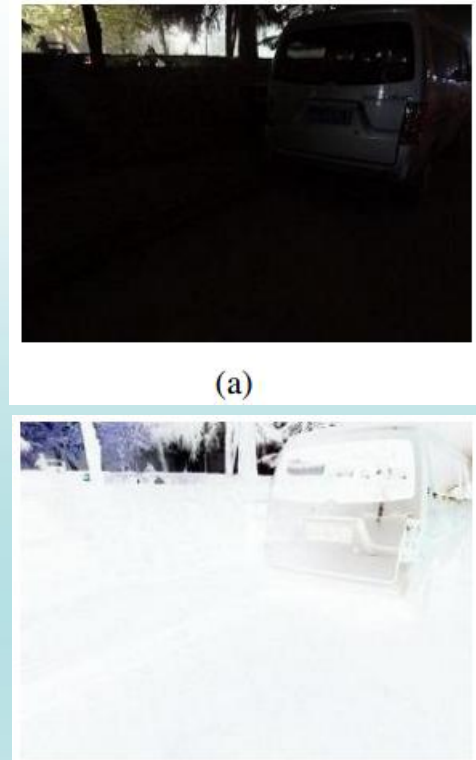
- Henrik Malm, et al., “Adaptive enhancement and noise reduction in very low light-level video,” ICCV2007, pp.1-8

Related Work (4): Denoising / Contrast Enhancement

- **Some work addresses contrast enhancement as a dehazing problem:**
  - Three image channels are inverted;

$$I^c(\mathbf{x}) = 255 - L^c(\mathbf{x})$$

  - The well-known dark-channel prior is used. [CVPR09]
- **BM3D or bilateral filter is used for denoising.**
- **Perform two tasks separately.**



(a) (b)

- X. Zhang, P. Shen, L. Luo, L. Zhang, J. Song, " Enhancement and noise reduction of very low light level images," ICPR 2012: 2034-2037
- K. He, J. Sun, X. Tang, " Single image haze removal using dark channel prior," CVPR 2009: 1956-1963

Outline

- Compressed Image Restoration using GSP
  - Background
  - Related Work
  - The Proposed Algorithm
- Joint Image Denoising and Contrast Enhancement using GSP
  - Background
  - Related Work
  - The Proposed Algorithm

Joint Denoising/ Contrast Enhancement using GSP

➢ **We propose a joint optimization framework to accomplish both tasks simultaneously:**

$$\min_{\{\mathbf{y},\boldsymbol{\alpha}\}} \|\mathbf{y}-(\mathbf{I}+\ \mathbf{H})\mathbf{x}\|_2^2+\lambda\left(\|\mathbf{y}-\ \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2+\gamma\|\boldsymbol{\alpha}\|_1\right)$$

- **x** is the captured image,
- **y** is the restored image,
- **I** is the identity matrix,
- **H** is the graph Laplacian matrix;
- **D** is the online-learned dictionary from **y**.

Why using graph Laplacian for contrast enhancement？
How does it work？

- Xianming Liu, Gene Chueng, Xiaolin Wu, "Joint Denoising and Contrast Enhancement of Images using Graph Laplacian Operator," IEEE International Conference on Acoustics, Speech and Signal Processing, 2015

Context-sensitive Contrast Enhancement using GSP

- **Context-sensitive Approach:** the contrast is defined in terms of the rate of change in intensity between neighboring pixels.
- Suppose i and j are neighboring pixels of grey values x_i and x_j , we enhance x_i to:

$$x_i^* = x_i + \sum_{j \in N_i} p(x_i, x_j)(x_i - x_j)$$

  - **Contrast increment:** $(x_i - x_j)$
  - **Joint probability:** $p(x_i, x_j)$
  - **Expected pairwise contrast gain:** $p(x_i, x_j)(x_i - x_j)$

Context-sensitive Contrast Enhancement using GSP

- **Goal:** maximize the expected total contrast gain in a neighborhood N_i :

$$C_i = \sum_{j \in N_i} p(x_i, x_j)(x_i - x_j)$$

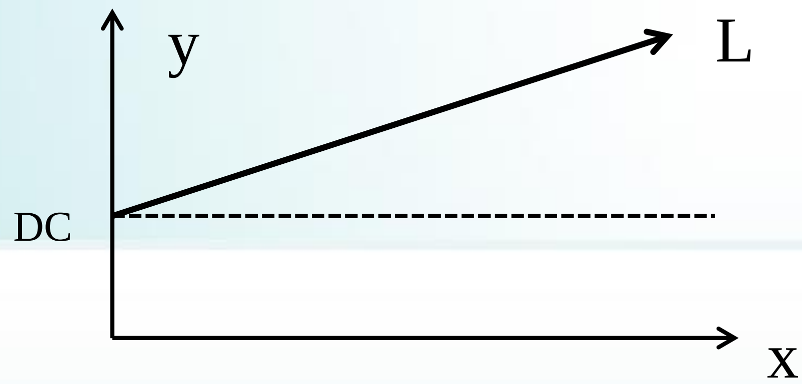
- Since natural images can be modeled as a 2D Gaussian Markov random field, it is reasonable to define the joint probability:

$$p(x_i, x_j) = \exp\left\{ -\left\| x_i - x_j \right\|_2^2 / \sigma^2 \right\}$$

$$\Downarrow$$

$$p(x_i, x_j) = W_{i,j}$$

Contrast Enhancement using Graph Laplacian



- **The optimal expected high-boosting filter:**

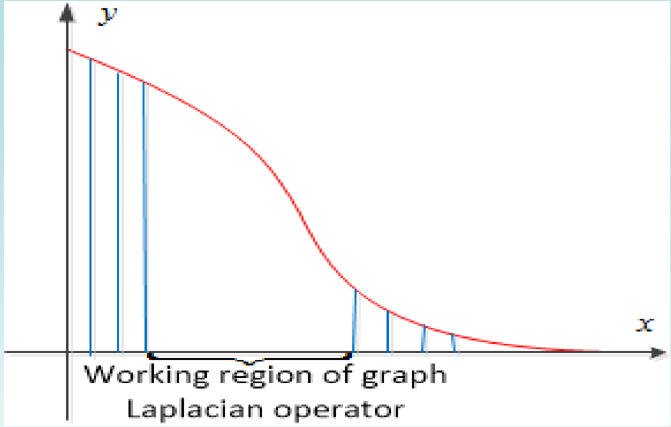
$$C = \sum_{i} \sum_{j \in N_i} W_{i,j}(x_i - x_j) = \mathbf{Lx}$$

- **L**: The graph Laplacian operator

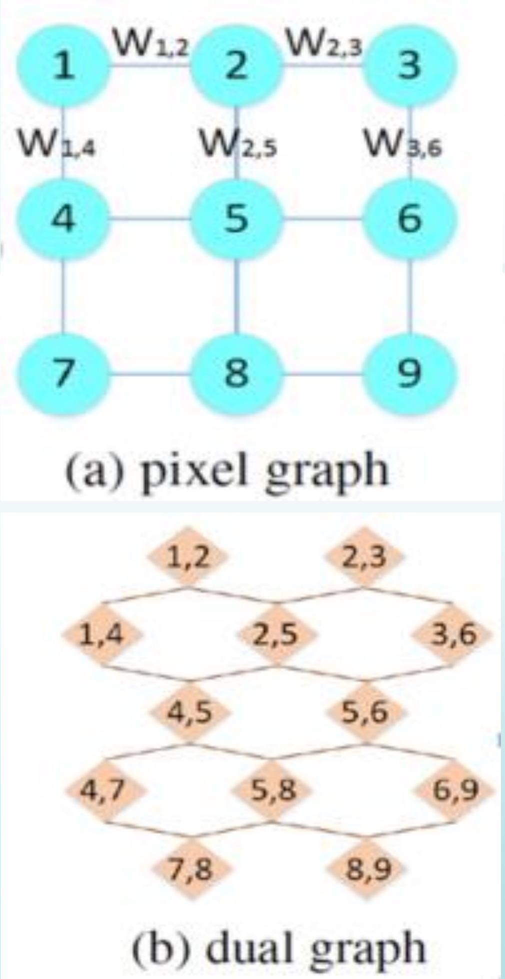
✓ L is a **high-pass** filter
→ keep DC unchanged
✓ A **linear** operator
→ make optimization easy
✓ **Signal-adaptive** enhancer
→ choose right regions to enhance

- **Finally, the contrast-enhanced image is:**

$$\mathbf{y} = \mathbf{x} + \mathbf{Lx} = (\mathbf{I} + \mathbf{L})\mathbf{x}$$



Robust Enhancer



(a) pixel graph

(b) dual graph

- **Pixel Graph:** model inter-pixel correlation, in which the edge weight is computed as:

$$W_{i,j} = \exp\left\{ -\left\| x_i - x_j \right\|_2^2 / \sigma^2 \right\}.$$

- **Dual-graph:** further constructed for robust estimate of edge weights:

$$\min_{\mathbf{z}} \left\| \mathbf{w} - \mathbf{z} \right\|_2^2 + \lambda \mathbf{z}^T \mathbf{L} \mathbf{z}$$

- **Normalized graph Laplacian**

$$\mathbf{H} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{\frac{1}{2}}$$

✓ Outliers in edge weights is removed

→ Choose the right HF to enhance

✓ DC can be changed

→ Make images brighter

Optimization

$$\min_{\{\mathbf{y},\boldsymbol{\alpha}\}} \|\mathbf{y}-(\mathbf{I}+\mathbf{H})\mathbf{x}\|_2^2+\lambda\left(\|\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2+\gamma\|\boldsymbol{\alpha}\|_1\right)$$

- **The output image should satisfy contrast enhancement and denoised simultaneously.**
- **Iterative Optimization**
  - Initialization Process:

$$\mathbf{y}_0=(\mathbf{I}+\mathbf{H}_0)\mathbf{x}$$

  - Optimization of $\boldsymbol{\alpha}$

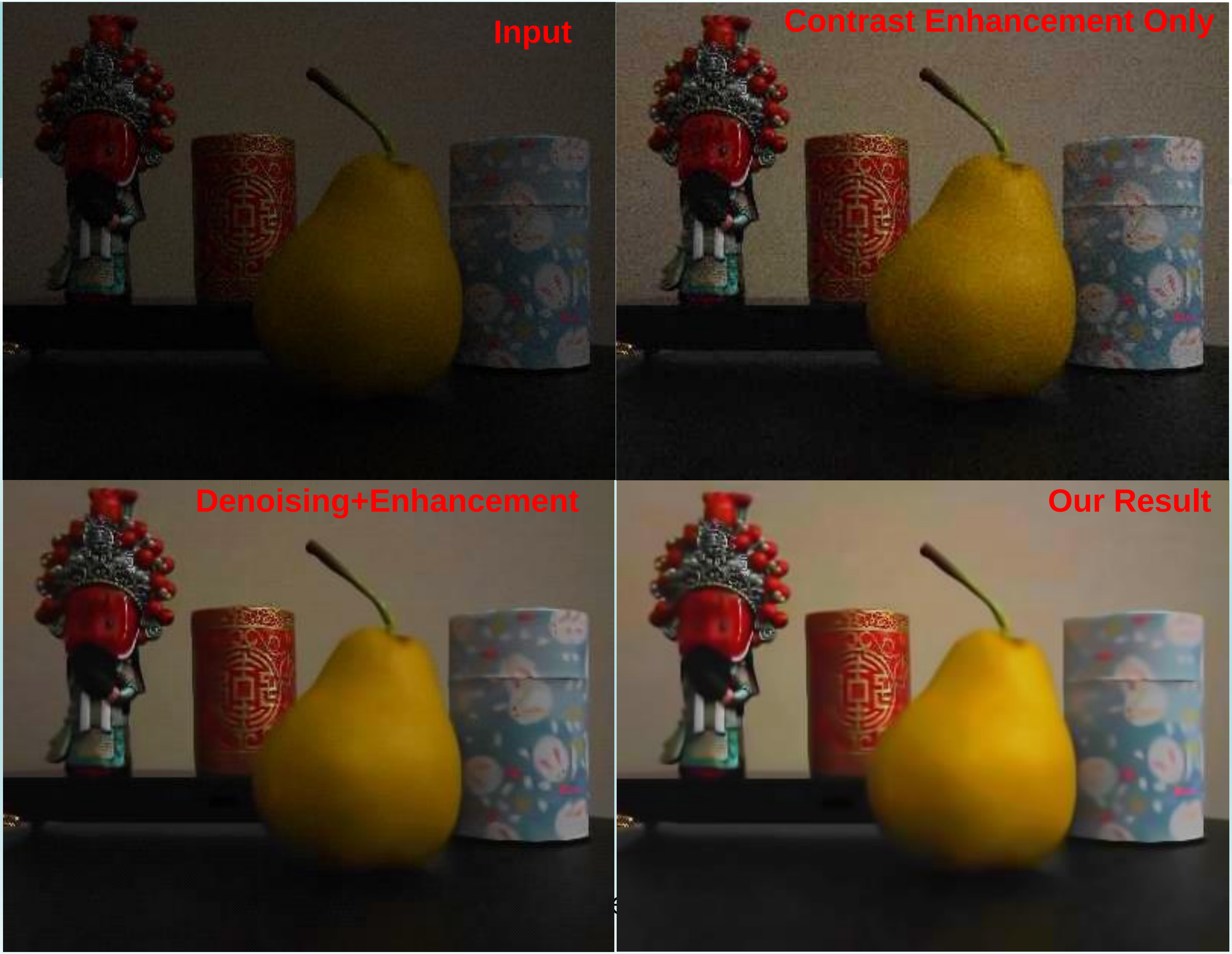
$$\arg\min_{\boldsymbol{\alpha}}\left\{\|\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2+\gamma\|\boldsymbol{\alpha}\|_1\right\}$$

  - Optimization of $\mathbf{y}$

$$\min_{\mathbf{y}}\|\mathbf{y}-(\mathbf{I}+\mathbf{H})\mathbf{x}\|_2^2+\lambda\|\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2$$

Jointly vs. Separately:
change one shot to many iterations


Input
Contrast Enhancement Only
Denoising+Enhancement
Our Result
REAL-TIME RENDERING


Input
Contrast Enhancement Only
Denoising+Enhancement
Our Result

Thank you! Any question？