Gene Cheung
Associate Professor, York University
16th November, 2018

Graph Spectral Image Processing

.

Acknowledgement

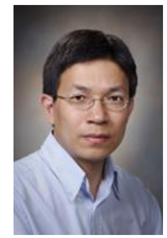
Collaborators:

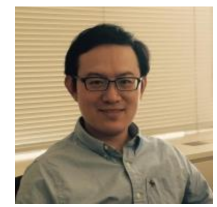
- Y. Nakatsukasa (NII, Japan)
- S. Muramatsu (Niigata, Japan)
- **A. Ortega** (USC, USA)
- **D. Florencio** (MSR, USA)
- **P. Frossard** (EPFL, Switzerland)
- J. Liang, I. Bajic (SFU, Canada)
- X. Wu (McMaster U, Canada)
- V. Stankovic (U of Strathclyde, UK)
- **P. Le Callet** (U of Nantes, France)
- **X. Liu** (HIT, China)
- W. Hu, J. Liu, Z. Guo, W. Gao (Peking U., China)
- X. Ji, L. Fang (Tsinghua, China)
- Y. Zhao (BJTU, China)
- **C.-W. Lin** (National Tsing Hua University, Taiwan)
- E. Peixoto, B. Macchiavello, E. M. Hung (U. Brasilia, Brazil)

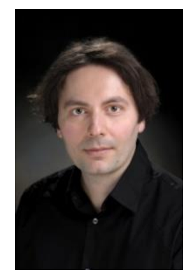








Outline

- GSP Fundamentals

- GSP for Image Compression
  - Optimality of GFT

- GSP for Inverse Imaging
  - Graph Laplacian Regularizer (GLR)
  - Reweighted Graph TV

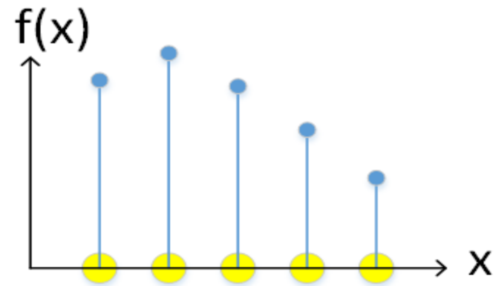
- Deep GLR

- Ongoing & Future Work

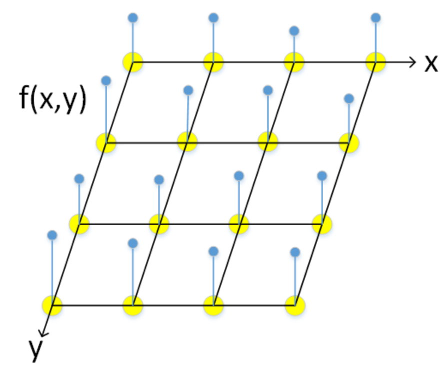
Outline

- GSP Fundamentals
- GSP for Image Compression
    - Optimality of GFT
- GSP for Inverse Imaging
    - Graph Laplacian Regularizer
    - Reweighted Graph TV
- Deep GLR
- Ongoing & Future Work

Digital Signal Processing

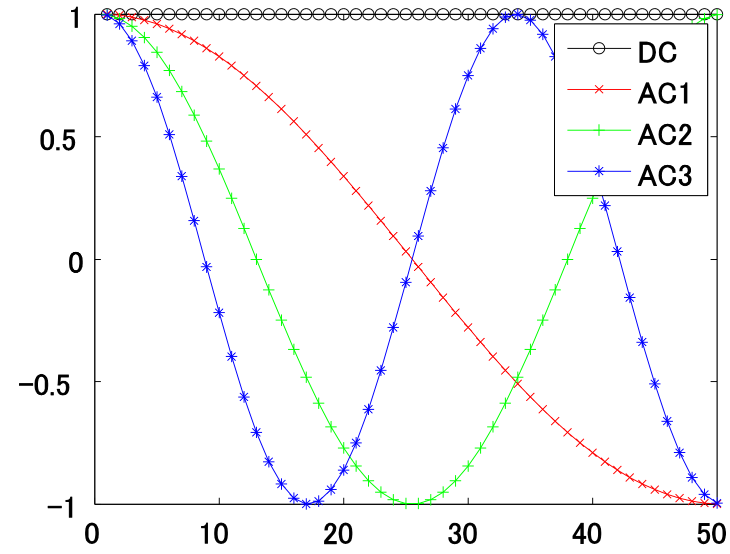


- Discrete signals on ***regular*** data kernels.
  - Ex.1: audio on regularly sampled timeline.
  - Ex.2: image on 2D grid.



- **Harmonic analysis** tools (transforms, wavelets) for diff. tasks:
  - Compression.
  - Restoration.
  - Segmentation, classification.

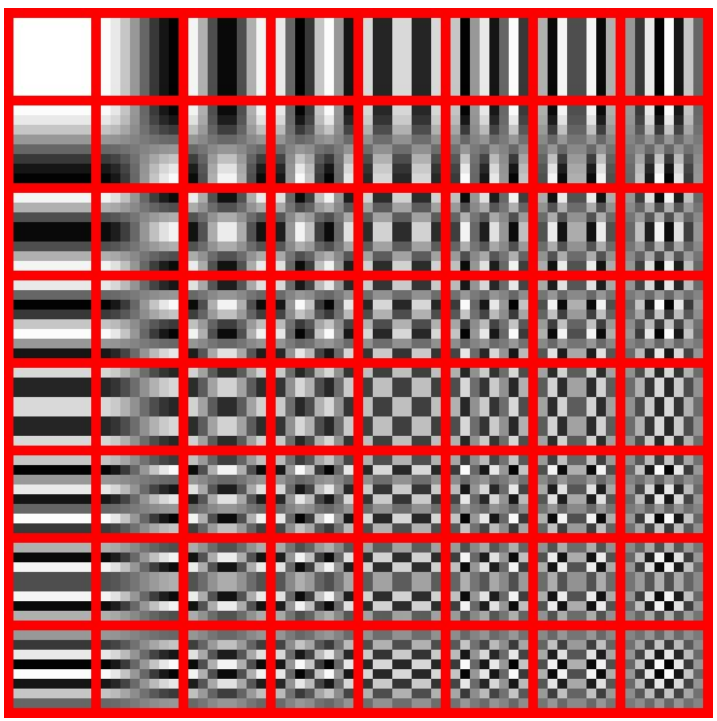
Smoothness of Signals



- Signals are often **smooth**.
- Notion of *frequency*, *band-limited*.
- Ex.: **DCT**:

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right)$$

2D DCT basis



$$\mathrm{a} = \Phi\,\mathrm{x}$$

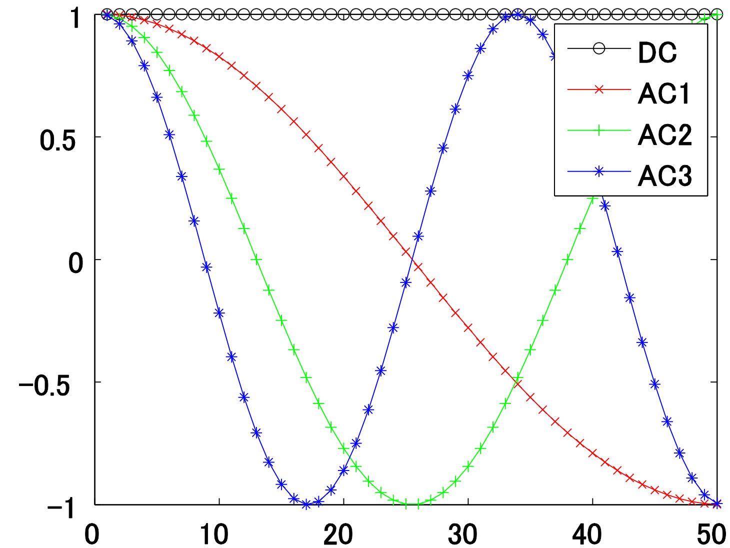
desired signal

transform

transform coeff.

$$\mathrm{a} = \begin{bmatrix} a_0 \\ a_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

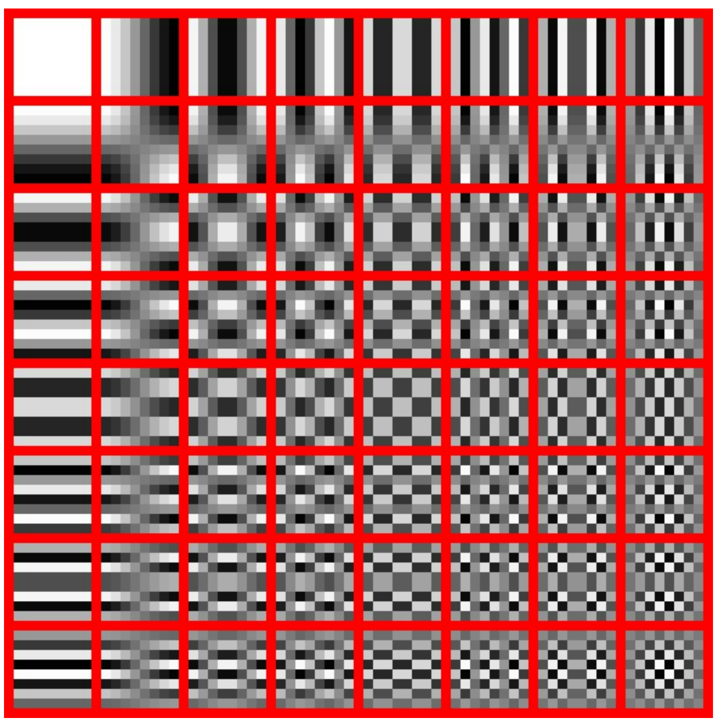
Smoothness of Signals



- Signals are often **smooth**.
- Notion of *frequency*, *band-limited*.
- Ex.: **DCT**:

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right)$$

2D DCT basis



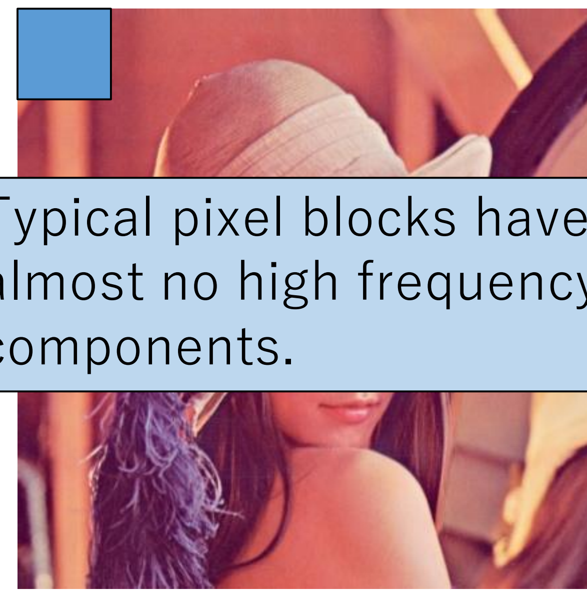
$$\mathrm{a} = \Phi\,\mathrm{x}$$

desired signal

transform

transform coeff.

$$\mathrm{a} = \begin{bmatrix} a_0 \\ a_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

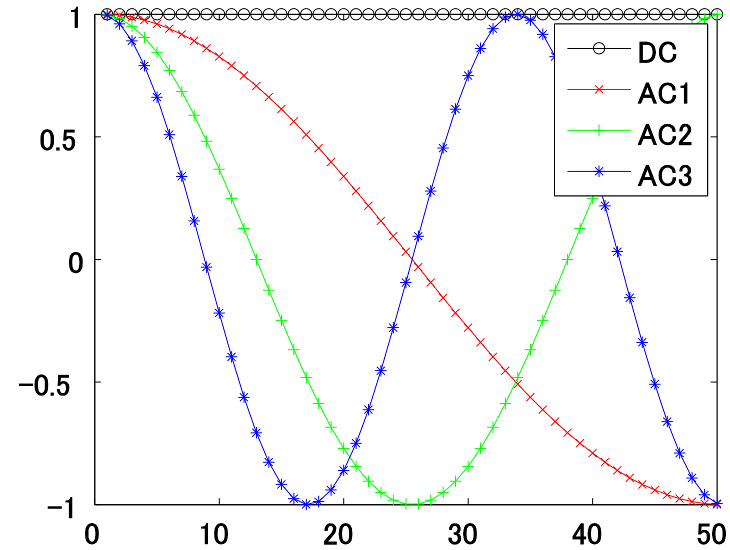


Typical pixel blocks have almost no high frequency components.

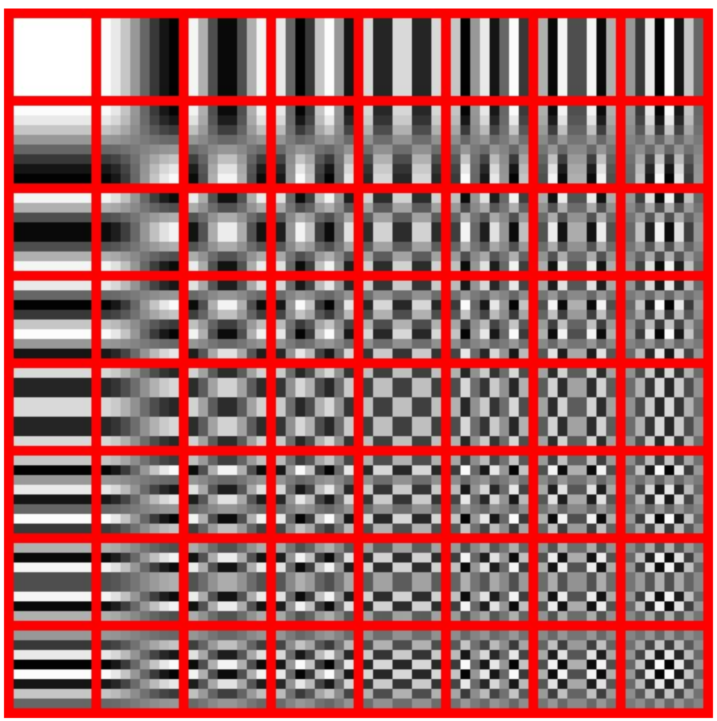
Smoothness of Signals

- Signals are often **smooth**.
- Notion of *frequency*, *band-limited*.
- Ex.: **DCT**:

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right)$$



2D DCT basis



$$\mathrm{a} = \Phi\,\mathrm{x}$$

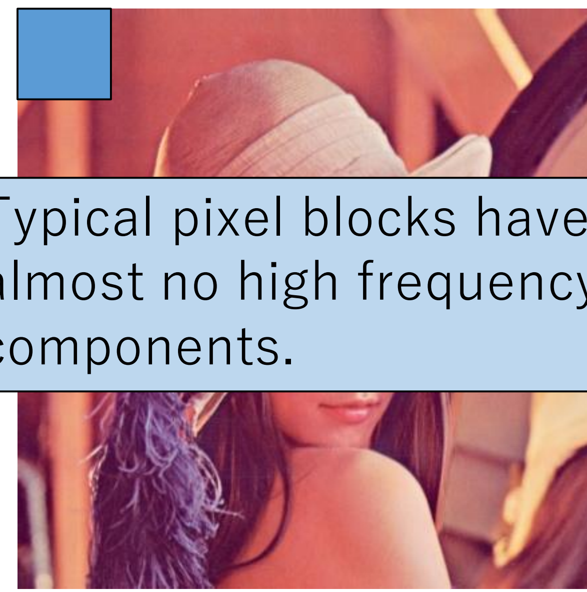
desired signal

transform

transform coeff.

$$\mathrm{a} = \begin{bmatrix} a_0 \\ a_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

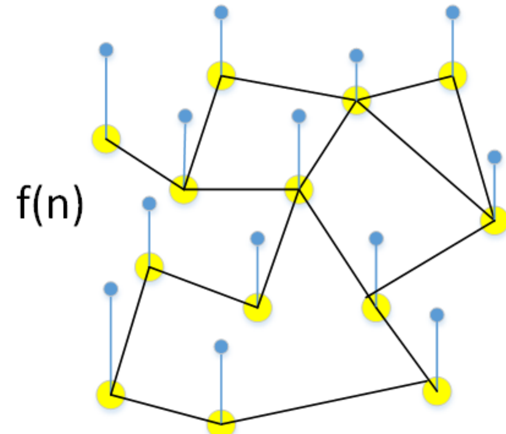
Compact signal representation



Typical pixel blocks have almost no high frequency components.

Graph Signal Processing

- Signals on ***irregular*** data kernels described by graphs.
  - Graph: nodes and edges.
  - Edges reveals *node-to-node relationships*.



example graph-signal

1. Data domain is naturally a graph.
   - Ex: ages of users on social networks.

2. Underlying data structure unknown.
   - Ex: images: 2D grid → structured graph.

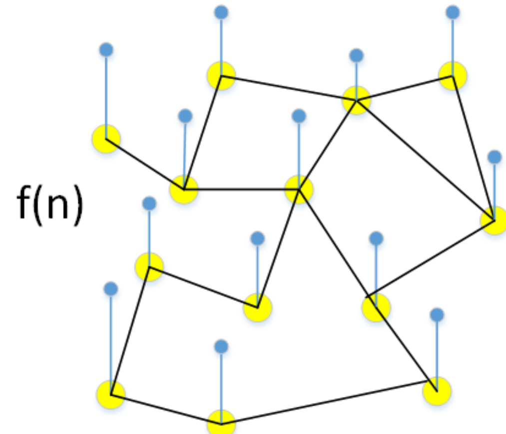
[1] D. I. Shuman et al.,"**The Emerging Field of Signal Processing on Graphs: Extending High-dimensional Data Analysis to Networks and other Irregular Domains**," *IEEE Signal Processing Magazine*, vol.30, no.3, pp.83-98, 2013.

Graph Signal Processing

- Signals on ***irregular*** data kernels described by graphs.
  - Graph: nodes and edges.
  - Edges reveals *node-to-node relationships*.

1. Data domain is naturally a graph.
   - Ex: ages of users on social networks.

2. Underlying data structure unknown.
   - Ex: images: 2D grid → structured graph.



example graph-signal

Graph Signal Processing (GSP) addresses the problem of processing signals that live on graphs.

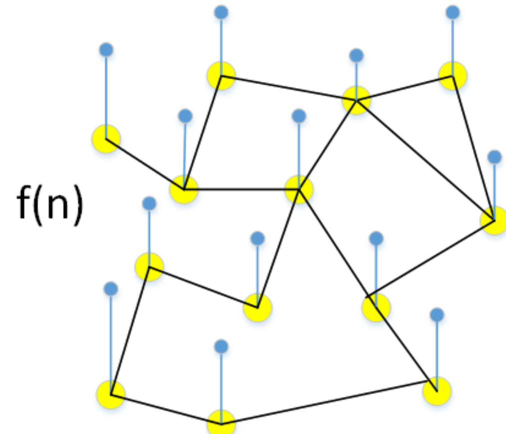
[1] D. I. Shuman et al.,"**The Emerging Field of Signal Processing on Graphs: Extending High-dimensional Data Analysis to Networks and other Irregular Domains**," *IEEE Signal Processing Magazine*, vol.30, no.3, pp.83-98, 2013.

Graph Signal Processing

- Signals on ***irregular*** data kernels described by graphs.
  - Graph: nodes and edges.
  - Edges reveals *node-to-node relationships*.

1. Data domain is naturally a graph.
   - Ex: ages of users on social networks.

2. Underlying data structure unknown.
   - Ex: images: 2D grid → structured graph.



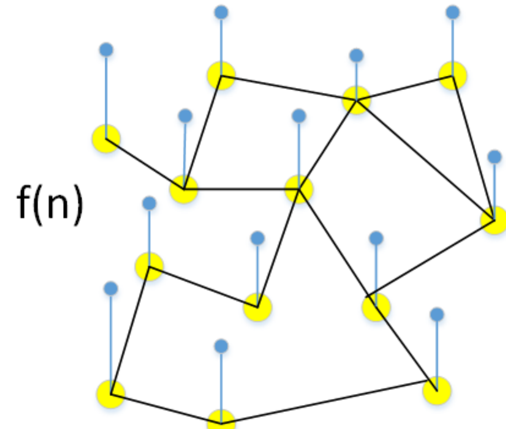
example graph-signal

Graph Signal Processing (GSP) addresses the problem of processing signals that live on graphs.

[1] D. I. 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.

Graph Signal Processing

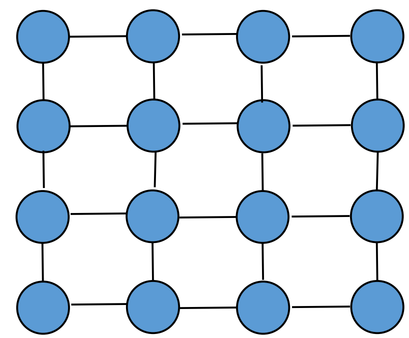
- Signals on ***irregular*** data kernels described by graphs.
  - Graph: nodes and edges.
  - Edges reveals *node-to-node relationships*.



example graph-signal

1. Data domain is naturally a graph.
   - Ex: ages of users on social networks.

2. Underlying data structure unknown.
   - Ex: images: 2D grid → structured graph.



Graph Signal Processing (GSP) addresses the problem of processing signals that live on graphs.

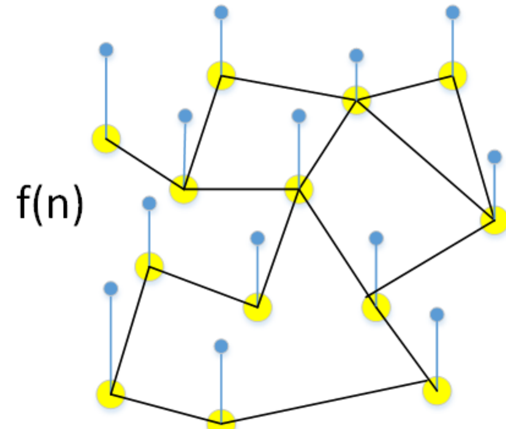
[1] D. I. Shuman et al.,"**The Emerging Field of Signal Processing on Graphs: Extending High-dimensional Data Analysis to Networks and other Irregular Domains**," *IEEE Signal Processing Magazine*, vol.30, no.3, pp.83-98, 2013.

Graph Signal Processing

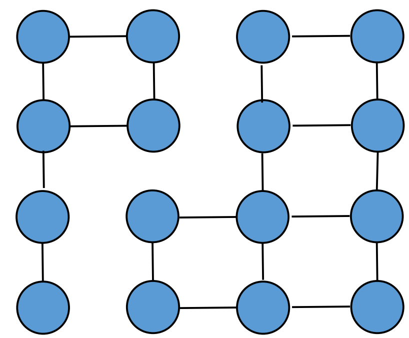
- Signals on ***irregular*** data kernels described by graphs.
  - Graph: nodes and edges.
  - Edges reveals *node-to-node relationships*.

1. Data domain is naturally a graph.
   - Ex: ages of users on social networks.

2. Underlying data structure unknown.
   - Ex: images: 2D grid → structured graph.



example graph-signal



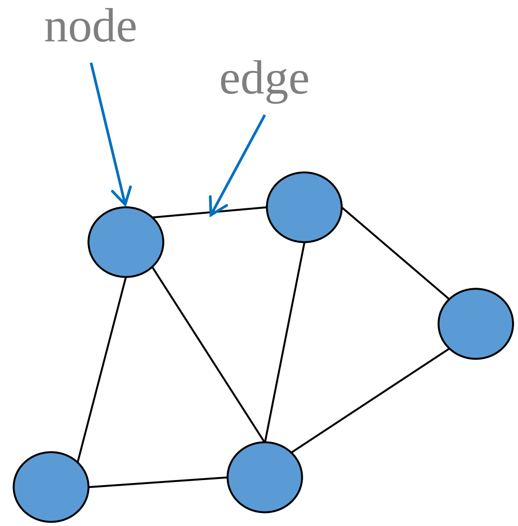
Graph Signal Processing (GSP) addresses the problem of processing signals that live on graphs.

[1] D. I. 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.

Graph Signal Processing

Research questions*:

- **Sampling**: how to efficiently acquire / sense a graph-signal?
  - Graph sampling theorems.
- **Representation**: Given graph-signal, how to compactly represent it?
  - Transforms, wavelets, dictionaries.
- **Signal restoration**: Given noisy and/or partial graph-signal, how to recover it?
  - Graph-signal priors.

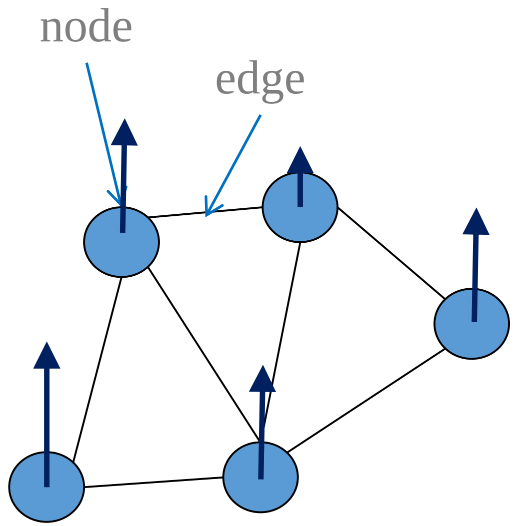


*__Graph Signal Processing Workshop__, Philadelphia, US, May, 2016. https://alliance.seas.upenn.edu/~gsp16/wiki/index.php?n=Main.Program
*__Graph Signal Processing Workshop__, Pittsburgh, US, May, 2017. https://gsp17.ece.cmu.edu/
*__Graph Signal Processing Workshop__, Lausanne, Switzerland, June, 2018. https://gsp18.epfl.ch/

Graph Signal Processing

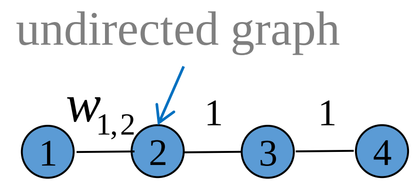
Research questions*:

- **Sampling**: how to efficiently acquire / sense a graph-signal?
  - Graph sampling theorems.
- **Representation**: Given graph-signal, how to compactly represent it?
  - Transforms, wavelets, dictionaries.
- **Signal restoration**: Given noisy and/or partial graph-signal, how to recover it?
  - Graph-signal priors.



*__Graph Signal Processing Workshop__, Philadelphia, US, May, 2016. https://alliance.seas.upenn.edu/~gsp16/wiki/index.php?n=Main.Program
*__Graph Signal Processing Workshop__, Pittsburgh, US, May, 2017. https://gsp17.ece.cmu.edu/
*__Graph Signal Processing Workshop__, Lausanne, Switzerland, June, 2018. https://gsp18.epfl.ch/

Graph Fourier Transform (GFT)



Graph Laplacian:

- *Adjacency Matrix A*: entry $A_{i,j}$ has *non-negative* edge weight $w_{i,j}$ connecting nodes *i* and *j*.

$$A = \begin{bmatrix} 0 & w_{1,2} & 0 & 0 \\ w_{1,2} & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- *Degree Matrix D*: diagonal matrix w/ entry $D_{i,i}$ being sum of column entries in row *i* of *A*.

$$D_{i,i} = \sum_j A_{i,j}$$

$$D = \begin{bmatrix} w_{1,2} & 0 & 0 & 0 \\ 0 & w_{1,2}+1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- *Combinatorial Graph Laplacian L*: $L = D\text{-}A$
  - ***L*** is *symmetric* (graph undirected).
  - ***L*** is a *high-pass* filter.
  - ***L*** is related to *2nd derivative*.

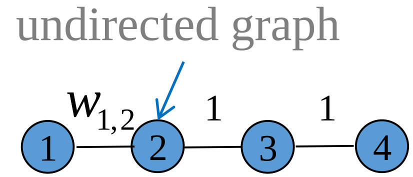
$$L = \begin{bmatrix} w_{1,2} & -w_{1,2} & 0 & 0 \\ -w_{1,2} & w_{1,2}+1 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$L_{3,:}x = -x_2 + 2x_3 - x_4$$

$$f''(x) = \lim_{h\to 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

*https://en.wikipedia.org/wiki/Second_derivative

Graph Fourier Transform (GFT)



Graph Laplacian:

$$A = \begin{bmatrix} 0 & w_{1,2} & 0 & 0 \\ w_{1,2} & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- *Adjacency Matrix A*: entry $A_{i,j}$ has *non-negative* edge weight $w_{i,j}$ connecting nodes i and j.

$$D = \begin{bmatrix} w_{1,2} & 0 & 0 & 0 \\ 0 & w_{1,2}+1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- *Degree Matrix D*: diagonal matrix w/ entry $D_{i,i}$ being sum of column entries in row i of A.

$$D_{i,i} = \sum_j A_{i,j}$$

- *Combinatorial Graph Laplacian L*: $L = D\text{-}A$
  - L is *symmetric* (graph undirected).
  - L is a *high-pass* filter.
  - L is related to *2nd derivative*.

$$L = \begin{bmatrix} w_{1,2} & -w_{1,2} & 0 & 0 \\ -w_{1,2} & w_{1,2}+1 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$L_{3,:}x = -x_2 + 2x_3 - x_4$$

$$f''(x) = \lim_{h \to 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

*https://en.wikipedia.org/wiki/Second_derivative

Graph Spectrum from GFT

- **Graph Fourier Transform** (GFT) is eigen-matrix of graph Laplacian L.

eigenvalues along diagonal

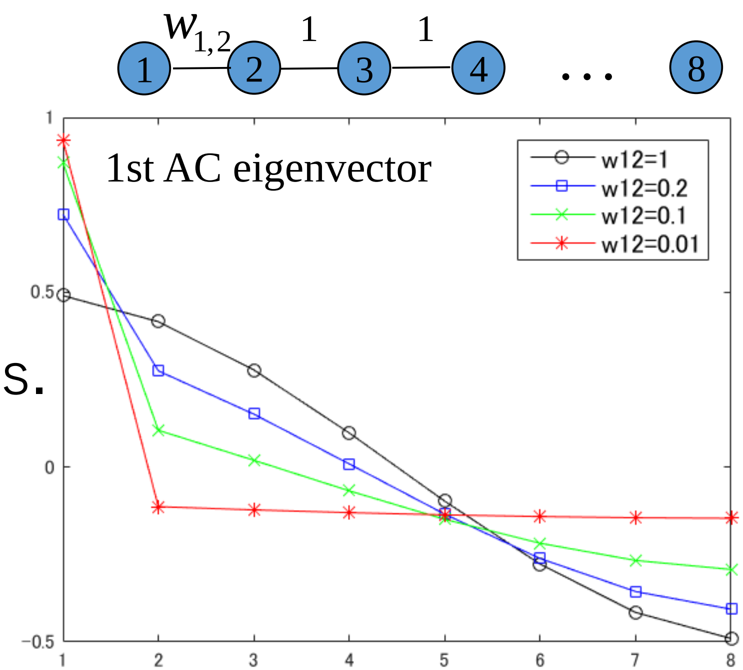
$$L = V \Sigma V^T$$

GFT

eigenvectors in columns

$$\tilde{x} = V^T x$$

GFT coefficients



1. Edge weights affect shapes of eigenvectors.
2. Eigenvalues (≥ 0) as *graph frequencies*.
   - Constant eigenvector is DC.
   - \# *zero-crossings* increases as λ increases.

- GFT defaults to *DCT* for un-weighted connected line.
- GFT defaults to *DFT* for un-weighted connected circle.

Graph Spectrum from GFT

- **Graph Fourier Transform** (GFT) is eigen-matrix of graph Laplacian *L*.

eigenvalues along diagonal

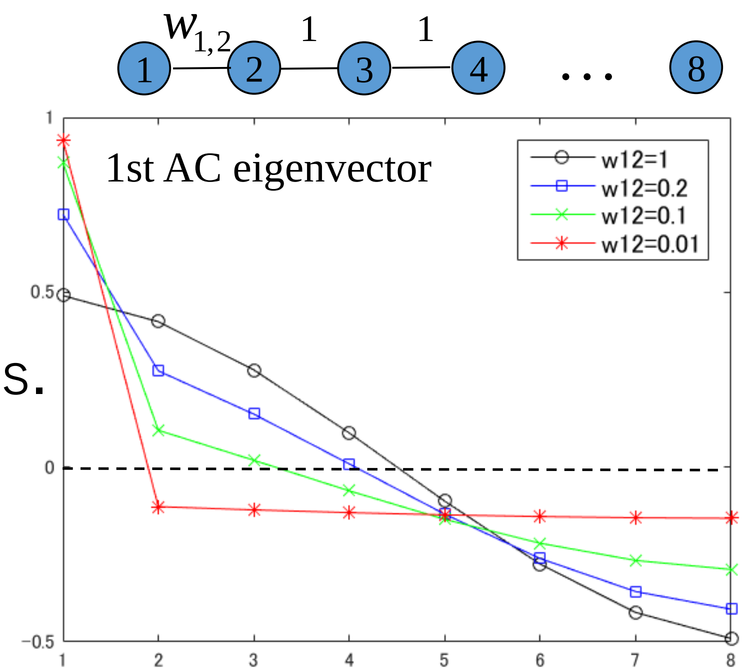
$$\mathbf{L} = \mathbf{V}\,\Sigma\,\mathbf{V}^T$$

GFT

eigenvectors in columns

$$\tilde{\mathbf{x}} = \mathbf{V}^T \mathbf{x}$$

GFT coefficients



1. Edge weights affect shapes of eigenvectors.
2. Eigenvalues (≥ 0) as *graph frequencies*.
   - Constant eigenvector is DC.
   - \# *zero-crossings* increases as λ increases.

- GFT defaults to *DCT* for un-weighted connected line.
- GFT defaults to *DFT* for un-weighted connected circle.

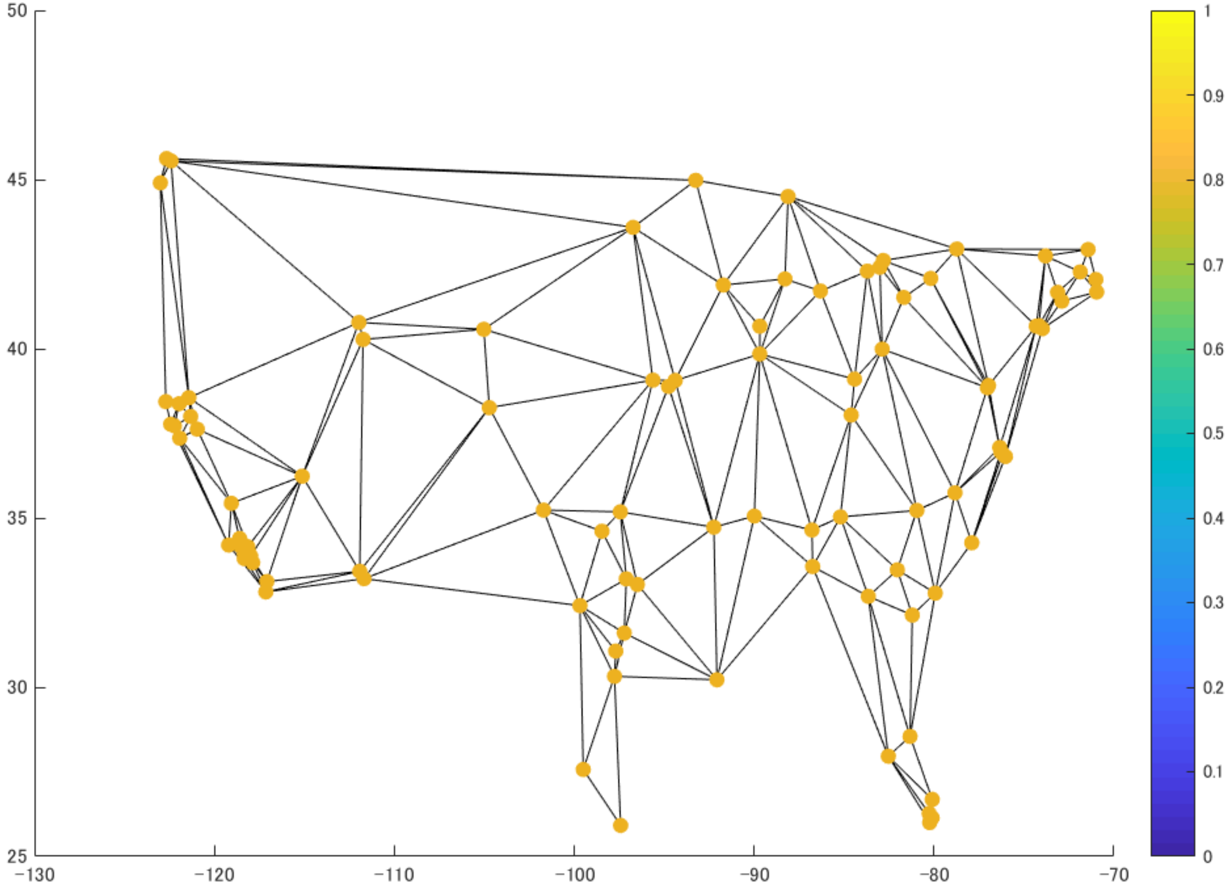
Graph Frequency Examples (US Temperature)

- Weather stations from 100 most populated cities.
- Graph connections from Delaunay Triangulation*.
- Edge weights inverse proportion to distance.

location diff.

$$w_{i,j} = \exp\left(\frac{-\left\| l_i - l_j \right\|_2^2}{\sigma^2} \right)$$

Edge weights



V1: DC component

*https://en.wikipedia.org/wiki/Delaunay triangulation

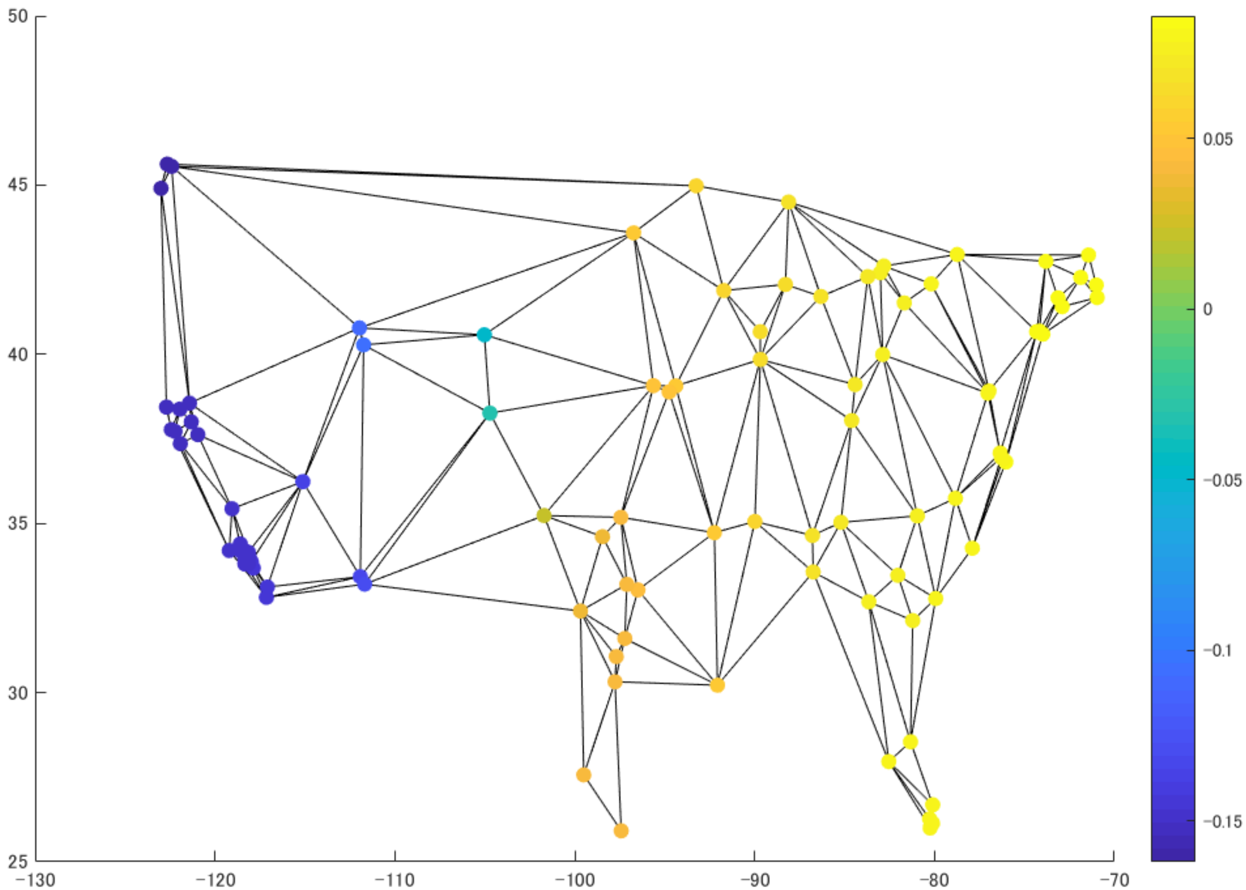
Graph Frequency Examples (US Temperature)

- Weather stations from 100 most populated cities.
- Graph connections from Delaunay Triangulation*.

location diff.

$$w_{i,j} = \exp\left(\frac{-\left\| l_i - l_j \right\|_2^2}{\sigma^2} \right)$$

Edge weights



V2: 1st AC component

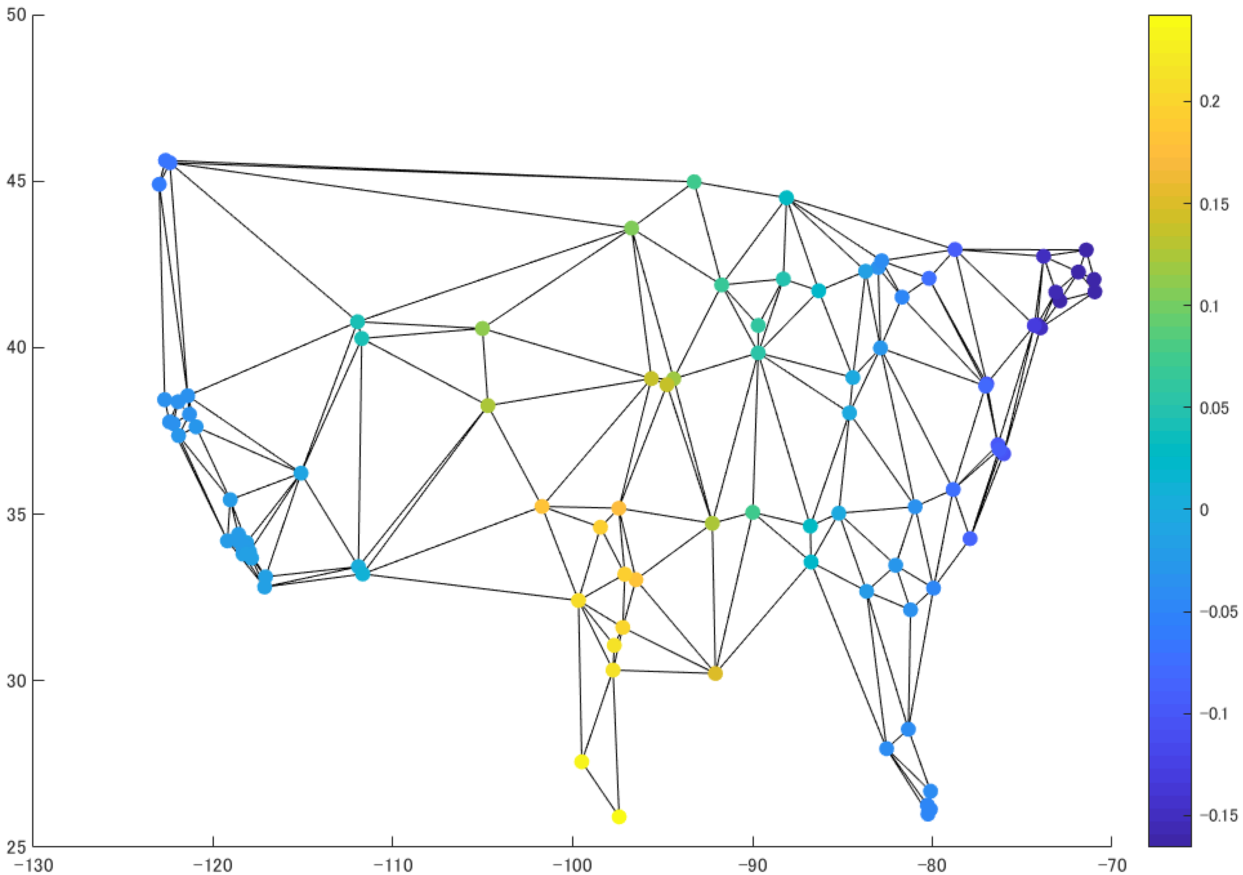
Graph Frequency Examples (US Temperature)

- Weather stations from 100 most populated cities.
- Graph connections from Delaunay Triangulation*.

location diff.

$$w_{i,j} = \exp\left(\frac{-\|l_i - l_j\|_2^2}{\sigma^2} \right)$$

Edge weights



V3: 2nd AC component

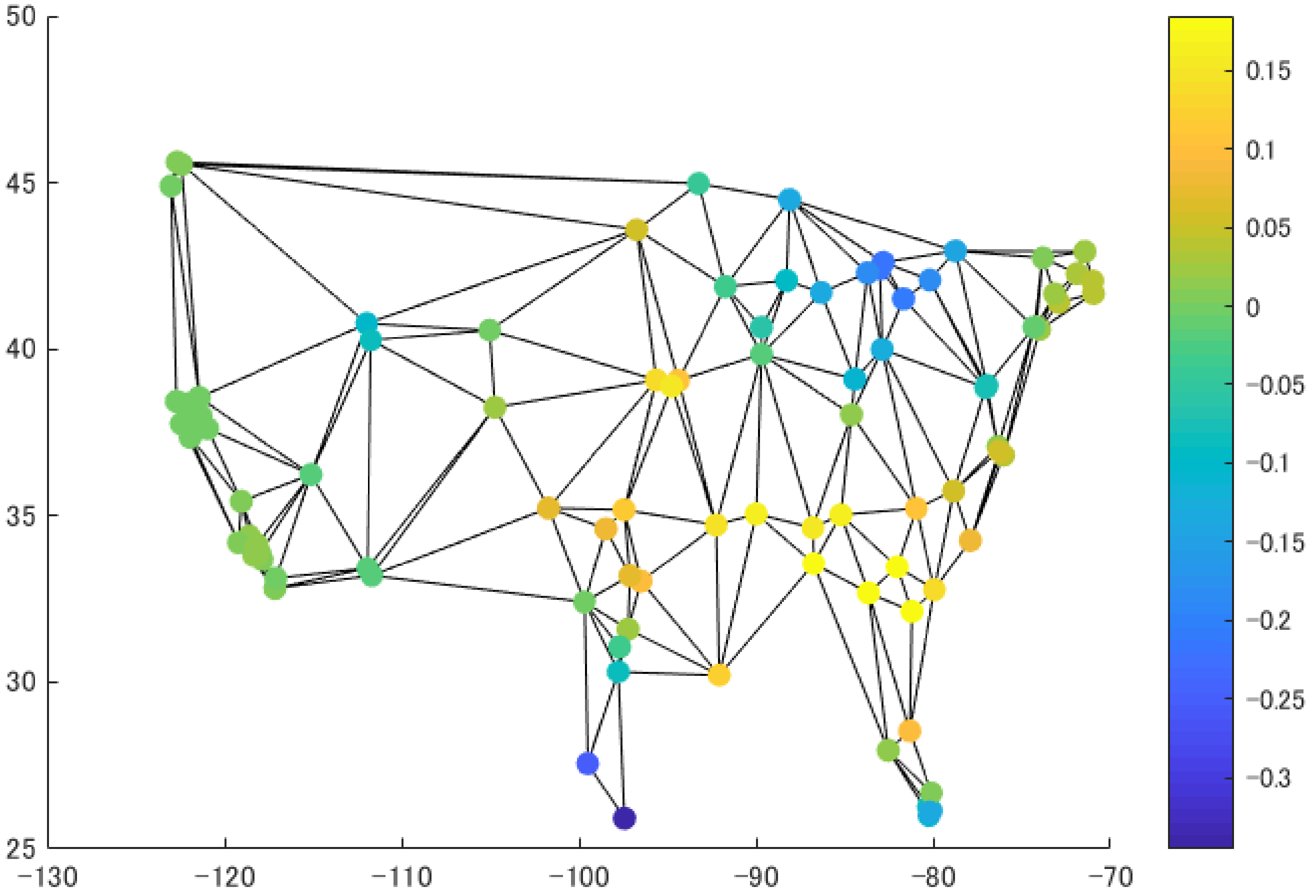
Graph Frequency Examples (US Temperature)

- Weather stations from 100 most populated cities.
- Graph connections from Delaunay Triangulation*.

location diff.

$$w_{i,j} = \exp\left(\frac{-\|l_i - l_j\|_2^2}{\sigma^2}\right)$$

Edge weights



V4: 9th AC component

Variants of Graph Laplacians

- **Graph Fourier Transform** (GFT) is eigen-matrix of graph Laplacian L.

eigenvalues along diagonal

$$\mathrm{L} = \mathrm{V}\Sigma\mathrm{V}^T$$

GFT

eigenvectors in columns

- Other definitions of graph Laplacians:
  - **Normalized** graph Laplacian:

$$L_n = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2}$$

  - **Random walk** graph Laplacian:

$$L_{rw} = D^{-1} L = I - D^{-1} A$$

  - **Generalized** graph Laplacian [1]:

$$L_g = L + D^*$$

<u>Characteristics:</u>

- Normalized.
- Symmetric.
- No DC component.

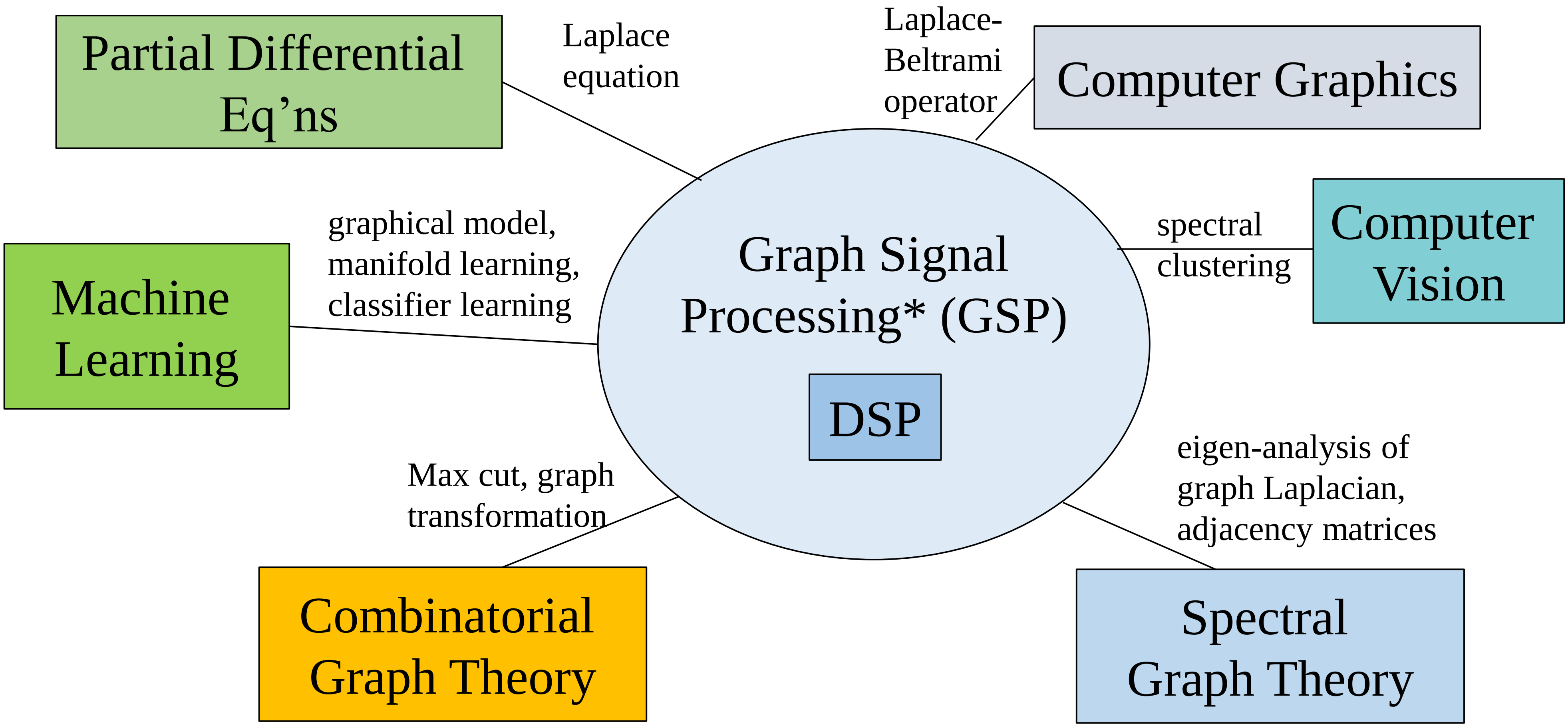
- Normalized.
- Asymmetric.
- Eigenvectors not orthog.

- Symmetric.
- **L** plus self loops.
- Defaults to DST, ADST.

[1] Wei Hu, Gene Cheung, Antonio Ortega, "**Intra-Prediction and Generalized Graph Fourier Transform for Image Coding**," *IEEE Signal Processing Letters*, vol.22, no.11, pp. 1913-1917, November 2015.

GSP and Graph-related Research

GSP: SP framework that unifies concepts from multiple fields.

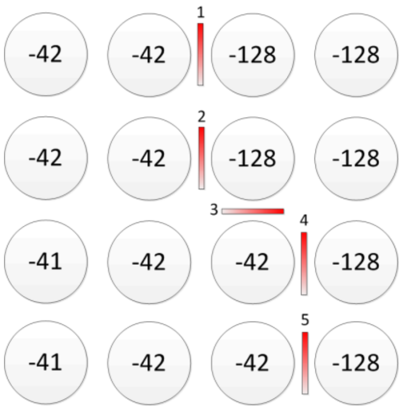


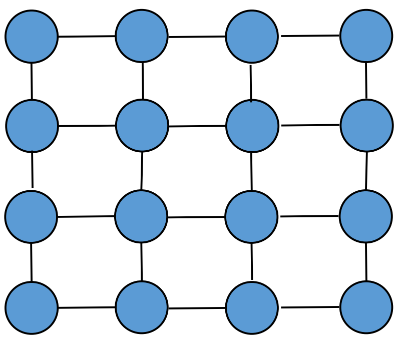
Outline

- GSP Fundamentals
- GSP for Image Compression
    - Optimality of GFT
- GSP for Inverse Imaging
    - Graph Laplacian Regularizer
    - Reweighted Graph TV
- Deep GLR
- Ongoing & Future Work

GFT for Image Compression

- DCT are ***fixed*** basis. Can we do better?
- **Idea**: use ***adaptive*** GFT to improve sparsity [1].





1. Assign edge weight 1 to adjacent pixel pairs.
2. Assign edge weight 0 to sharp signal discontinuity.
3. Compute GFT for transform coding, transmit coeff.

$$\tilde{\mathbf{x}} = \mathbf{V}^T \mathbf{x}$$

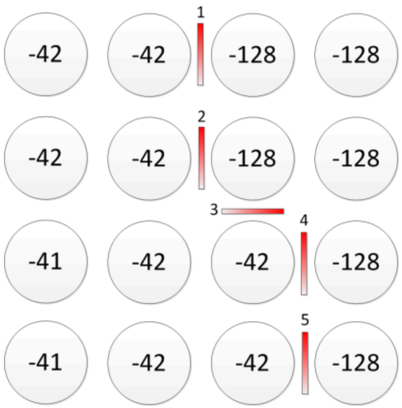
GFT

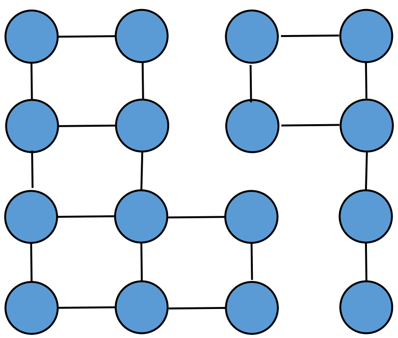
4. Transmit bits (*contour*) to identify chosen GFT to decoder (overhead of GFT).

[1] G. Shen et al., "**Edge-adaptive Transforms for Efficient Depth Map Coding**," *IEEE Picture Coding Symposium*, Nagoya, Japan, December 2010.

[2] 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.

GFT for Image Compression





- DCT are ***fixed*** basis. Can we do better?
- **Idea**: use ***adaptive*** GFT to improve sparsity [1].

1. Assign edge weight 1 to adjacent pixel pairs.
2. Assign edge weight 0 to sharp signal discontinuity.
3. Compute GFT for transform coding, transmit coeff.

GFT

$$\tilde{\mathrm{x}} = \mathrm{V}^T \mathrm{x}$$

4. Transmit bits (*contour*) to identify chosen GFT to decoder (overhead of GFT).

[1] G. Shen et al., "**Edge-adaptive Transforms for Efficient Depth Map Coding**," *IEEE Picture Coding Symposium*, Nagoya, Japan, December 2010.

[2] 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.

GFT: Derivation of Optimal Edge Weights

- Assume a 1D 1st-order ***autoregressive (AR) process*** $\mathbf{x} = [x_1, ..., x_N]^T$ where,

0-mean r.v. with large var. σ^2

$$x_k = \begin{cases} \eta & k = 1 \\ x_{k-1} + e_k & 1 < k \le N \end{cases}$$

0-mean r.v. with var. σ_k^2

$$\mathbf{F}\mathbf{x} = \mathbf{b}, \qquad \mathbf{x} = \mathbf{F}^{-1}\mathbf{b}$$

$$\begin{aligned} x_1 &= \eta \\ x_2 - x_1 &= e_2 \\ &\vdots \\ x_N - x_{N-1} &= e_N \end{aligned} \Longrightarrow \mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad \mathrm{b} = \begin{bmatrix} \eta \\ e_2 \\ \vdots \\ \\ \\ e_N \end{bmatrix}$$

GFT: Derivation of Optimal Edge Weights

- Covariance matrix

$$\begin{aligned}\mathbf{C} &= E\left[(\mathrm{x}-\mu)(\mathrm{x}-\mu)^T\right] \\ &= E\left[\mathrm{x}\,\mathrm{x}^T\right] = E\left[\mathbf{F}^{-1}\mathrm{b}\,\mathrm{b}^T\left(\mathbf{F}^{-1}\right)^T\right] \\ &= \mathbf{F}^{-1}E\left[\mathrm{b}\,\mathrm{b}^T\right]\left(\mathbf{F}^{-1}\right)^T\end{aligned}$$

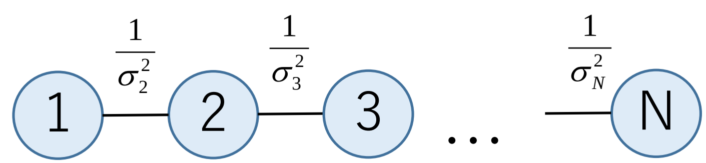
$$E\left[\mathbf{bb}^T\right] = \begin{bmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_N^2 \end{bmatrix}$$

- Precision matrix (tri-diagonal)

large

$$\mathrm{Q} = \mathrm{C}^{-1} = \begin{bmatrix} \frac{1}{\sigma^2}+\frac{1}{\sigma_2^2} & -\frac{1}{\sigma_2^2} & 0 & \cdots & 0 \\ -\frac{1}{\sigma_2^2} & \frac{1}{\sigma_2^2}+\frac{1}{\sigma_3^2} & -\frac{1}{\sigma_2^2} & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -\frac{1}{\sigma_{N-1}^2} & \frac{1}{\sigma_{N-1}^2}+\frac{1}{\sigma_N^2} & -\frac{1}{\sigma_N^2} \\ 0 & \cdots & 0 & -\frac{1}{\sigma_N^2} & \frac{1}{\sigma_N^2} \end{bmatrix} \approx \mathbf{L}$$

Graph Laplacian matrix!



Multi-resolution-GFT Implementation

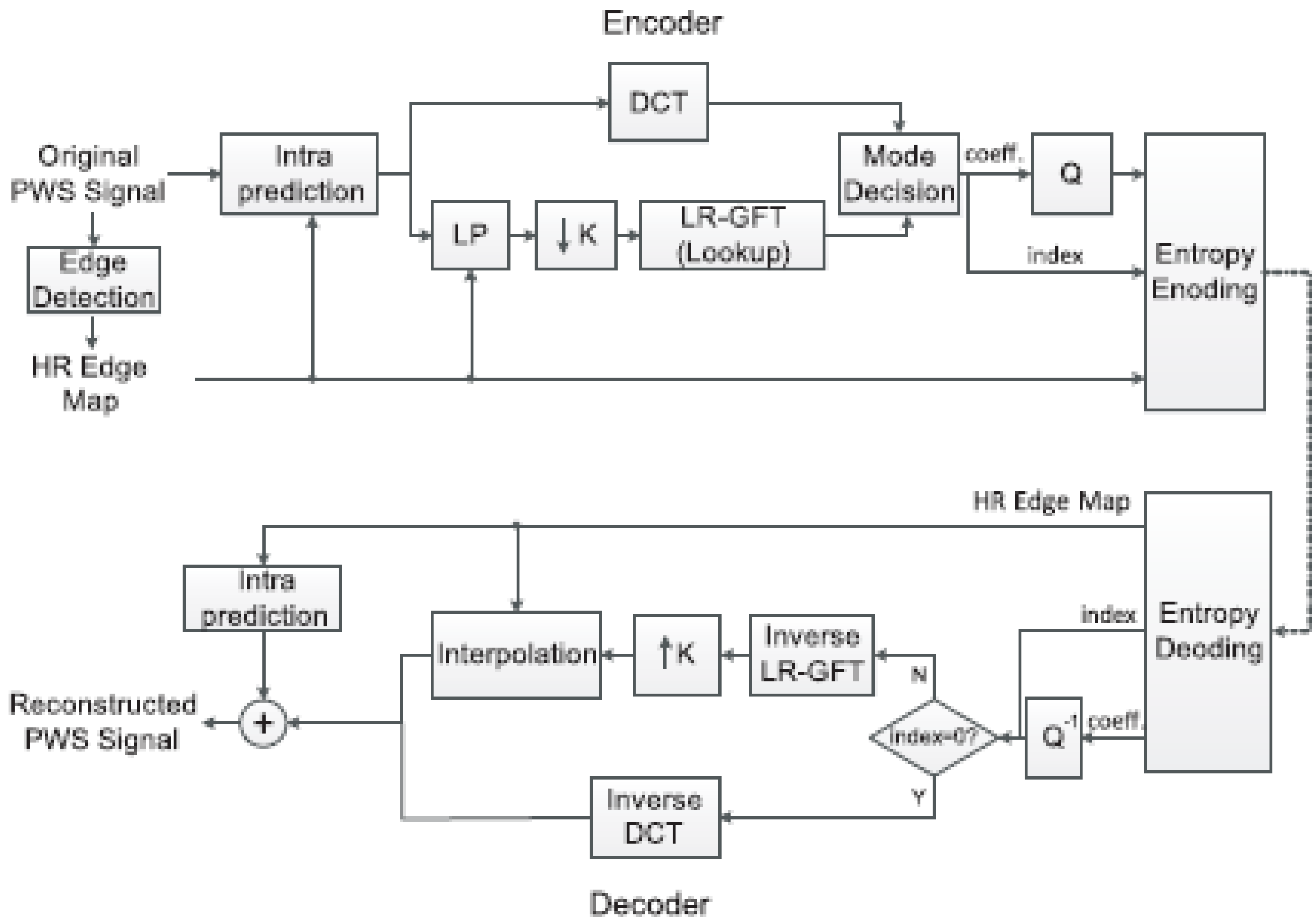
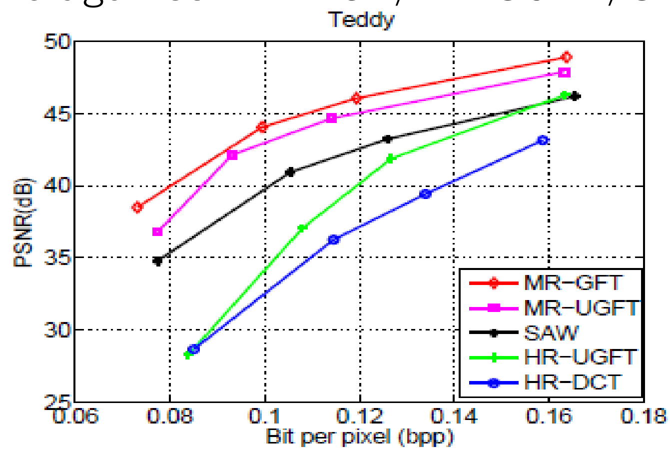


Fig. 2. MR-GFT coding system for PWS images.

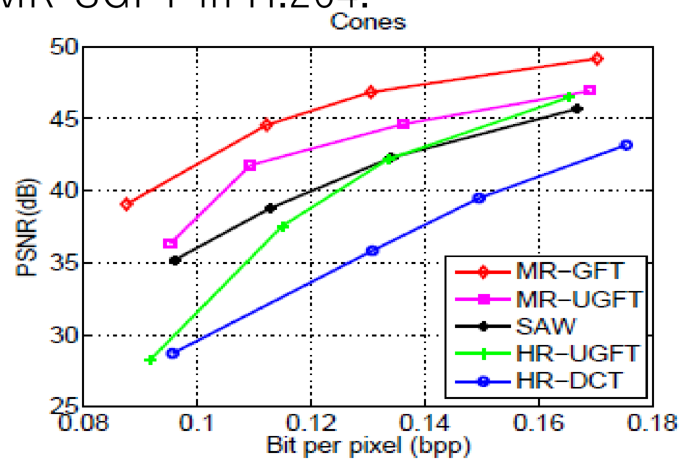
[1] Wei Hu, Gene Cheung, Antonio Ortega, Oscar Au, "**Multiresolution Graph Fourier Transform for Compression of Piecewise Smooth Images**," *IEEE Transactions on Image Processing*, vol.24, no.1, pp.419-433, January 2015.

Experimentation

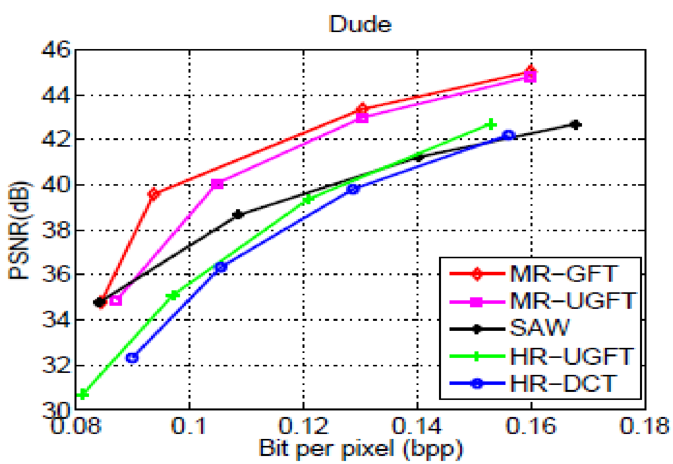
- Setup
  - Test images: depth maps of *Teddy* and *Cones*, and graphics images of *Dude* and *Tsukuba*.
  - Compare against: HR-DCT, HR-SGFT, SAW, MR-SGFT in H.264.
- Results



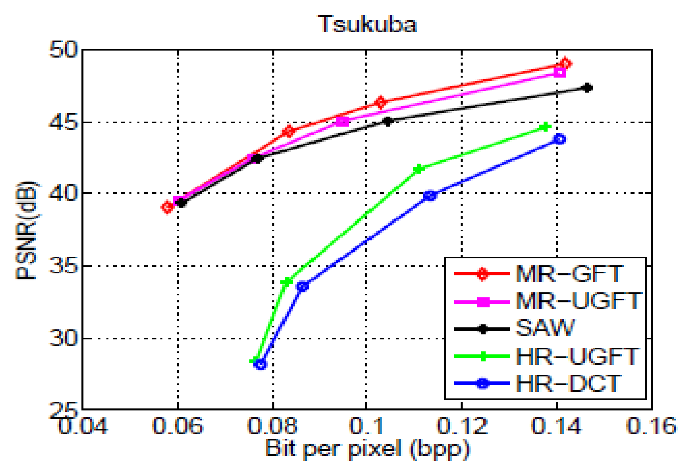
(a)



(b)



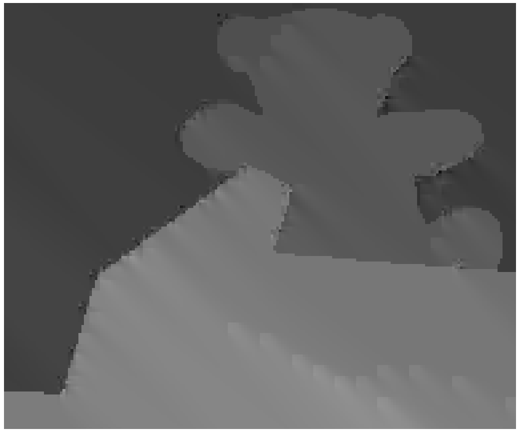
(c)

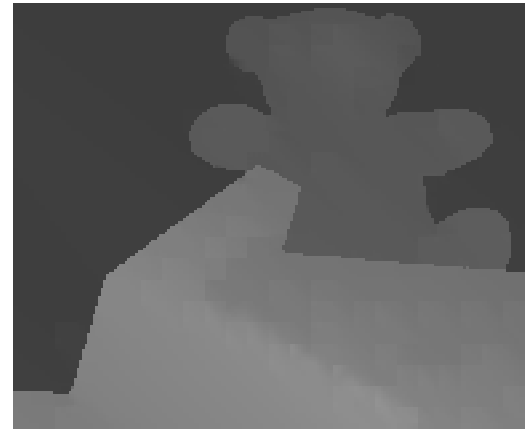


(d)

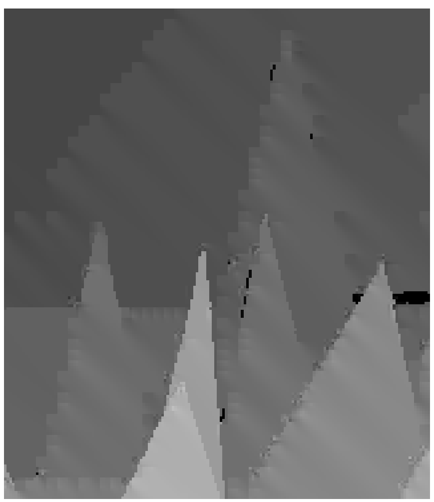
HR-DCT: 6.8dB
HR-SGFT: 5.9dB
SAW: 2.5dB
MR-SGFT: 1.2dB

Subjective Results





HR-DCT HR-SGFT MR-GFT



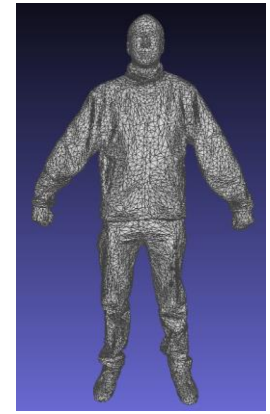
Summary of GFT for Image Coding

- Optimality of GFT for AR model.
- Variants of GFT for prediction residuals, anti-correlated pixels.
- Fast implementation (w/o eigen-decomposition) via **Graph Lifting Transform** (GLT) [1] or **Fast Graph Fourier Transform** (FGFT) [2].

[1] Y.-H. Chao et al., "Edge-Adaptive Depth Map Coding with Lifting Transform on Graphs," 31st Picture Coding Symposium, Cairns, Australia, May, 2015.

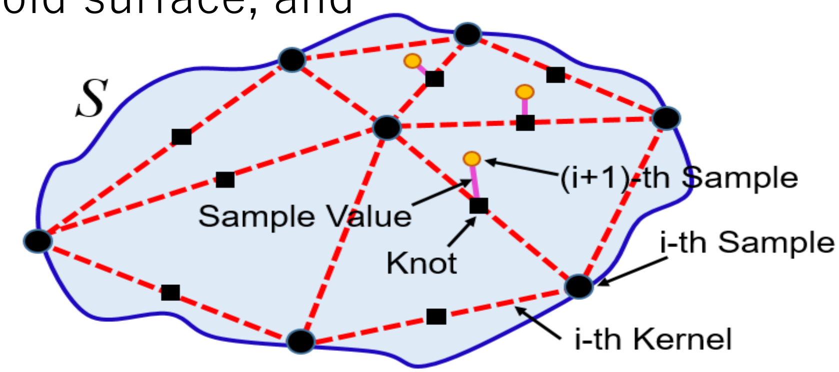
[2] L. Le Magoarou et al., "Approximate Fast Graph Fourier Transforms via Multilayer Sparse Approximations," *IEEE TSIPN*, May, 2018.

Graph-Signal Sampling / Encoding for 3D Point Cloud



MIT *dataset**

- **Problem**: Point clouds require encoding specific 3D coordinates.
- **Assumption**: smooth 2D manifold in 3D space.
- **Proposal**: progressive 3D geometry rep. as series of graph-signals.
    1. adaptively identifies new samples on the manifold surface, and
    2. encodes them efficiently as graph-signals.

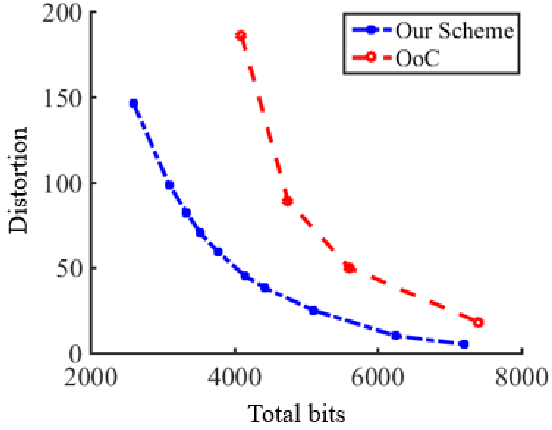


- **Example**:

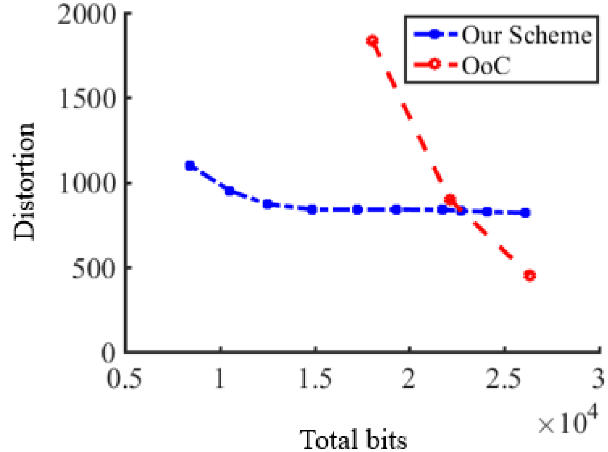
1. Interpolate i^{th} iteration samples (black circles) to a **continuous kernel** (mesh), an approximation of the target surface $\boldsymbol{S}$.
2. New sample locations, **knots** (squares), are located on the kernel surface.
3. **Signed distances** between knots and $\boldsymbol{S}$ are recorded as sample values.
4. **Sample values** (green circles) are encoded as a **graph-signal via GFT**.

Graph-Signal Sampling / Encoding for 3D Point Cloud

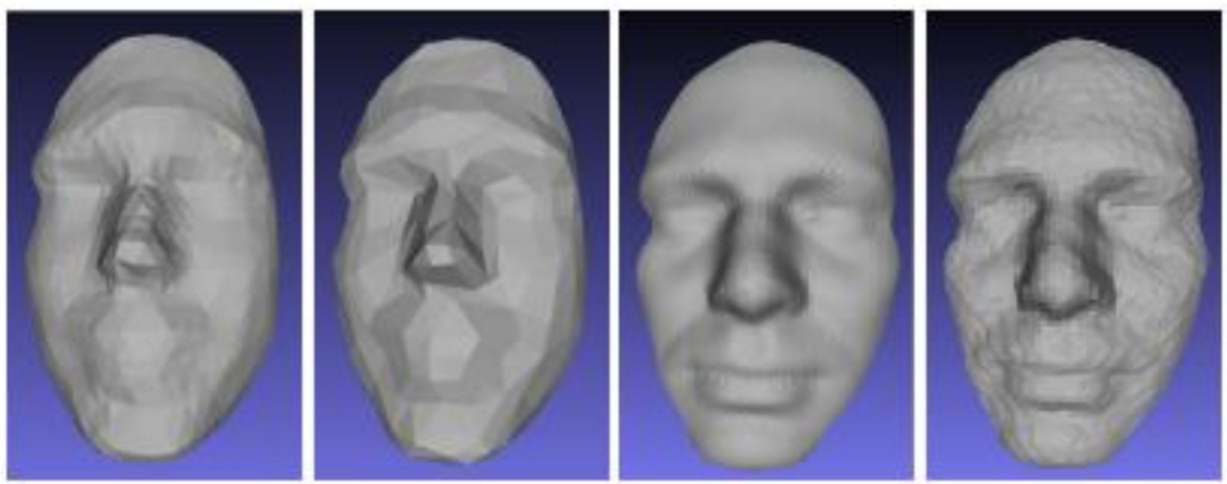
- **Experimental Results**:



(a) Dataset1



(b) Dataset2

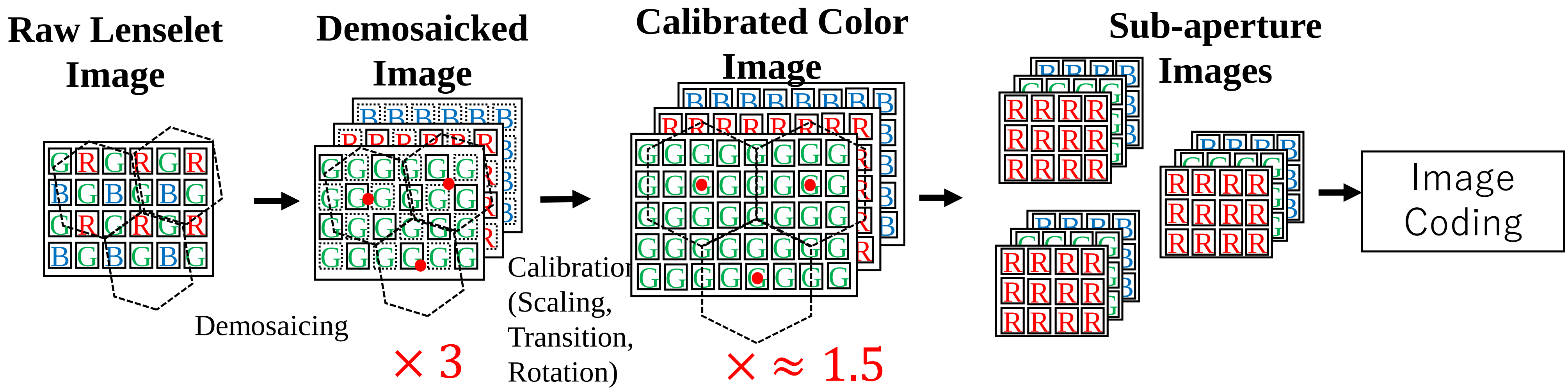


(a) (b)

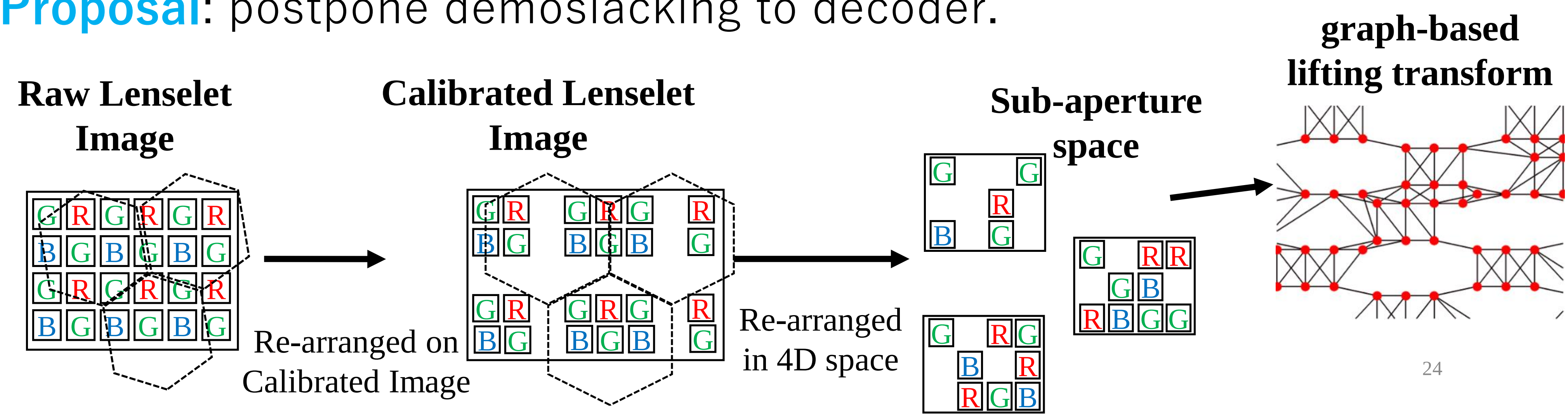
[1] M. Zhao, G. Cheung, D. Florencio, X. Ji, "**Progressive Graph-Signal Sampling and Encoding for Static 3D Geometry Representation**," *IEEE International Conference on Image Processing*, Beijing, China, September, 2017.

Pre-Demosiac Light Field Image Compression Using Graph Lifting Transform

- **Problem**: Sub-aperture images in Light field data are huge.



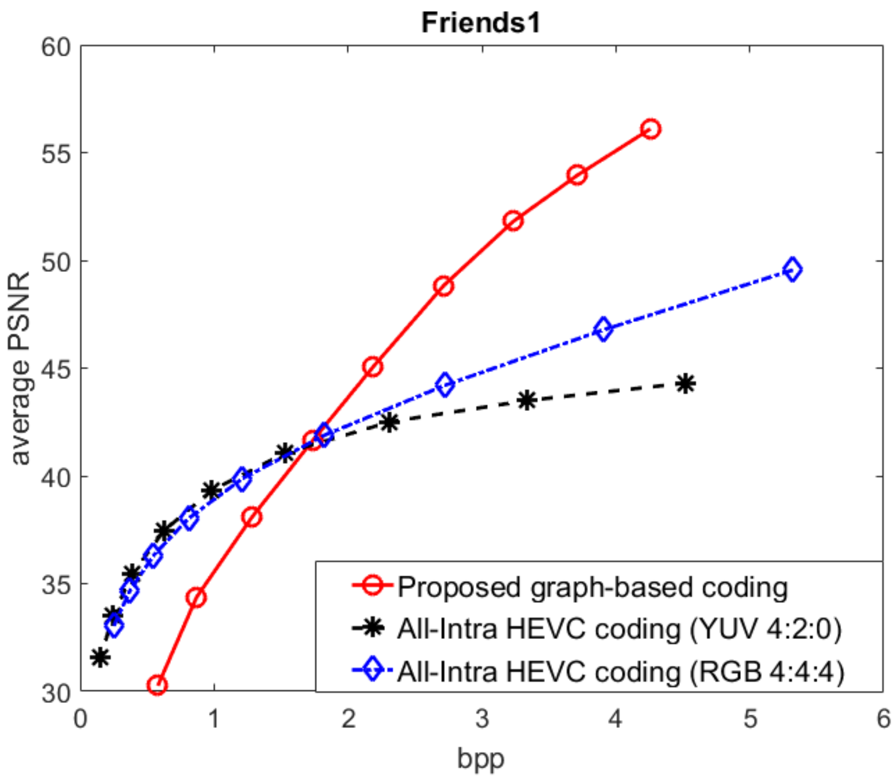
- **Proposal**: postpone demosiacking to decoder.

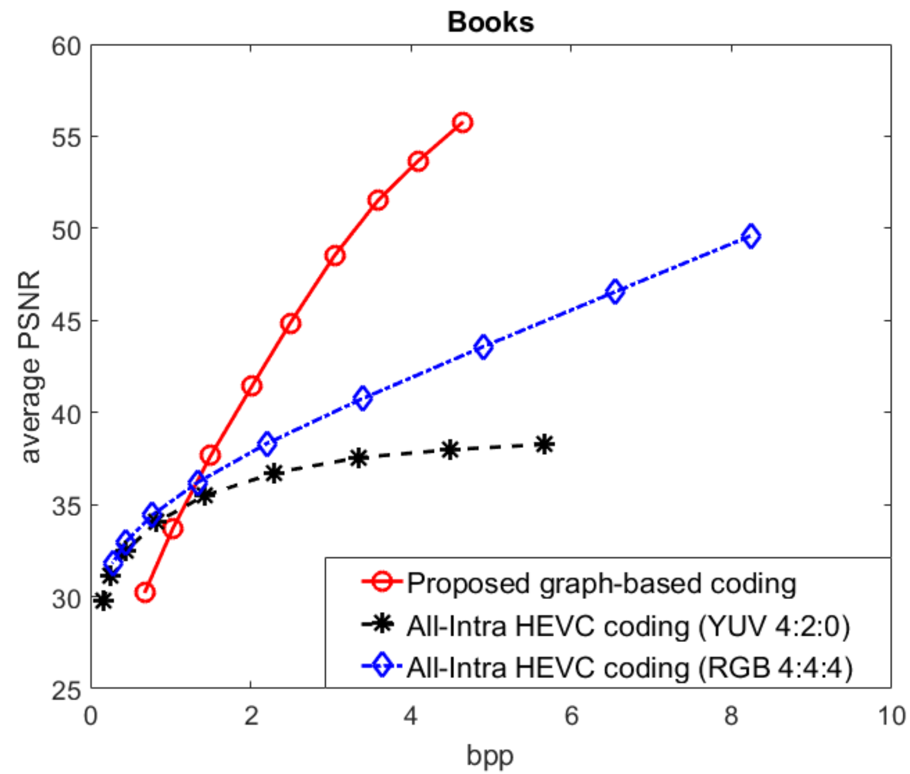


Pre-Demosiac Light Field Image Compression Using Graph Lifting Transform

- **Experimental Results**:

Dataset: EPFL light field image dataset
Baseline: All-intra HEVC coding in YUV4:2:0 and RGB 4:4:4





[1] Y.-H. Chao, G. Cheung, A. Ortega, "**Pre-Demosiac Light Field Image Compression Using Graph Lifting Transform**," *IEEE Int'l Conf. on Image Processing*, Beijing, China, September, 2017. (**Best student paper award**)

Outline

- GSP Fundamentals
- GSP for Image Compression
    - Optimality of GFT
- GSP for Inverse Imaging
    - Graph Laplacian Regularizer
        - Image denoising, contrast enhancement
    - Reweighted Graph TV
- Deep GLR
- Ongoing & Future Work

Graph Laplacian Regularizer

- $\mathbf{x}^T \mathbf{L}\,\mathbf{x}$ (graph Laplacian regularizer) [1]) is one smoothness measure.

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \frac{1}{2} \sum_{i,j} w_{i,j} \left(x_i - x_j \right)^2 = \sum_k \lambda_k \tilde{x}_k^2$$

signal smooth in nodal domain

signal contains mostly low graph freq.

- **Signal Denoising**:

observation

desired signal

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

noise

- **MAP Formulation**:

fidelity term

$$\min_{x} \left\| \mathbf{y} - \mathbf{x} \right\|_2^2 + \mu\, \mathbf{x}^T \mathbf{L} \mathbf{x}$$

smoothness prior

update edge weights

$$\left(\mathbf{I} + \mu \mathbf{L} \right) \mathbf{x}^* = \mathbf{y}$$

linear system of eqn's w/ sparse, symmetric PD matrix

[1] P. Milanfar, "**A Tour of Modern Image Filtering: New Insights and Methods, Both Practical and Theoretical**," *IEEE Signal Processing Magazine*, vol.30, no.1, pp.106-128, January 2013.

Graph Laplacian Regularizer

- $\mathbf{x}^T \mathbf{L} \mathbf{x}$ (graph Laplacian regularizer) [1]) is one smoothness measure.

$$\mathbf{x}^T \mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{i,j} w_{i,j}\left(x_i - x_j\right)^2 = \sum_k \lambda_k \tilde{x}_k^2$$

signal smooth in nodal domain

signal contains mostly low graph freq.

- **Signal Denoising**:

- **MAP Formulation**:

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

$$\min_x \left\|\mathbf{y} - \mathbf{x}\right\|_2^2 + \mu\, \mathbf{x}^T \mathbf{L}\mathbf{x}$$

$$\left(\mathbf{I} + \mu \mathbf{L}\right)\mathbf{x}^* = \mathbf{y}$$

update edge weights

linear system of eqn's w/ sparse, symmetric PD matrix

pixel intensity diff.

pixel location diff.

$$w_{i,j} = \exp\left(\frac{-\left\|x_i - x_j\right\|_2^2}{\sigma_1^2}\right)\exp\left(\frac{-\left\|l_i - l_j\right\|_2^2}{\sigma_2^2}\right)$$

Bilateral filter weights

[1] P. Milanfar, "**A Tour of Modern Image Filtering: New Insights and Methods, Both Practical and Theoretical**," *IEEE Signal Processing Magazine*, vol.30, no.1, pp.106-128, January 2013.

Optimal Graph Laplacian Regularization for Denoising

- 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 **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.

Spatial

$$\mathbf{f}_1^D(i) = \sqrt{\sigma^2 + \alpha} \cdot x_i$$

$$\mathbf{f}_2^D(i) = \sqrt{\sigma^2 + \alpha} \cdot y_i$$

Intensity

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

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

[1] J. Pang, G. Cheung, "**Graph Laplacian Regularization for Inverse Imaging: Analysis in the Continuous Domain**," *IEEE Transactions on Image Processing*, vol. 26, no.4, pp.1770-1785, April 2017.

Denoising Experiments (natural images)

- Subjective comparisons ($\sigma_{\mathrm{I}} = 40$)

Original

Noisy, 16.48 dB

K-SVD, 26.84 dB

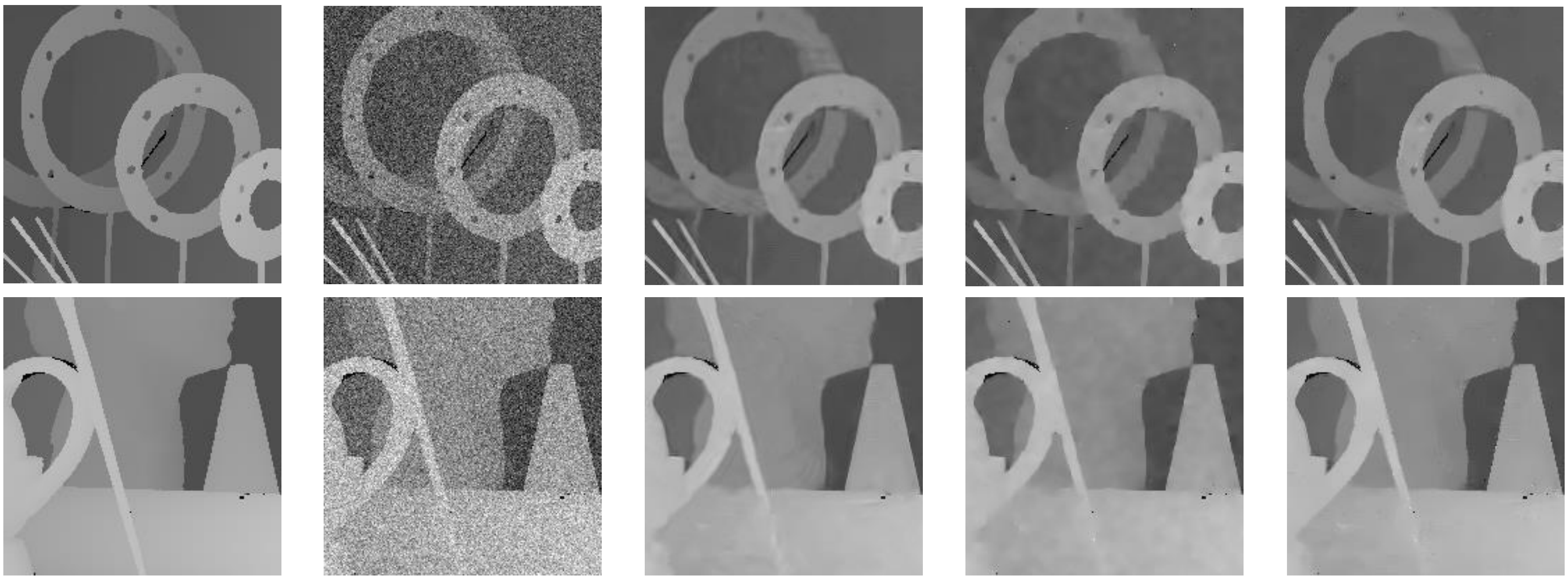
BM3D, 27.99 dB

PLOW, 28.11 dB

OGLR, 28.35 dB

Denoising Experiments (depth images)

- Subjective comparisons ($\sigma_I = 30$)



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

[1] W. Hu et al., "**Depth Map Denoising using Graph-based Transform and Group Sparsity**," *IEEE International Workshop on Multimedia Signal Processing*, Pula (Sardinia), Italy, October, 2013.

GLR for Joint Dequantization / Contrast Enhancement

- Retinex decomposition model:

$$\mathbf{y} = \underbrace{\tau}_{\text{scalar}}\, \underbrace{\mathbf{l}}_{\text{illumination}} \odot \overbrace{\mathbf{r}}^{\text{reflectance}} + \overbrace{\mathbf{z}}^{\text{noise}}$$

- **Objective**: general smoothness for luminance, smoothness w/ negative edges for reflectance.

$$\begin{aligned} \min_{\mathbf{l},\mathbf{r}} \quad & \mathbf{l}^\top \left(\mathbf{L}_l + \alpha \mathbf{L}_l^2\right) \mathbf{l} + \mu\, \mathbf{r}^\top \mathcal{L}_r \mathbf{r} \\ \text{s.t.} \quad & \left(\mathbf{q} - \tfrac{1}{2}\right) \mathbf{Q} \preceq \mathbf{T} \tau\, \mathbf{l} \odot \mathbf{r} \prec \left(\mathbf{q} + \tfrac{1}{2}\right) \mathbf{Q} \end{aligned}$$

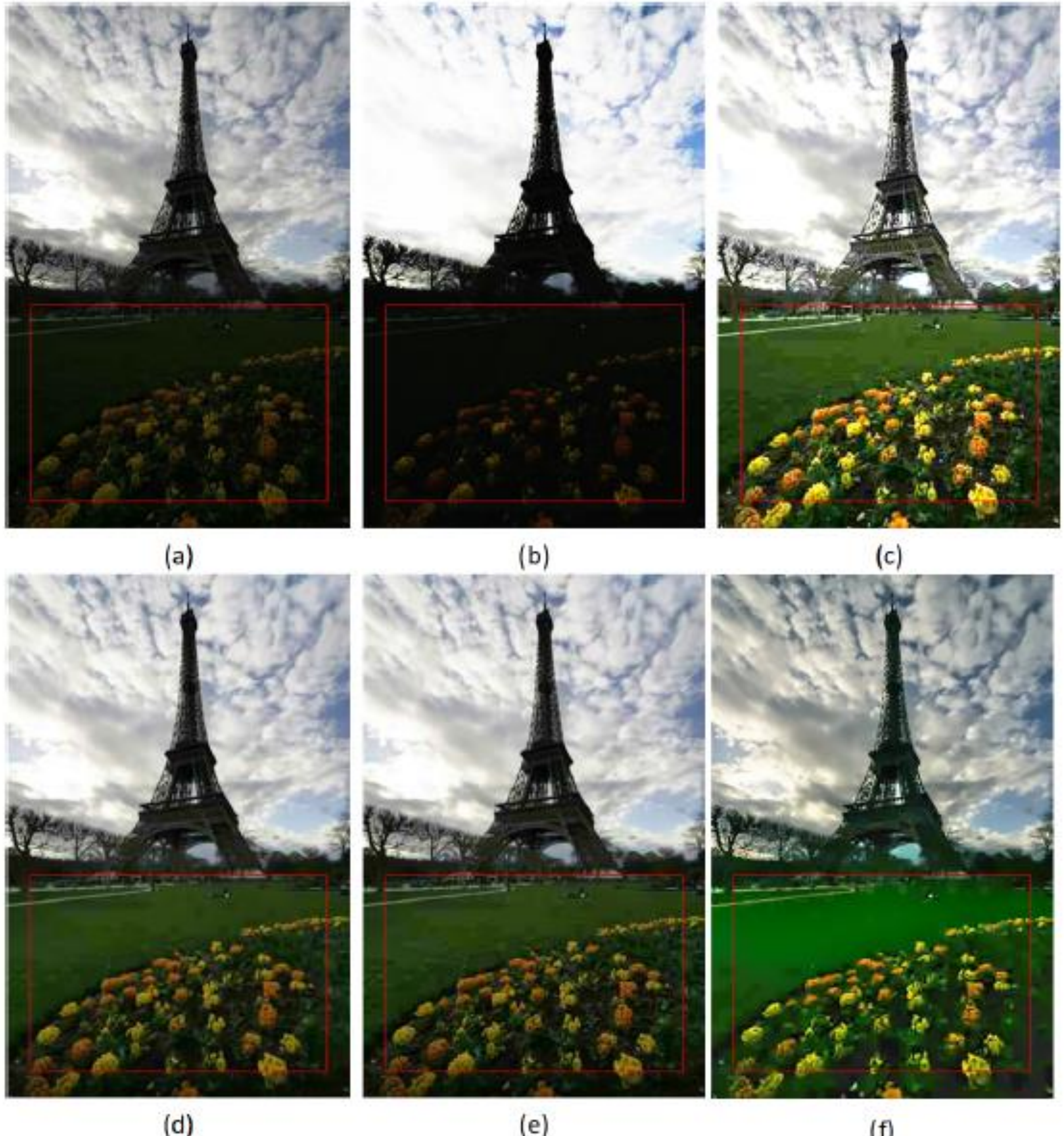
- **Constraints:** quantization bin constraints

- **Solution**: Alternating accelerated proximal gradient alg [1].

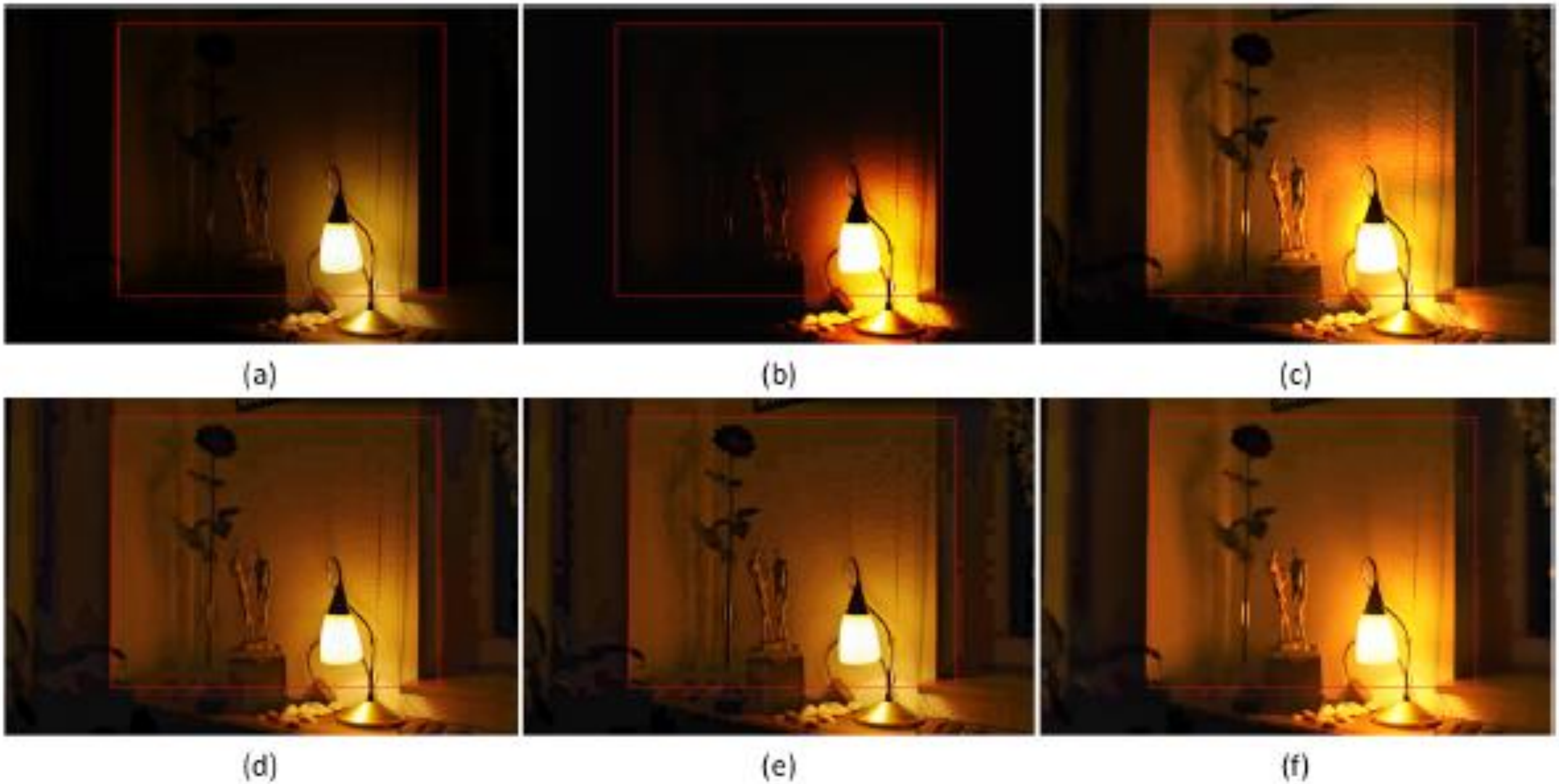
[1] Xianming Liu, Gene Cheung, Xiangyang Ji, Debin Zhao, Wen Gao, "**Graph-based Joint Dequantization and Contrast Enhancement of Poorly Lit JPEG Images**," accepted to *IEEE Transactions on Image Processing*, September 2018.

Experimental Results

Experimental Results



Experimental Results



(a) (b) (c) (d) (e) (f)

Outline

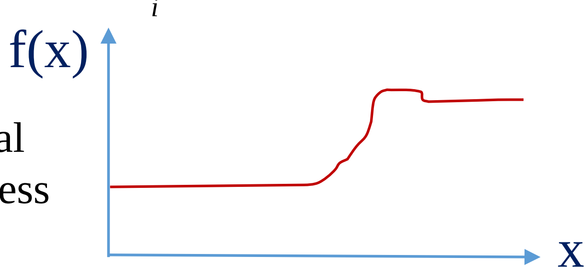
- GSP Fundamentals
- GSP for Image Compression
  - Optimality of GFT
- GSP for Inverse Imaging
  - Graph Laplacian Regularizer
  - Reweighted Graph TV
    - 3D point cloud denoising, image deblurring
- Deep GLR
- Ongoing & Future Work

GTV for Point Cloud Denoising

- Acquisition of point cloud introduces noise.
- Point cloud is irregularly sampled 2D manifold in 3D space.
- Not appropriate to apply GTV directly on 3D coordinates [1].
  - only **a singular 3D point has zero GTV value**.

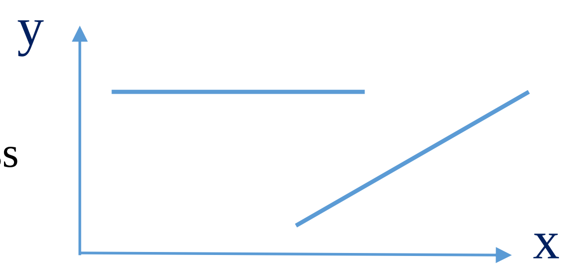
$$\sum_i \left| f(x_i) - f(x_{i-1}) \right|$$

functional smoothness



$$\sum_i \left| y_i - y_{i-1} \right|$$

geometry smoothness



- **Proposal**: Apply GTV is to the surface normals of 3D point cloud—**a generalization of TV to 3D geometry.**

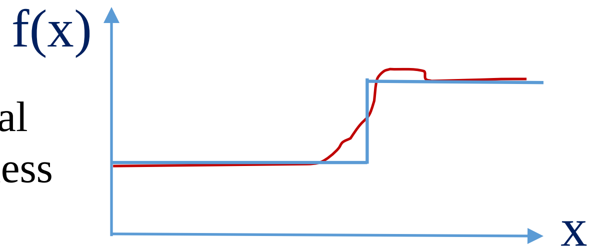
[1] Y. Schoenenberger, J. Paratte, and P. Vandergheynst, "**Graph-based denoising for time-varying point clouds**," in *IEEE 3DTV-Conference*, 2015, pp. 1–4

GTV for Point Cloud Denoising

- Acquisition of point cloud introduces noise.
- Point cloud is irregularly sampled 2D manifold in 3D space.
- Not appropriate to apply GTV directly on 3D coordinates [1].
  - only **a singular 3D point has zero GTV value**.

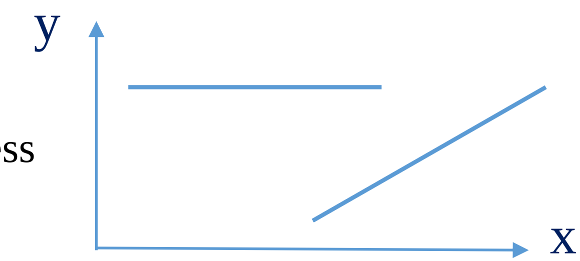
$$\sum_i \left| f(x_i) - f(x_{i-1}) \right|$$

functional smoothness



$$\sum_i \left| y_i - y_{i-1} \right|$$

geometry smoothness



- **Proposal**: Apply GTV is to the surface normals of 3D point cloud—**a generalization of TV to 3D geometry.**

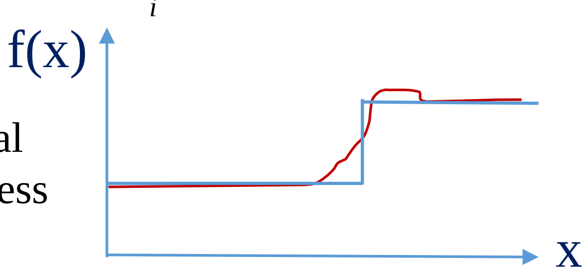
[1] Y. Schoenenberger, J. Paratte, and P. Vandergheynst, "**Graph-based denoising for time-varying point clouds**," in *IEEE 3DTV-Conference*, 2015, pp. 1–4

GTV for Point Cloud Denoising

- Acquisition of point cloud introduces noise.
- Point cloud is irregularly sampled 2D manifold in 3D space.
- Not appropriate to apply GTV directly on 3D coordinates [1].
  - only **a singular 3D point has zero GTV value**.

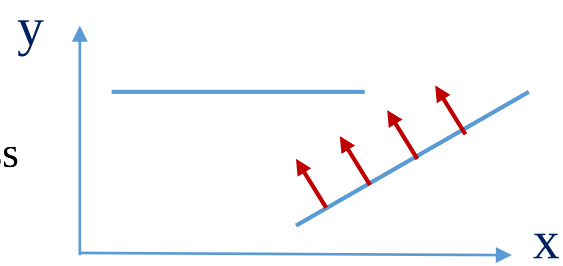
$$\sum_i \left| f(x_i) - f(x_{i-1}) \right|$$

functional smoothness



$$\sum_i \left| y_i - y_{i-1} \right| \longrightarrow \sum_i \left\| \mathrm{n}_i - \mathrm{n}_{i-1} \right\|_1$$

geometry smoothness



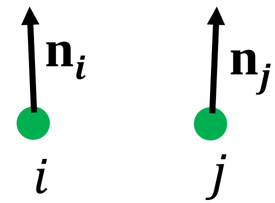
- **Proposal**: Apply GTV is to the surface normals of 3D point cloud—**a generalization of TV to 3D geometry.**

[1] Y. Schoenenberger, J. Paratte, and P. Vandergheynst, "**Graph-based denoising for time-varying point clouds**," in *IEEE 3DTV-Conference*, 2015, pp. 1–4

Algorithm Overview

- Use graph total variation (GTV) of surface normals over the K-NN graph:

$$||\mathbf{n}||_{\text{GTV}} = \sum_{i,j \in \mathcal{E}} w_{i,j} ||\mathbf{n}_i - \mathbf{n}_j||_1$$



$$w_{i,j} = \exp\left(-\frac{||\mathbf{p}_i - \mathbf{p}_j||_2^2}{\sigma_p^2}\right)$$

- Denoising problem as l2-norm fidelity plus GTV of surface normals:

$$\min_{\mathrm{p},\mathrm{n}} \|\mathrm{q} - \mathrm{p}\|_2^2 + \gamma \sum_{i,j \in E} w_{i,j} \|\mathrm{n}_i - \mathrm{n}_j\|_1$$

- Surface normal estimation of $\mathbf{n}_i$ is a nonlinear function of $\mathbf{p}_i$ and neighbors.

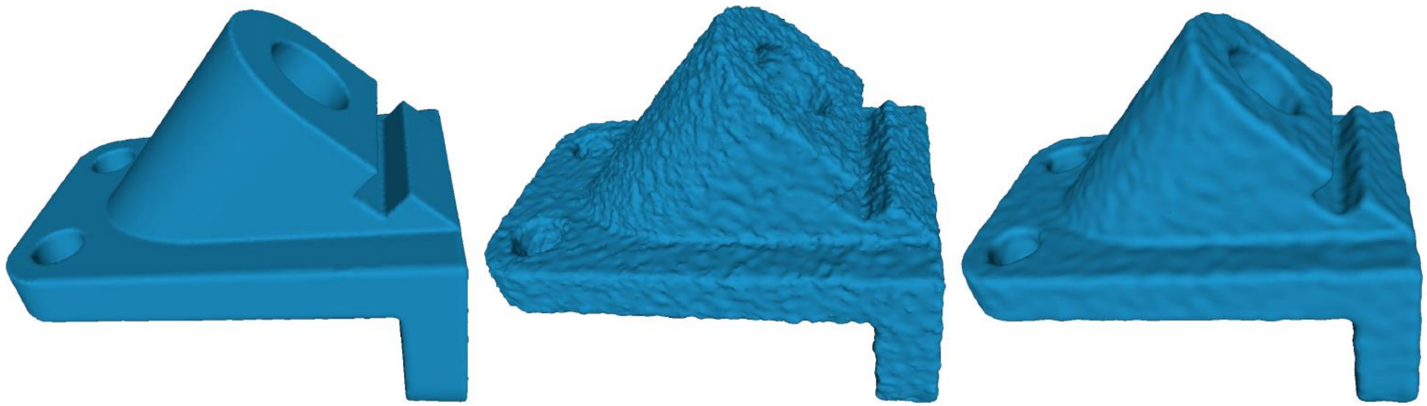
Proposal:

1. Partition point cloud into **two independent classes** (say red and blue).
2. When computing surface normal for a red node, use only neighboring blue points.
3. Solve convex optimization for red (blue) nodes alternately.

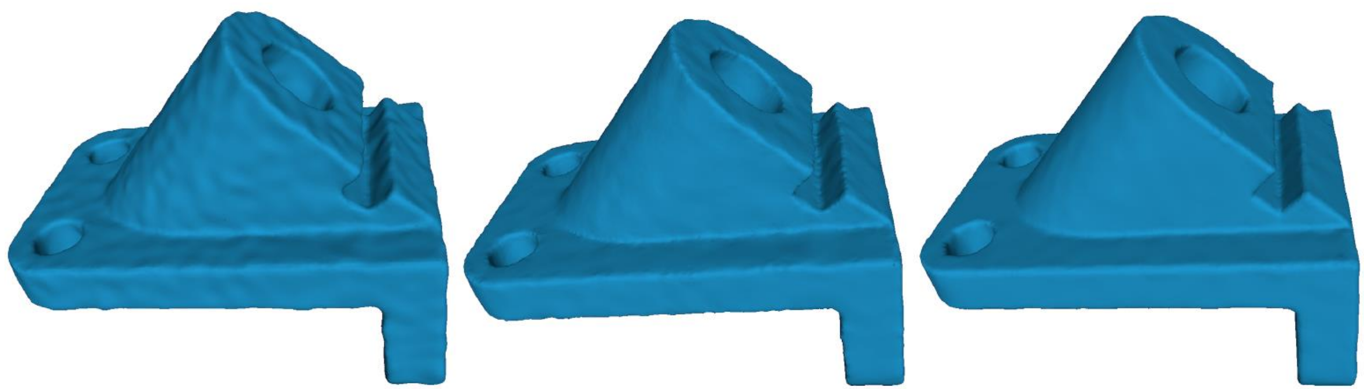
[1] C. Dinesh, G. Cheung, I. V. Bajic, C. Yang, "Fast 3D Point Cloud Denoising via Bipartite Graph Approximation & Total Variation," *IEEE 20th International Workshop on Multimedia Signal Processing*, Vancouver, Canada, August 2018.

Experimental Results – Visual Comparison

Anchor model (σ=0.3)



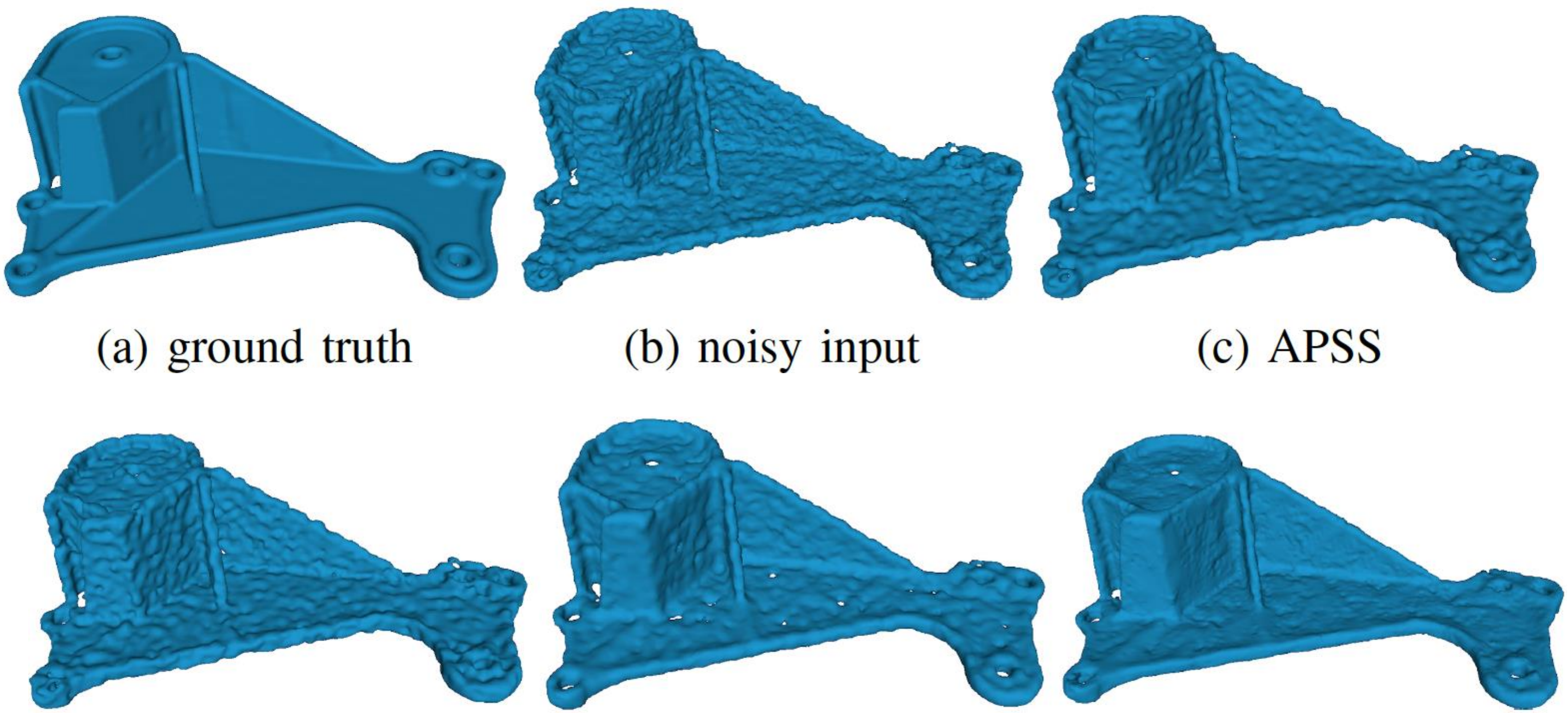
(a) ground truth (b) noisy input (c) APSS



(d) RIMLS (e) MRPCA (f) proposed

Experimental Results – Visual Comparison

Daratech model (σ=0.3)



(a) ground truth (b) noisy input (c) APSS

(d) RIMLS (e) MRPCA (f) proposed

Reweighted Graph Total Variation

- TV on graphs.

pixel intensity difference

$$w_{i,j} = \exp\left(\frac{-\|x_i - x_j\|_2^2}{\sigma_1^2} \right)$$

Gradient of nodes on the graph:

$$(\nabla_i \mathbf{x})_j \triangleq x_j - x_i,$$

Reweighted Graph TV:

$$\|\mathbf{x}\|_{RGTV} = \sum_{i \in \mathcal{V}} \| \operatorname{diag}(\mathbf{W}_{i,\cdot}(\mathbf{x})) \nabla_i \mathbf{x} \|_1$$
$$= \sum_{i=1}^{N} \sum_{j=1}^{N} w_{i,j}(x_i, x_j) |x_j - x_i|,$$

[1] M. Hidane, O. Lezoray, and A. Elmoataz, "Nonlinear multilayered representation of graph-signals," in *Journal of Mathematical Imaging and Vision*, February 2013, vol. 45, no.2, pp. 114–137.

[2] P. Berger, G. Hannak, and G. Matz, "Graph signal recovery via primal-dual algorithms for total variation minimization," in *IEEE Journal on Selected Topics in Signal Processing*, September 2017, vol. 11, no.6, pp. 842–855.

Background for Image Deblurring

- Image blur is a common image degradation.
- Typically, blur process is modeled:

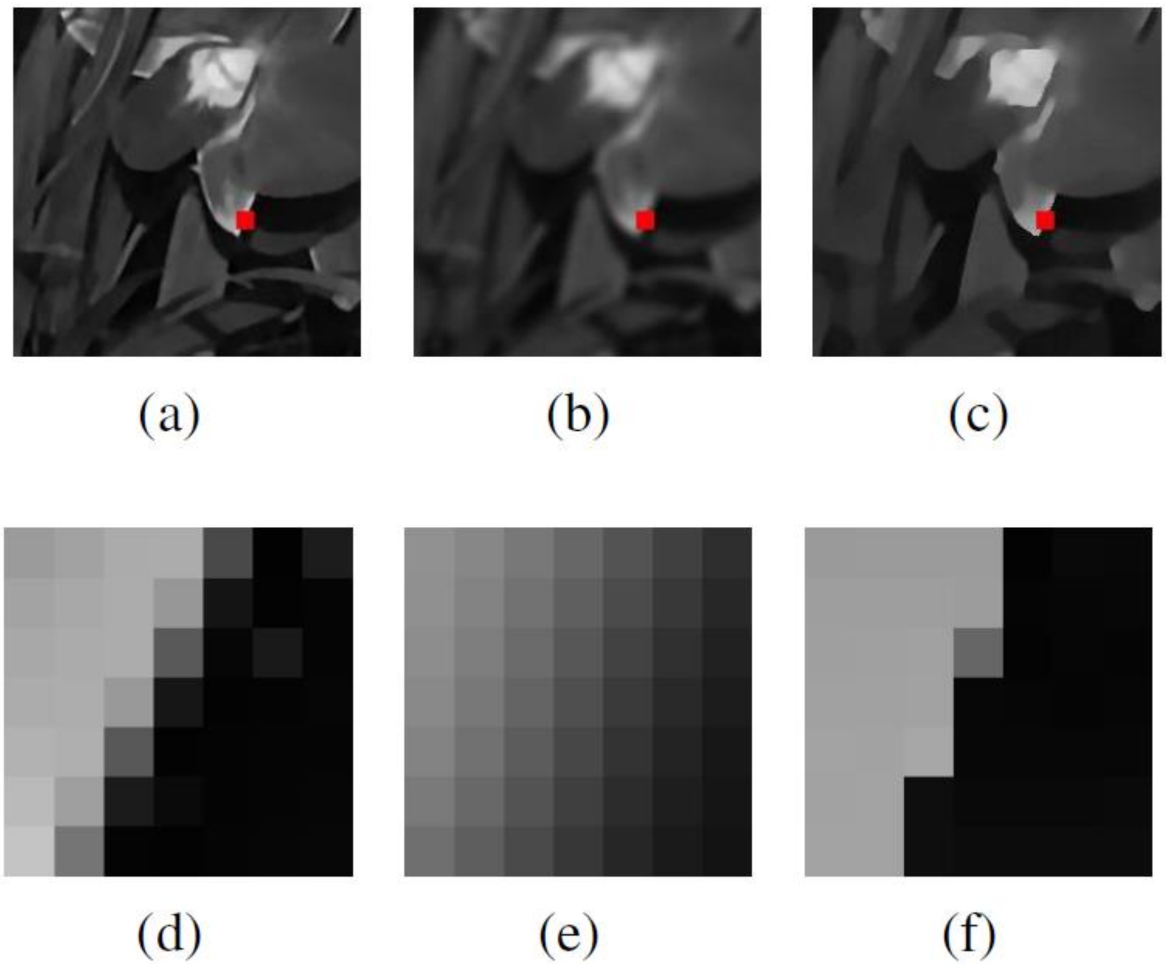
$$y = k \otimes x$$

where y is the blurry image, k is the blur kernel, x is the original sharp image.

- **Blind-image deblurring** focuses on estimating blur kernel k.
- Given k, problem becomes ***de-convolution***.

Observation

- ***Skeleton image***:
  - PWS image keeping only structural edges.
  - Proxy to estimate blur kernel *k*.

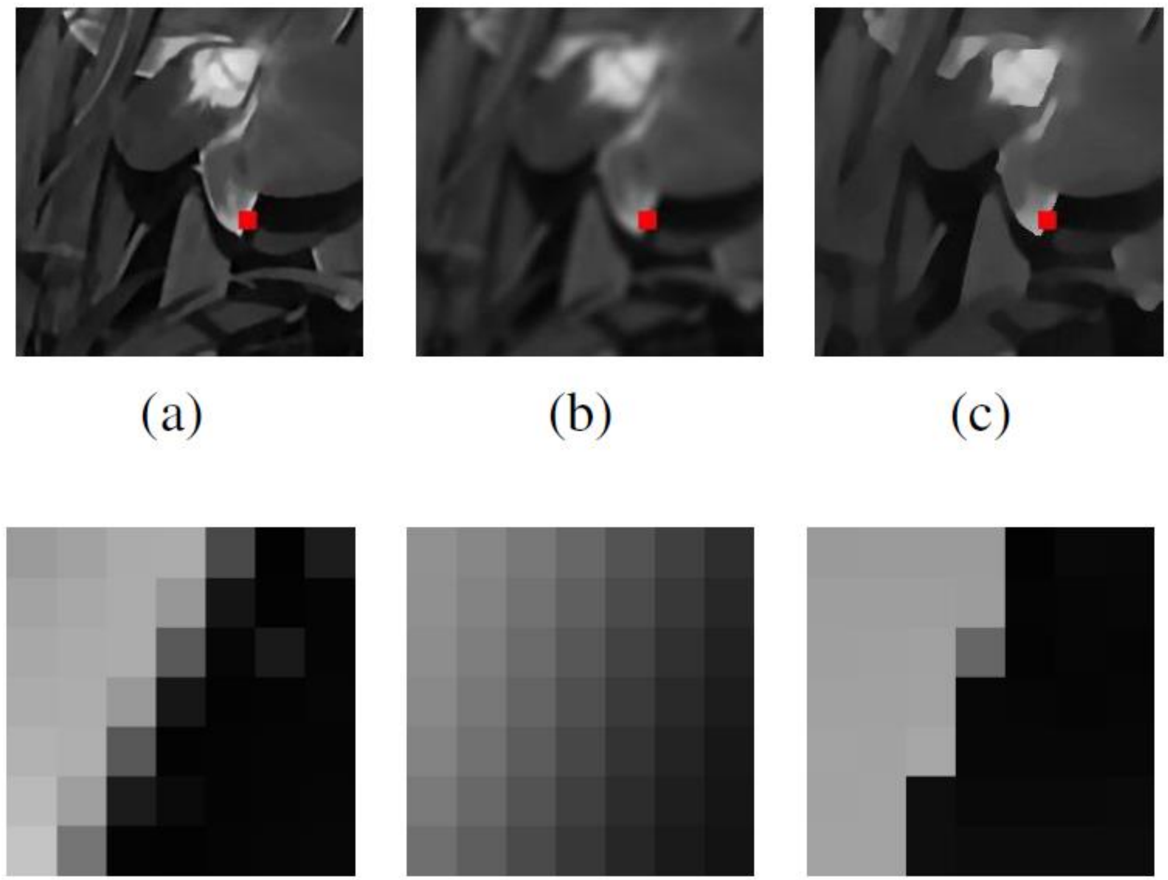


(a) (b) (c)

(d) (e) (f)

Observation

- ***Skeleton image***:
  - PWS image keeping only structural edges.
  - Proxy to estimate blur kernel k.



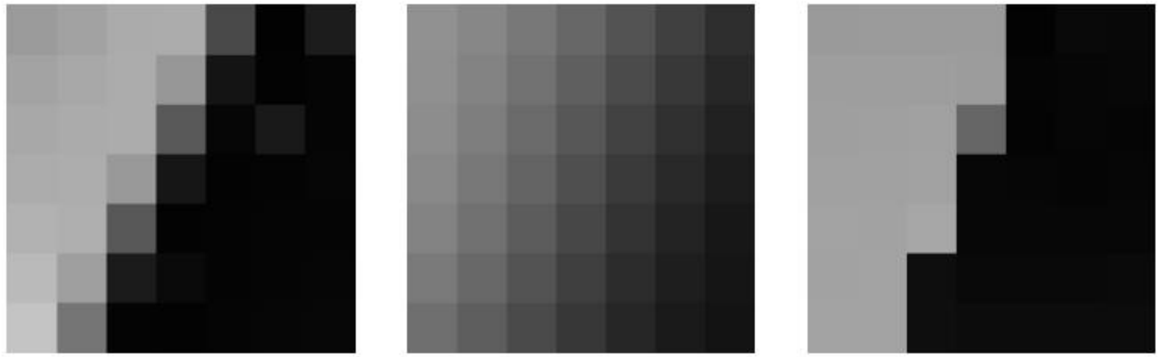
(a) (b) (c)

(d) (e) (f)

Observation

- ***Skeleton image***:
  - PWS image keeping only structural edges.
  - Proxy to estimate blur kernel *k*.

(a) (b) (c)



(d) (e) (f)

Our algorithm

- The optimization function can be written as follows,

$$\hat{\mathbf{x}}, \hat{\mathbf{k}} = \underset{\mathbf{x},\mathbf{k}}{\operatorname{argmin}}\, \varphi(\mathbf{x} \otimes \mathbf{k} - \mathbf{b}) + \mu_1 \cdot \theta_x(\mathbf{x}) + \mu_2 \cdot \theta_k(\mathbf{k})$$

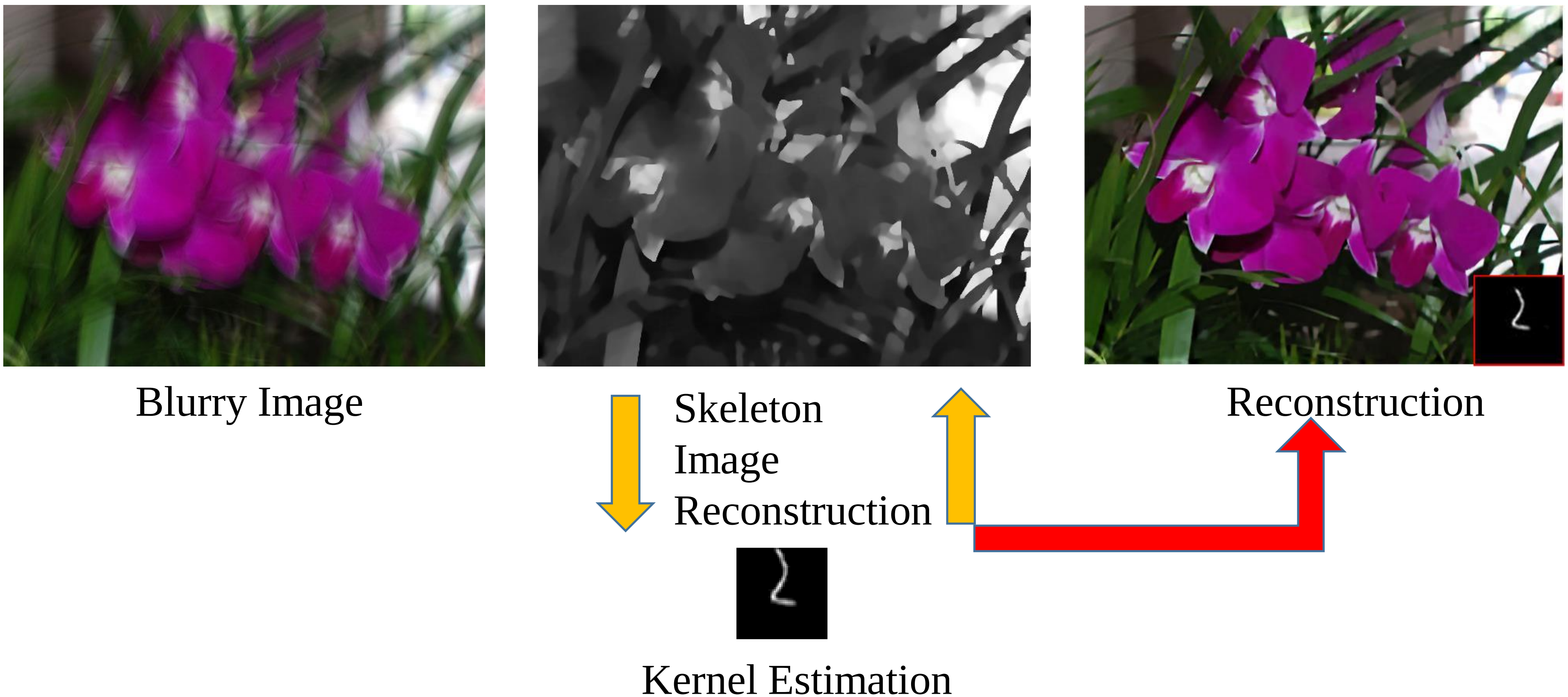
- Assume L_2 norm for fidelity term $\varphi(\cdot)$.
- $\theta_x(\cdot) = RGTV(\cdot)$.
- $\theta_k(\cdot) = ||\cdot||_2$, assuming zero mean Gaussian distribution of k.

- RGTV is non-differentiable and non-convex.

Solution:

- Solve x and k alternatingly.
- For x, spectral interpretation of GTV, fast spectral filter.

Workflow



Blurry Image
Skeleton Image Reconstruction
Kernel Estimation
Reconstruction

Experimental Results

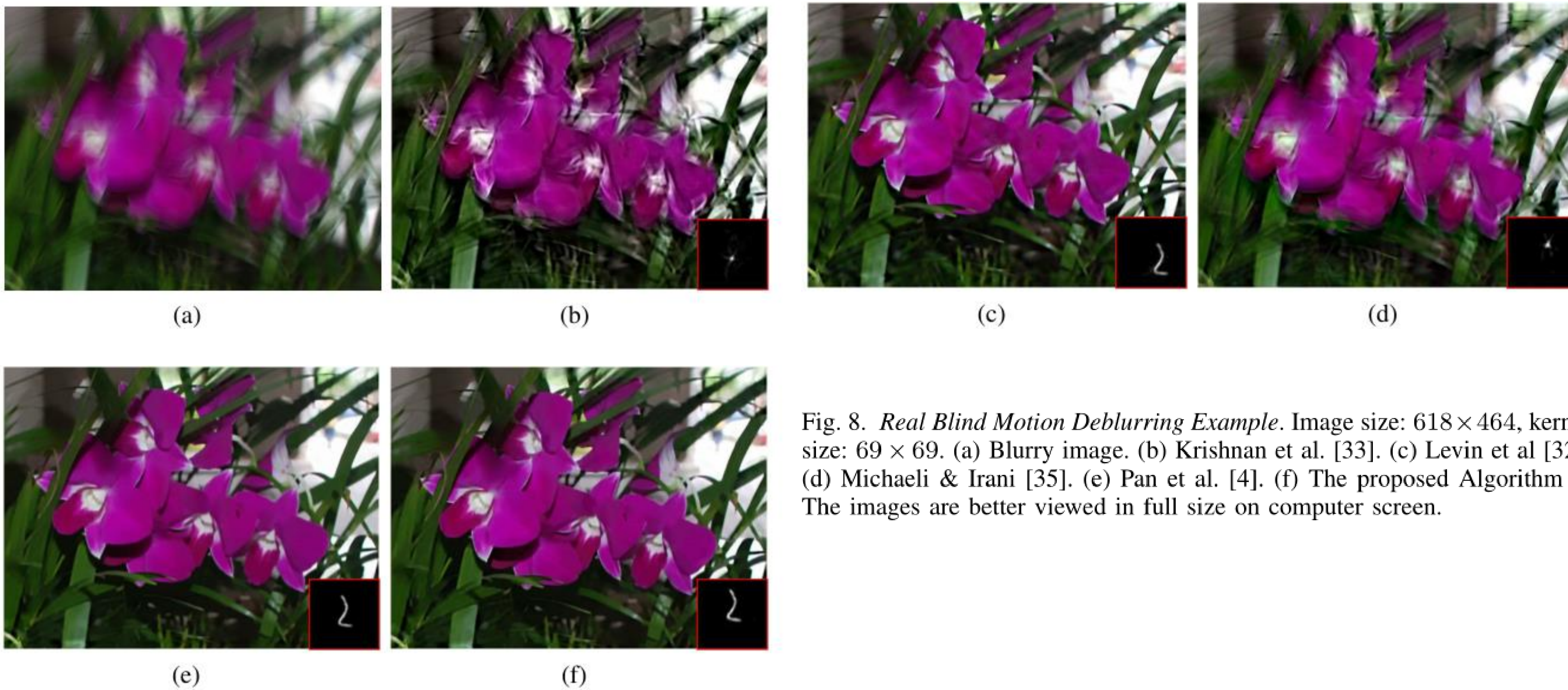
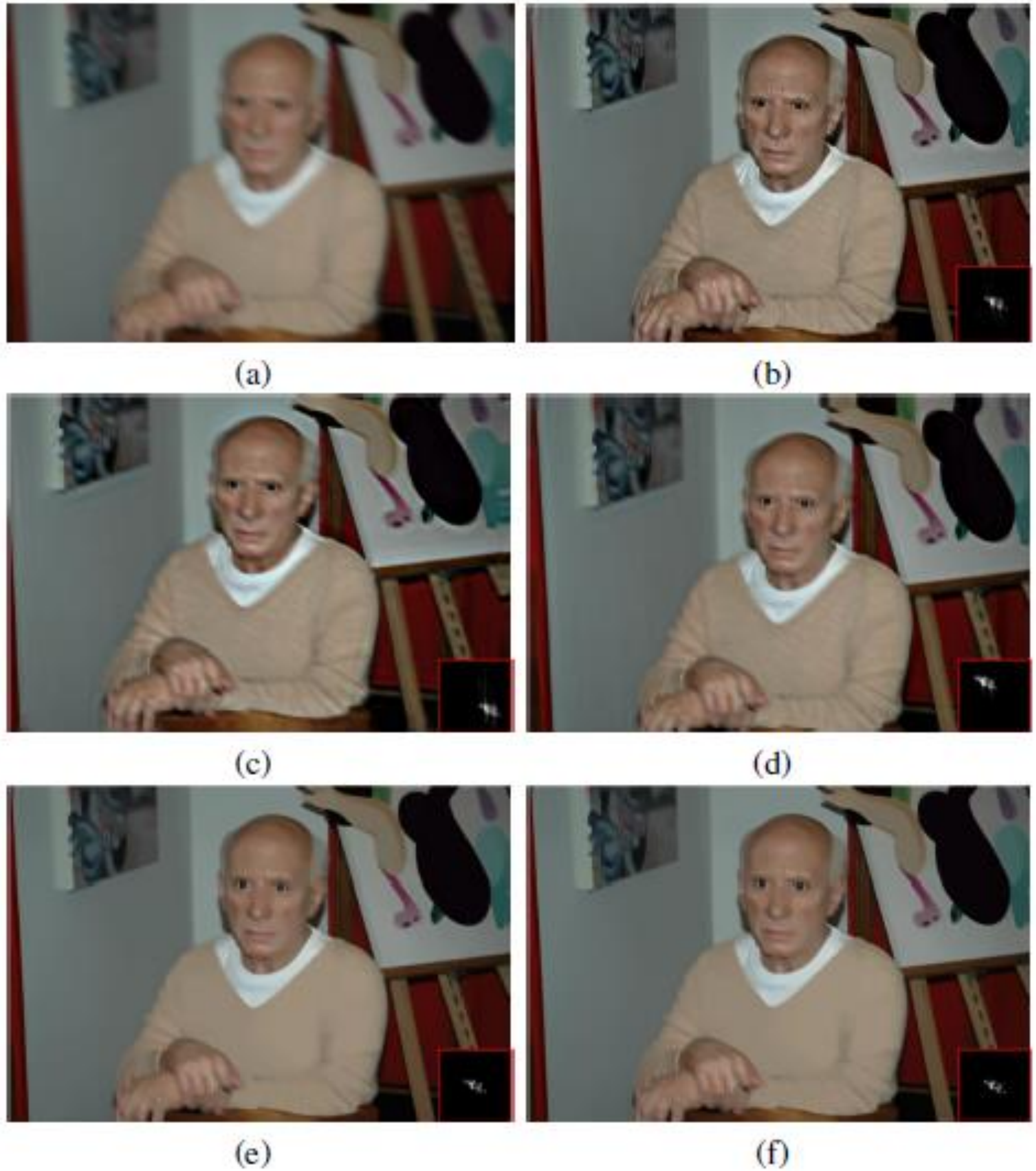


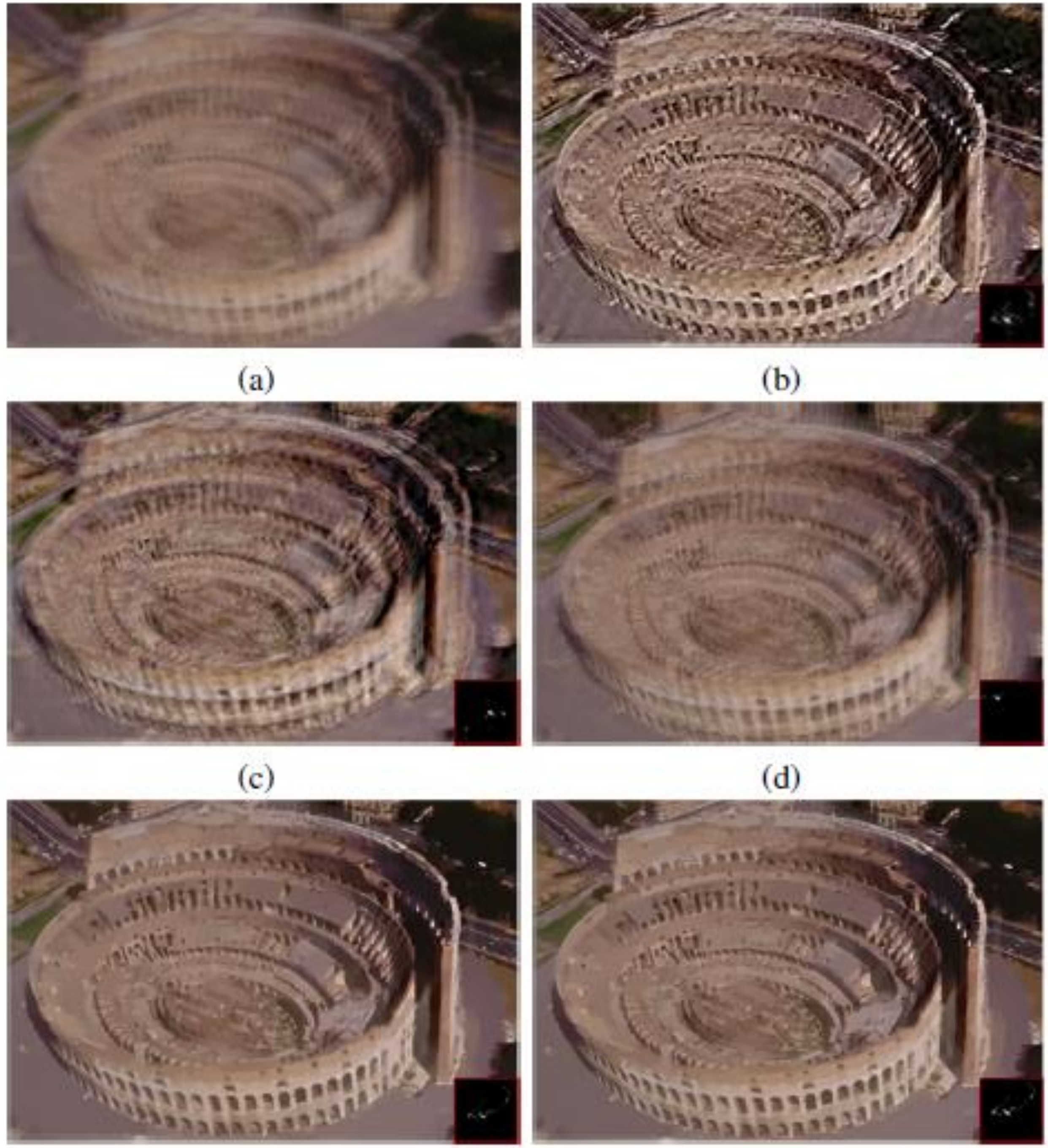
Fig. 8. *Real Blind Motion Deblurring Example*. Image size: 618×464, kernel size: 69×69. (a) Blurry image. (b) Krishnan et al. [33]. (c) Levin et al [32]. (d) Michaeli & Irani [35]. (e) Pan et al. [4]. (f) The proposed Algorithm 1. The images are better viewed in full size on computer screen.

[1] Y. Bai, G. Cheung, X. Liu, W. Gao, "**Graph-Based Blind Image Deblurring from a Single Photograph**," accepted to *IEEE Transactions on Image Processing*, October 2018.

Experimental Results



Experiment Results



(a) (b) (c) (d) (e) (f)

Outline

- GSP Fundamentals
- GSP for Image Compression
    - Optimality of GFT
- GSP for Inverse Imaging
    - Graph Laplacian Regularizer
    - Reweighted Graph TV
- Deep GLR
- Ongoing & Future Work

Unrolling Graph Laplacian Regularizer

- Recall MAP formulation of denoising problem with quadratic graph Laplacian regularizer:

$$\min_{x} \|\mathrm{y}-\mathrm{x}\|_2^2 + \mu\, \mathrm{x}^T \mathrm{L}\mathrm{x}$$

fidelity term

smoothness prior

- Solution is system of linear equations:

$$(\mathrm{I}+\mu\mathrm{L})\,\mathrm{x}^* = \mathrm{y}$$

linear system of eqn's w/ sparse, symmetric PD matrix

[1] J. Pang, G. Cheung, "**Graph Laplacian Regularization for Inverse Imaging: Analysis in the Continuous Domain**," *IEEE Transactions on Image Processing*, vol. 26, no.4, pp.1770-1785, April 2017.

Unrolling Graph Laplacian Regularizer

- Recall MAP formulation of denoising problem with quadratic graph Laplacian regularizer:

$$\min_{x} \underbrace{\|\mathrm{y}-\mathrm{x}\|_2^2}_{\text{fidelity term}} + \underbrace{\mu\, \mathrm{x}^T \mathrm{L}\mathrm{x}}_{\text{smoothness prior}}$$

- Solution is system of linear equations:

$$\underbrace{(\mathrm{I}+\mu\mathrm{L})}_{\text{linear system of eqn's w/ sparse, symmetric PD matrix}}\mathrm{x}^* = \mathrm{y}$$

Q: what is the "most appropriate" graph?

[1] J. Pang, G. Cheung, "**Graph Laplacian Regularization for Inverse Imaging: Analysis in the Continuous Domain**," *IEEE Transactions on Image Processing*, vol. 26, no.4, pp.1770-1785, April 2017.

Unrolling Graph Laplacian Regularizer

- Recall MAP formulation of denoising problem with quadratic graph Laplacian regularizer:

$$\min_{x} \| \mathrm{y} - \mathrm{x} \|_2^2 + \mu\, \mathrm{x}^T \mathrm{L} \mathrm{x}$$

fidelity term

smoothness prior

- Solution is system of linear equations:

$$\left(\mathrm{I} + \mu \mathrm{L} \right) \mathrm{x}^* = \mathrm{y}$$

linear system of eqn's w/ sparse, symmetric PD matrix

Q: what is the "most appropriate" graph?

Bilateral weights:

$$w_{i,j} = \exp\left(\frac{-\| x_i - x_j \|_2^2}{\sigma_1^2} \right) \exp\left(\frac{-\| l_i - l_j \|_2^2}{\sigma_2^2} \right)$$

[1] J. Pang, G. Cheung, "**Graph Laplacian Regularization for Inverse Imaging: Analysis in the Continuous Domain**," *IEEE Transactions on Image Processing*, vol. 26, no.4, pp.1770-1785, April 2017.

Unrolling Graph Laplacian Regularizer

- **Deep Graph Laplacian Regularization:**
  1. Learn features **f**'s using CNN.
  2. Compute distance from features.
  3. Compute edge weights using Gaussian kernel.
  4. Construct graph, solve QP.

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

$$\mathrm{dist}(i,j) = \sum_{n=1}^{N} \left(\mathbf{f}_n(i) - \mathbf{f}_n(j)\right)^2.$$

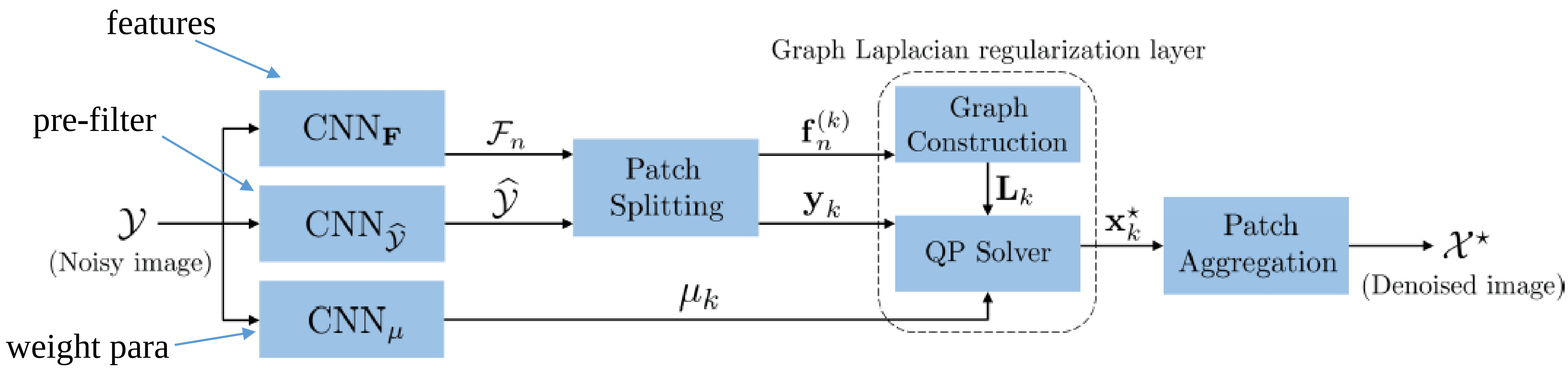


Fig. 1. Block diagram of the proposed GLRNet which employs a graph Laplacian regularization layer for image denoising.

[1] M. McCann et al., "**Convolutional Neural Networks for Inverse Problems in Imaging**," *IEEE SPM*, Nov. 2017.

[2] K. Gregor and Y. LeCun, "**Learning fast approximations of sparse coding,**" in Proc. 27th Int. Conf. Machine Learning, 2010..

Unrolling Graph Laplacian Regularizer

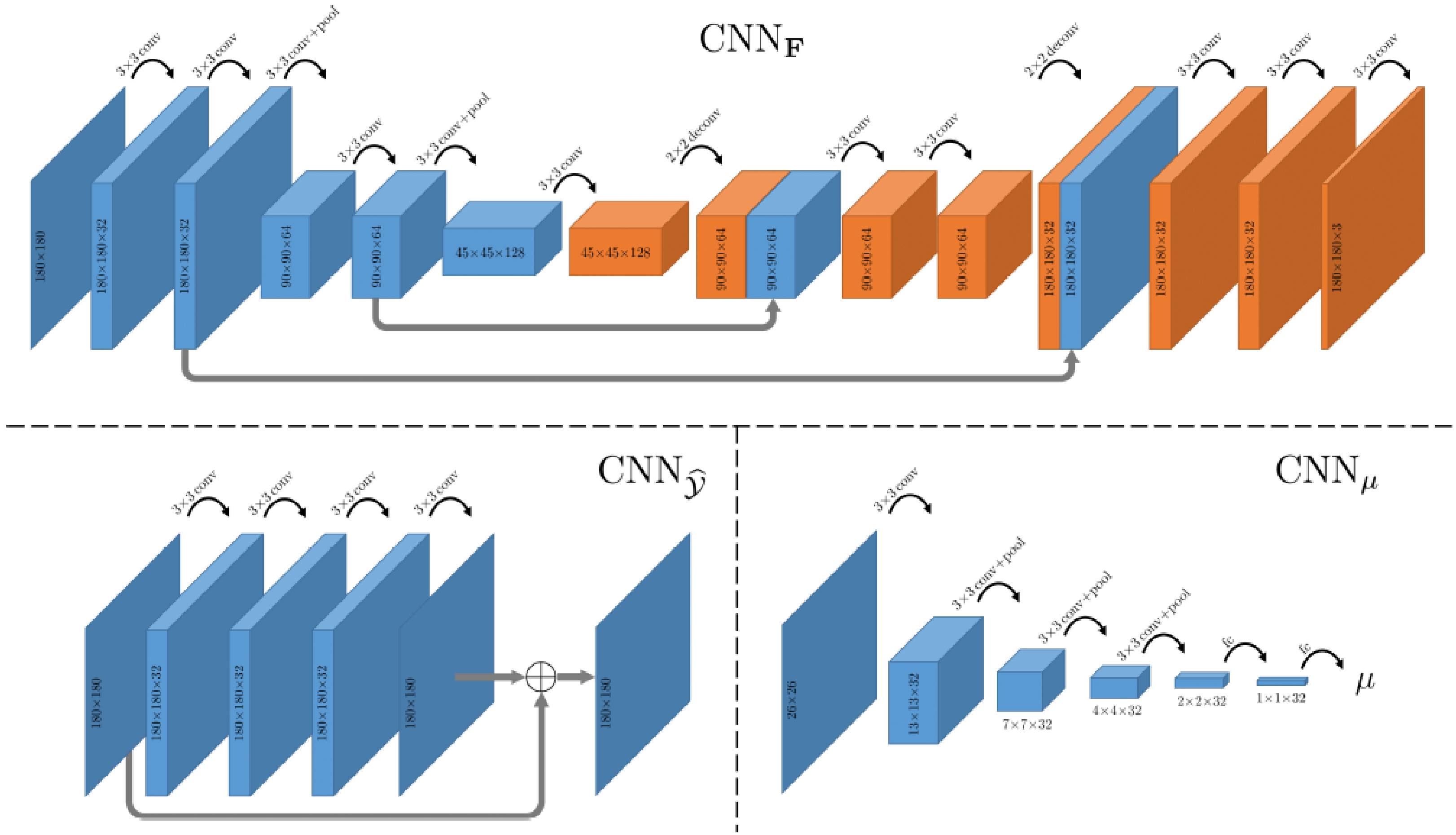


Fig. 3. Network architectures of $\mathrm{CNN}_{\mathbf{F}}$, $\mathrm{CNN}_{\hat{\mathcal{Y}}}$ and CNN_{μ} in the experiments. Data produced by the decoder of $\mathrm{CNN}_{\mathbf{F}}$ is colored in orange.

[1] J. Zeng et al., "**Deep Graph Laplacian Regularization**," submitted to arXiv, July 2018. (https://arxiv.org/abs/1807.11637)

Unrolling Graph Laplacian Regularizer

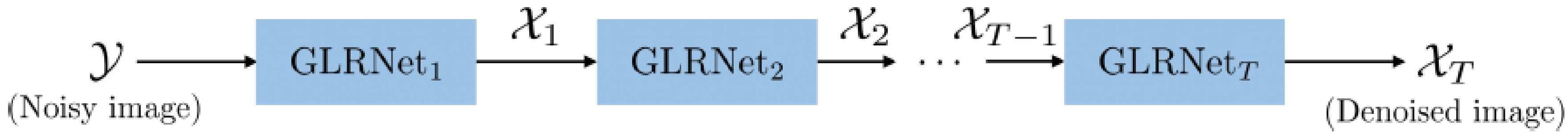


Fig. 2. Block diagram of the overall DeepGLR framework.

- **Graph Model** *guarantees numerical stability of solution*:

$$(\mathrm{I} + \mu \mathrm{L})\,\mathrm{x}^* = \mathrm{y}$$

- **Thm 1**: condition number κ of matrix satisfies [1]:

maximum node degree

$$\kappa \leq 1 + 2\,\mu\, d_{\max},$$

- **Observation**: By restricting search space of CNN to degree-bounded graphs, we achieve robust learning.

[1] J. Zeng et al., "**Deep Graph Laplacian Regularization**," submitted to arXiv, July 2018. (https://arxiv.org/abs/1807.11637)

Experimental Results – Numerical Comparison

- Trained on AWGN on 5 images, patches of size 26-by-26.
- Batch size is 4, model is trained for 200 epochs.
- Trained for both known and blind noise variance.

Table 1. Average PSNR (dB) and SSIM values of different methods for Gaussian noise removal. The best results for each metric is highlighted in boldface.

Noise	Metric	Method						
		BM3D	WNNM	OGLR	DnCNN-S	DnCNN-B	DeepGLR-S	DeepGLR-B
25	PSNR	29.95	30.28	29.78	**30.41**	30.33	30.26	30.21
	SSIM	0.8496	0.8554	0.8463	**0.8609**	0.8594	0.8599	0.8557
40	PSNR	27.62	28.08	27.68	28.10	28.13	**28.16**	28.04
	SSIM	0.7920	0.8018	0.7949	0.8080	0.8091	**0.8125**	0.8063
50	PSNR	26.69	27.08	26.58	27.15	27.18	**27.25**	27.12
	SSIM	0.7651	0.7769	0.7539	0.7809	0.7811	**0.7852**	0.7807

[1] Kai Zhang et al, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *TIP* 2017.
[2] Marc Lebrun et al, "The noise clinic: a blind image denoising algorithm," *IPOL* 2015.

Experimental Results – Numerical Comparison

- DeepGLR has average PSNR of 0.34 dB higher than CDnCNN [1].
- Model-based provides robustness against overfitting.

Table 2. Evaluation of different methods for low-light image denoising. The best results for each metric, except for those tested on the training set, are highlighted in boldface.

Metric	Noisy	Method					
		CBM3D	MC-WNNM	CDnCNN (train)	CDnCNN	CDeepGLR (train)	CDeepGLR
PSNR	20.36	26.08	26.23	33.43	31.26	32.31	**31.60**
SSIM Y	0.5198	0.8698	0.8531	0.9138	0.8978	0.9013	**0.9028**
SSIM R	0.2270	0.6293	0.5746	0.8538	0.8218	0.8372	**0.8297**
SSIM G	0.4073	0.8252	0.7566	0.8979	0.8828	0.8840	**0.8854**
SSIM B	0.1823	0.5633	0.5570	0.8294	0.7812	0.8138	**0.7997**

[1] Kai Zhang et al, “Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising,” *TIP* 2017.
[2] Marc Lebrun et al, “The noise clinic: a blind image denoising algorithm,” *IPOL* 2015.

Experimental Results – Visual Comparison

- trained on Gaussian noise, tested on low-light images in (RENOIR).
- Competing methods: DnCNN [1], noise clinic [2].
- outperformed DnCNN by 5.52 dB, and noise clinic by 1.87 dB.

DnCNN

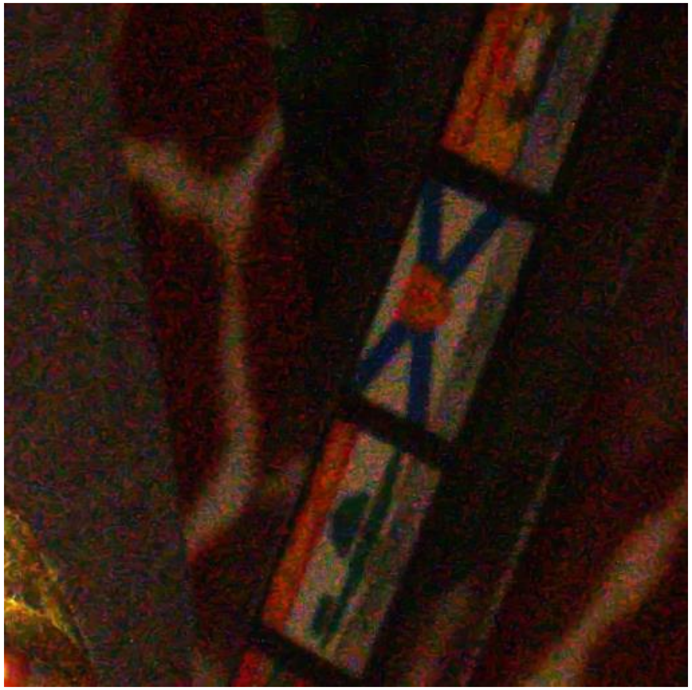
clinic

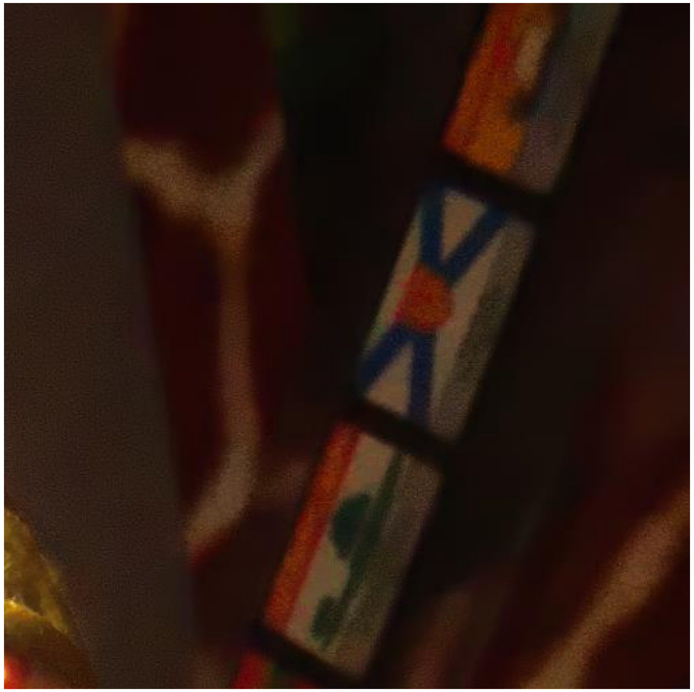
DeepGLR

[1] Kai Zhang et al, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *TIP* 2017.
[2] Marc Lebrun et al, "The noise clinic: a blind image denoising algorithm," *IPOL* 2015.

Experimental Results – Visual Comparison

- trained on Gaussian noise, tested on low-light images in (RENOIR).
- Competing methods: DnCNN [1], noise clinic [2].
- outperformed DnCNN by 5.52 dB, and noise clinic by 1.87 dB.


DnCNN

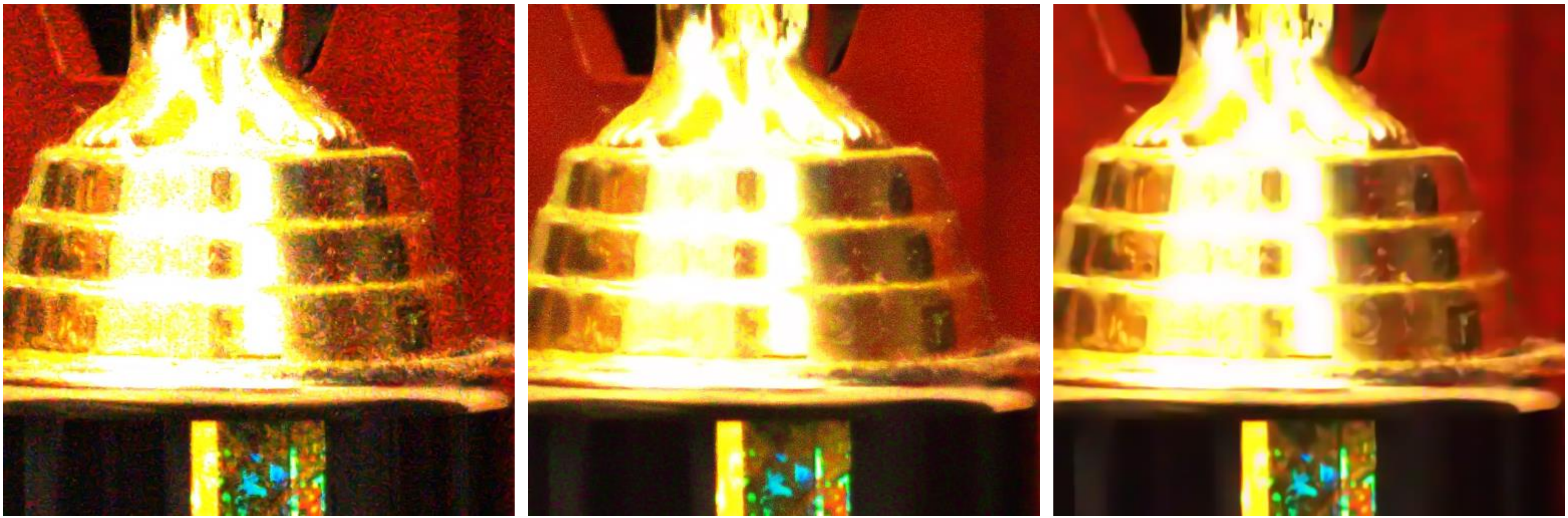

clinic


DeepGLR

[1] Kai Zhang et al, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *TIP* 2017.
[2] Marc Lebrun et al, "The noise clinic: a blind image denoising algorithm," *IPOL* 2015.

Experimental Results – Visual Comparison

- trained on Gaussian noise, tested on low-light images in (RENOIR).
- Competing methods: DnCNN [1], noise clinic [2].
- outperformed DnCNN by 5.52 dB, and noise clinic by 1.87 dB.



DnCNN clinic DeepGLR

[1] Kai Zhang et al, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *TIP* 2017.
[2] Marc Lebrun et al, "The noise clinic: a blind image denoising algorithm," *IPOL* 2015.

Outline

- GSP Fundamentals
- GSP for Image Compression
  - Optimality of GFT
- GSP for Inverse Imaging
  - Graph Laplacian Regularizer
  - Reweighted Graph TV: 3D Point Cloud Denoising
- Deep GLR
- Ongoing & Future Work

Summary

- Frequencies for Graphs
  - Optimality of GFT for signal decorrelation
  - Compression for PWS images, point cloud, light field images
- GSP for Inverse Imaging
  - PWS-promoting Graph Laplacian Regularizer, RGTV
  - Image / point cloud denoising, deblurring, contrast enhancement
- Hybrid graph-based / data-driven approach
  - Robustness against CNN overfitting

Ongoing & Future Work

- Graph learning given small data
  - Bayesian networks, DAG
- Unrolling of graph-based convex optimization
  - Unrolling of ADMM, proximal gradient with GTV prior, convex set constraints.
- Graph sampling
  - Fast sampling w/o eigen-decomposition, matrix inverse [1]
  - Reconstruction-cognizant sampling [2]

[1] F. Wang et al, "Fast Sampling of Graph Signals with Noise via Neumann Series Conversion," submitted to *ICASSP*, May 2019.
[2] Y. Bai et al., "Reconstruction-Cognizant Graph Sampling Using Gershgorin Disc Alignment," submitted to *ICASSP*, May 2019.

Q&A

- Email: genec@yorku.ca
- Homepage: https://www.eecs.yorku.ca/~genec/index.html