Gene Cheung

Associate Professor, York University

19th November, 2018

Graph Spectral Image Compression & Restoration

(an intuitive & fun introduction)

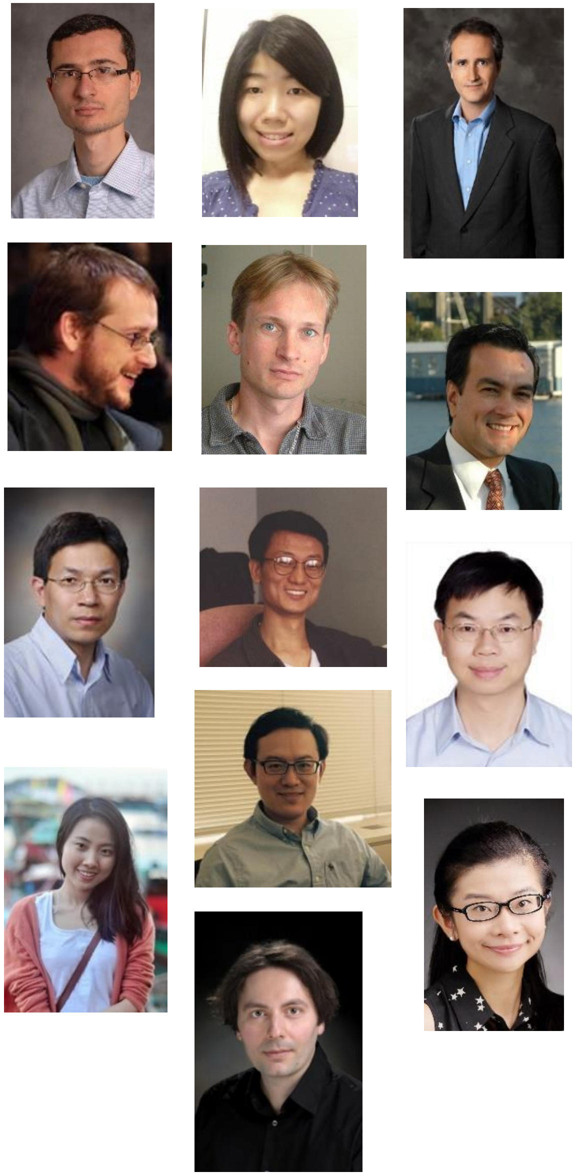
.

[1] Gene Cheung, Enrico Magli, Yuichi Tanaka, Michael Ng, "**Graph Spectral Image Processing**," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 907-930, May 2018..

Acknowledgement

Collaborators:

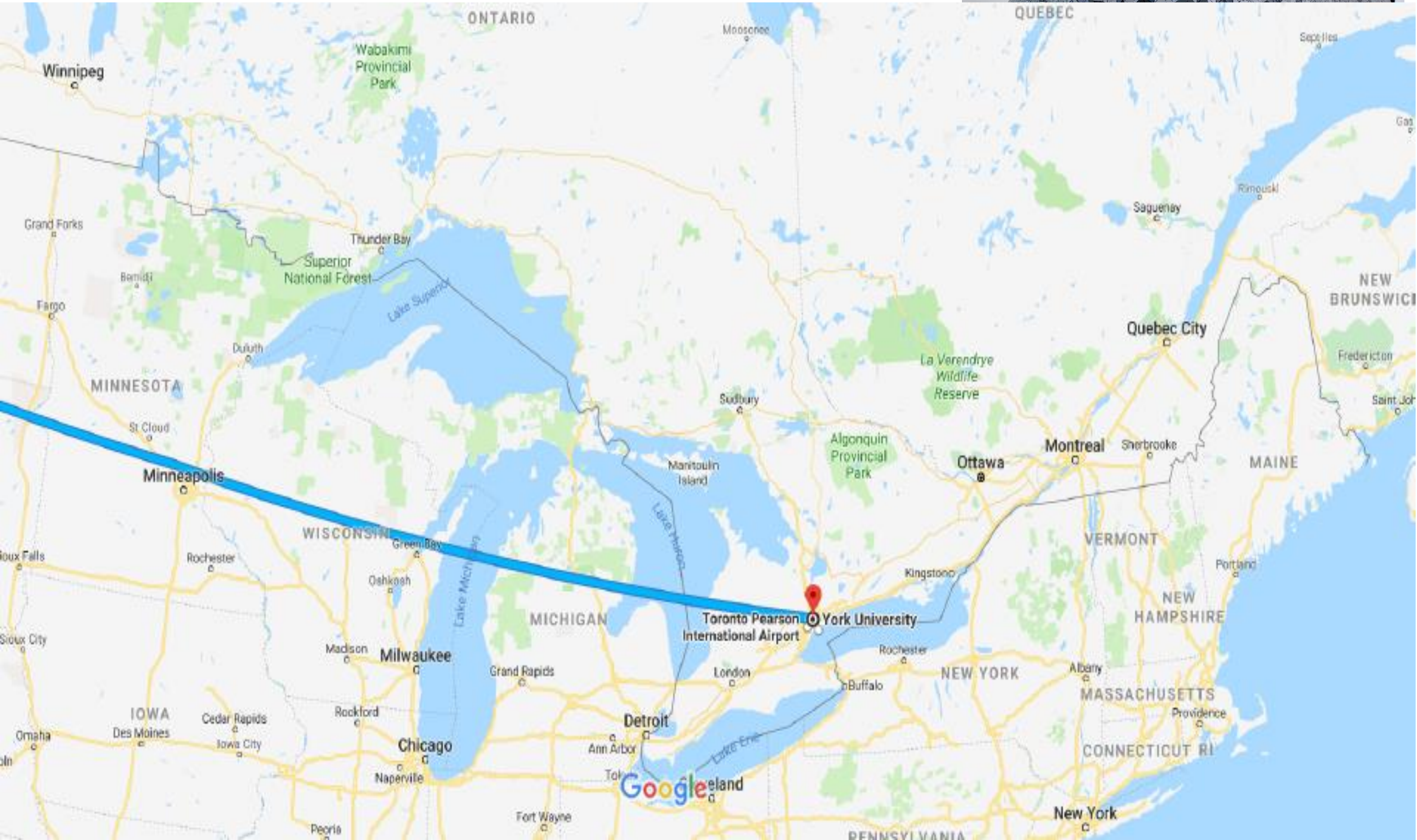
- Y. Nakatsukasa (NII, Japan)
- S. Muramatsu (Niigata, Japan)
- **A. Ortega** (USC, USA)
- **D. Florencio** (MSR, USA)
- **P. Frossard** (EPFL, Switzerland)
- J. Liang, I. Bajic (SFU, Canada)
- X. Wu (McMaster U, Canada)
- V. Stankovic (U of Strathclyde, UK)
- **P. Le Callet** (U of Nantes, France)
- **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)



York University

- Founded in 1959 in Toronto, Ontario (largest city in Canada).
- 3rd largest public university in Canada:
    - 50,000 undergrads, 6,000 grads.
    - 7,000 faculties.
- Lassonde School of Engineering:
    - 4 departments, 130 professors, 3300 students
- Department of EECS:
    - **Computer vision**, machine learning, communications, power electronics.
- https://www.youtube.com/watch?v=W54mk0YAS_0

York University



York University

- Founded in 1959 in Toronto, Ontario (largest city in Canada).
- 3rd largest public university in Canada:
  - 50,000 undergrads, 6,000 grads.
  - 7,000 faculties.
- Lassonde School of Engineering:
  - 4 departments, 130 professors, 3300 students
- Department of EECS:
  - **Computer vision**, machine learning, communications, power electronics.
- https://www.youtube.com/watch?v=W54mk0YAS_0

Outline

- GSP Fundamentals
- GSP for Image Compression
    - Graph Fourier Transform
- GSP for Inverse Imaging
    - Graph Laplacian Regularizer (GLR)
    - Reweighted Graph TV
- Summary

Outline

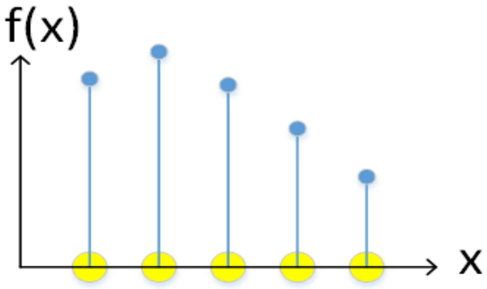
- GSP Fundamentals

- GSP for Image Compression
  - Graph Fourier Transform

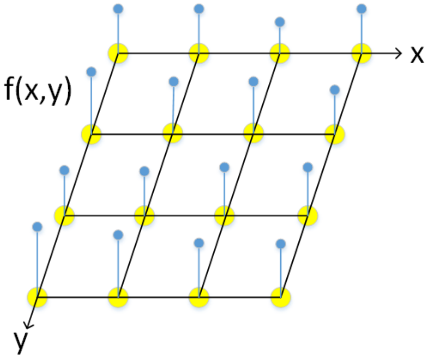
- GSP for Inverse Imaging
  - Graph Laplacian Regularizer (GLR)
  - Reweighted Graph TV

- Summary

Digital Signal Processing

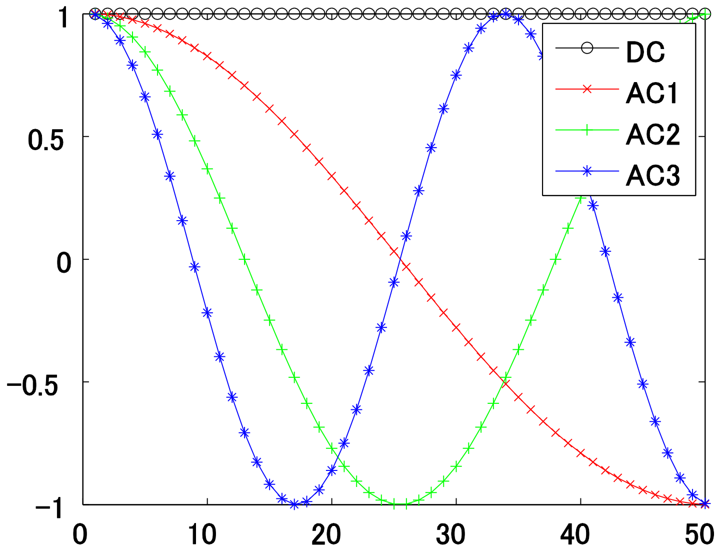


- Discrete signals on ***regular*** data kernels.
  - Ex.1: audio on regularly sampled timeline.
  - Ex.2: image on 2D grid.



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

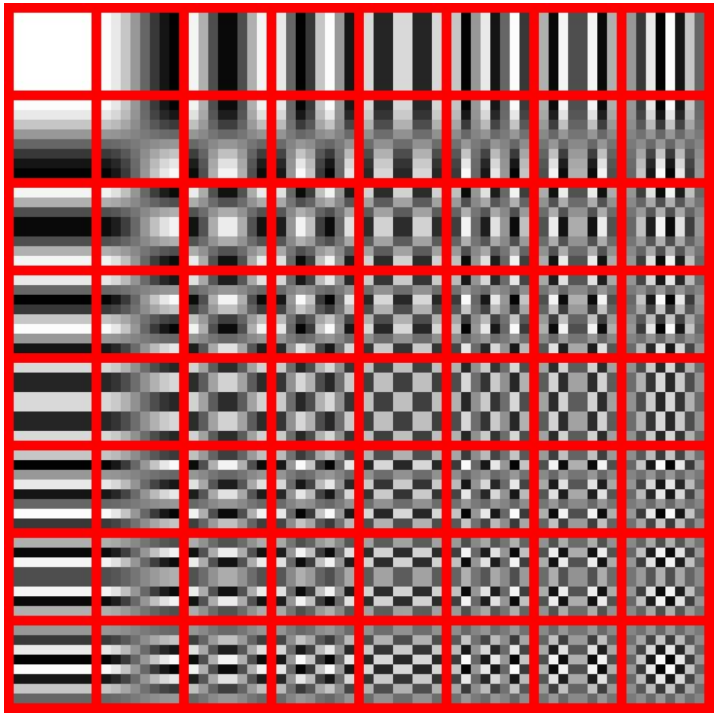
Smoothness of Signals



- 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



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

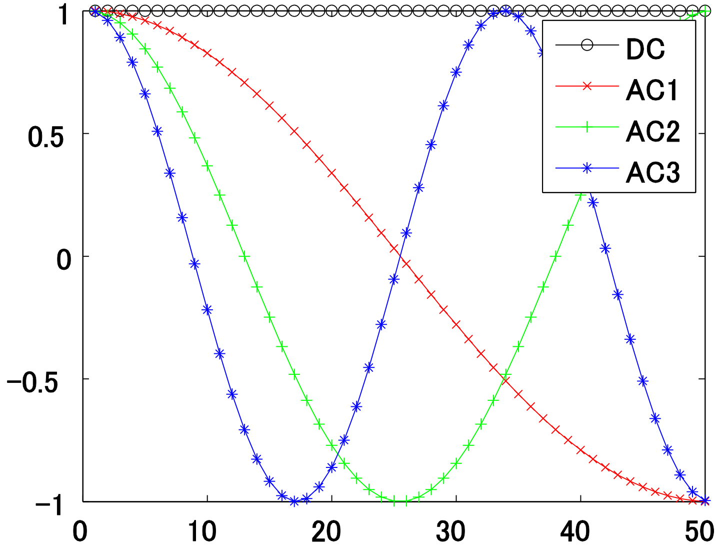
desired signal

transform

transform coeff.

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

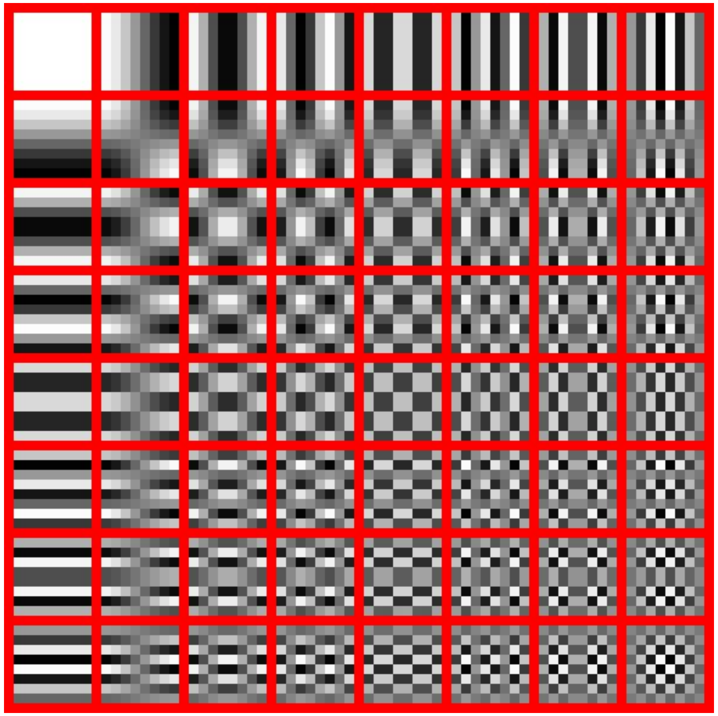
Smoothness of Signals



- 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



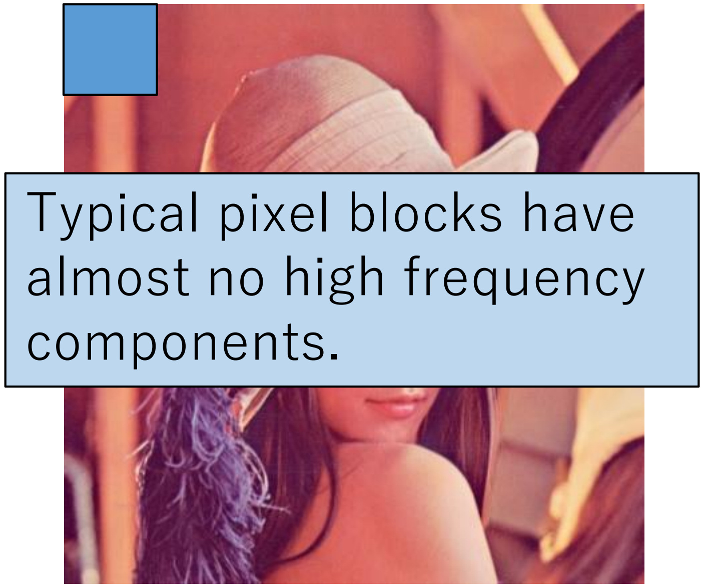
$$\mathrm{a} = \Phi\,\mathrm{x}$$

desired signal

transform

transform coeff.

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

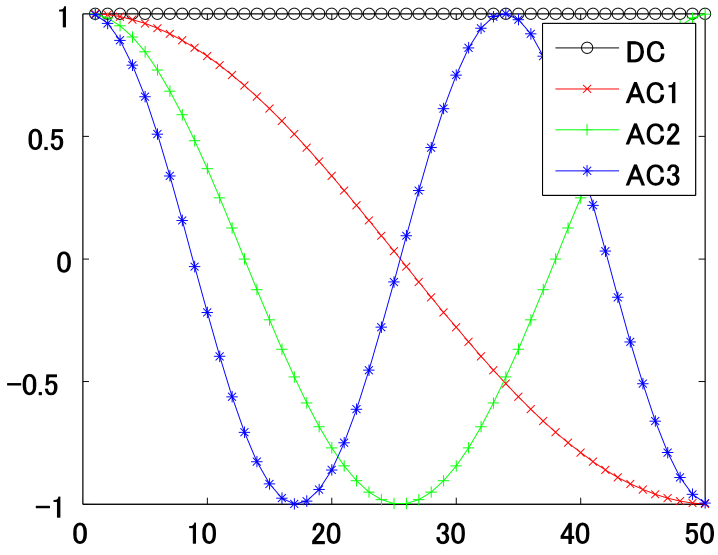


Typical pixel blocks have almost no high frequency components.

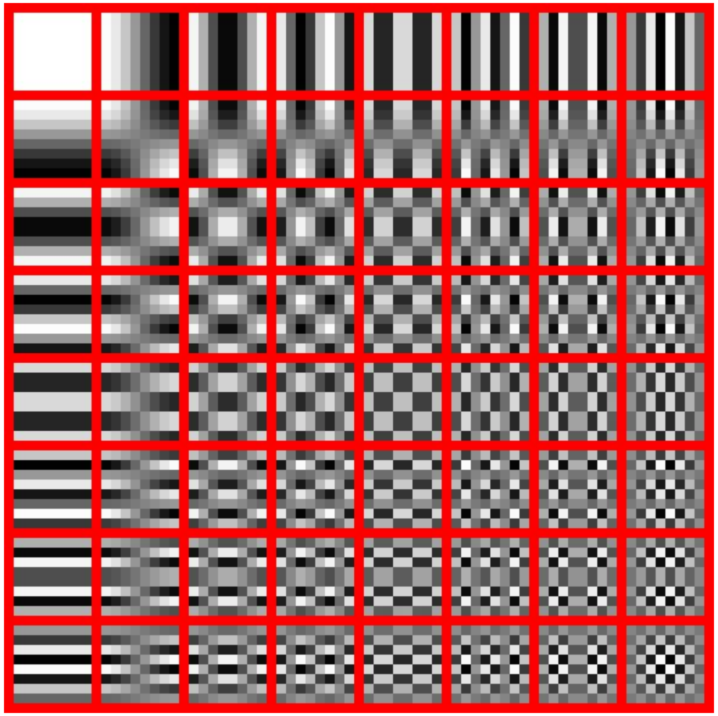
Smoothness of Signals

- 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



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

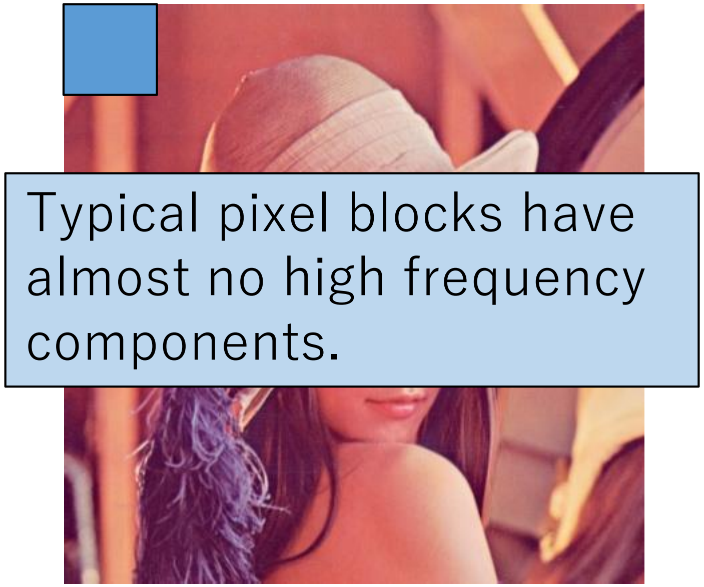
desired signal

transform

transform coeff.

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

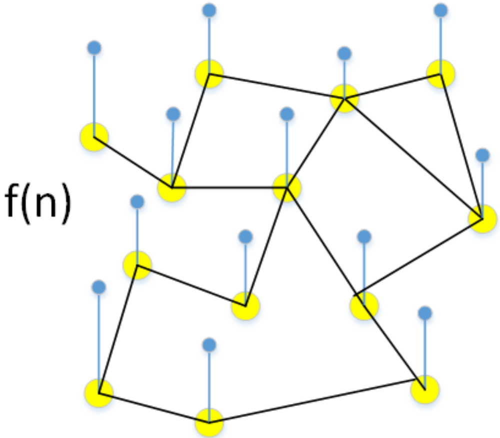
Compact signal representation



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

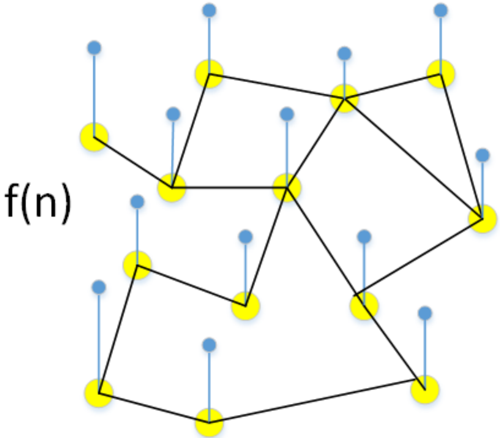
1. Data domain is naturally a graph.
   - Ex: ages of users on social networks.

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*.



example graph-signal

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

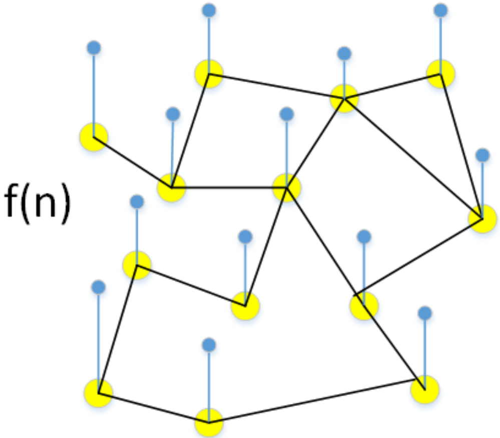
2. Underlying data structure unknown.
   - Ex: images: 2D grid → structured graph.

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

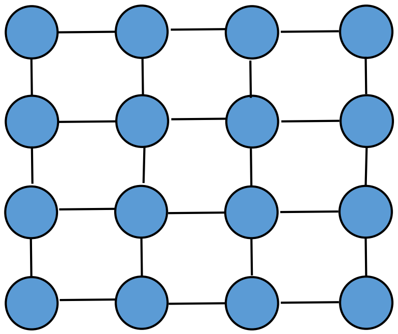
- Signals on ***irregular*** data kernels described by graphs.
  - Graph: nodes and edges.
  - Edges reveals *node-to-node relationships*.



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.

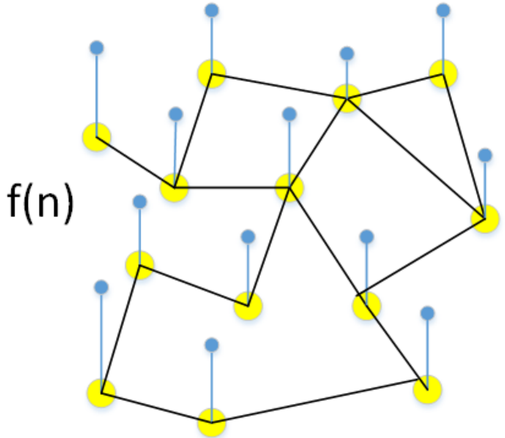


Graph Signal Processing (GSP) addresses the problem of processing signals that live on graphs.

[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

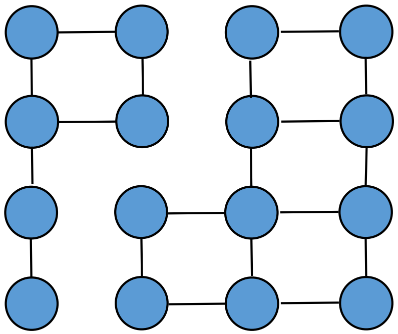
- Signals on ***irregular*** data kernels described by graphs.
  - Graph: nodes and edges.
  - Edges reveals *node-to-node relationships*.



example graph-signal

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

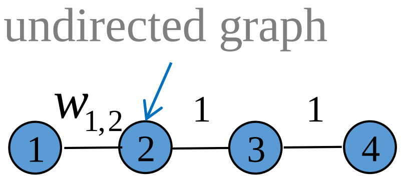
2. Underlying data structure unknown.
   - Ex: images: 2D grid → structured graph.



> 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 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*.

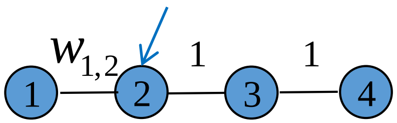
$$L = \begin{bmatrix} w_{1,2} & -w_{1,2} & 0 & 0 \\ -w_{1,2} & w_{1,2}+1 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

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

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

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

Graph Fourier Transform (GFT)



Graph Laplacian:

- *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 Spectrum from GFT

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

eigenvalues along diagonal

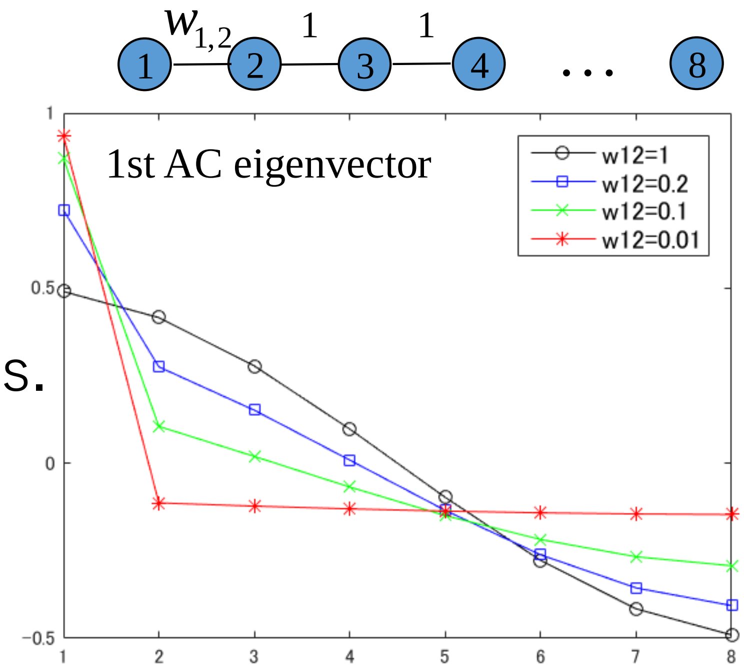
$$L = V \Sigma \enclose{circle}{V^T}$$

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

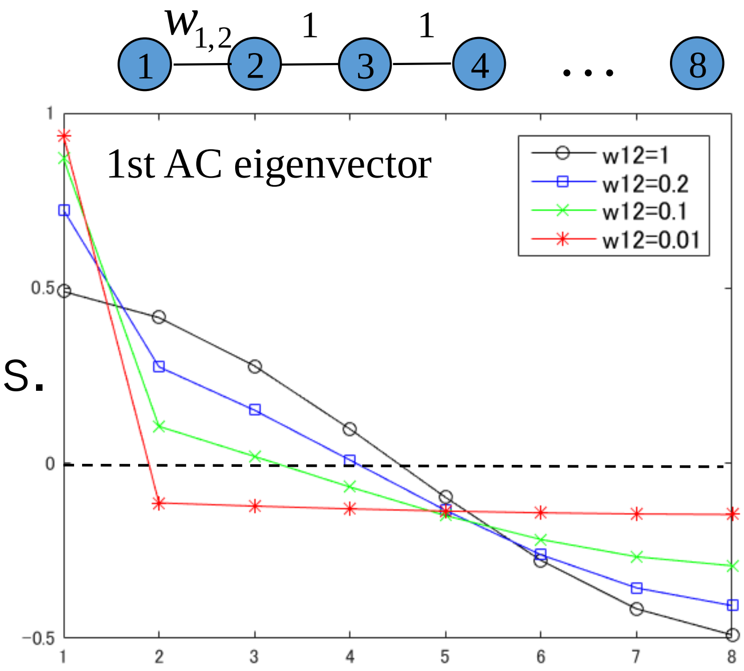
$$\mathrm{L} = \mathrm{V}\Sigma\boxed{\mathrm{V}^T}$$

GFT

eigenvectors in columns

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

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.

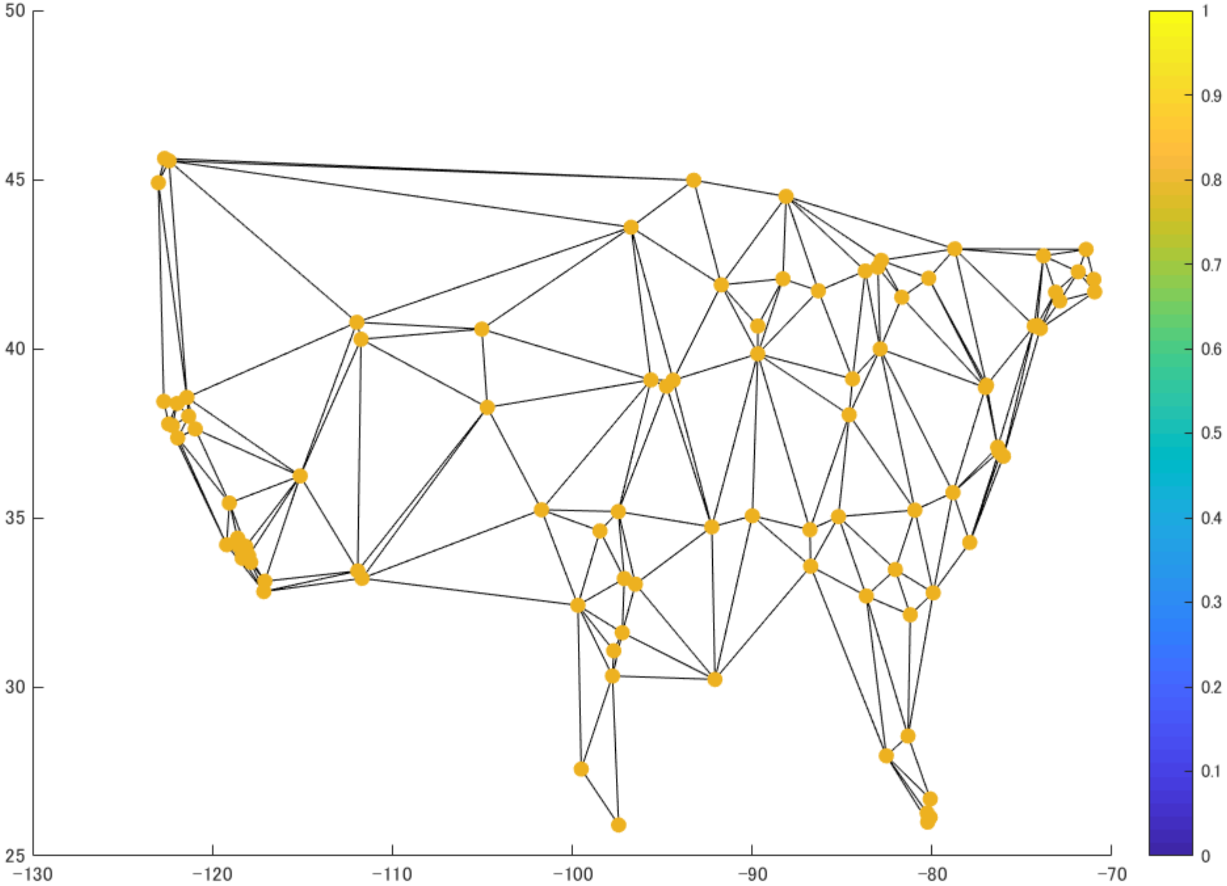
Graph Frequency Examples (US Temperature)

- 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

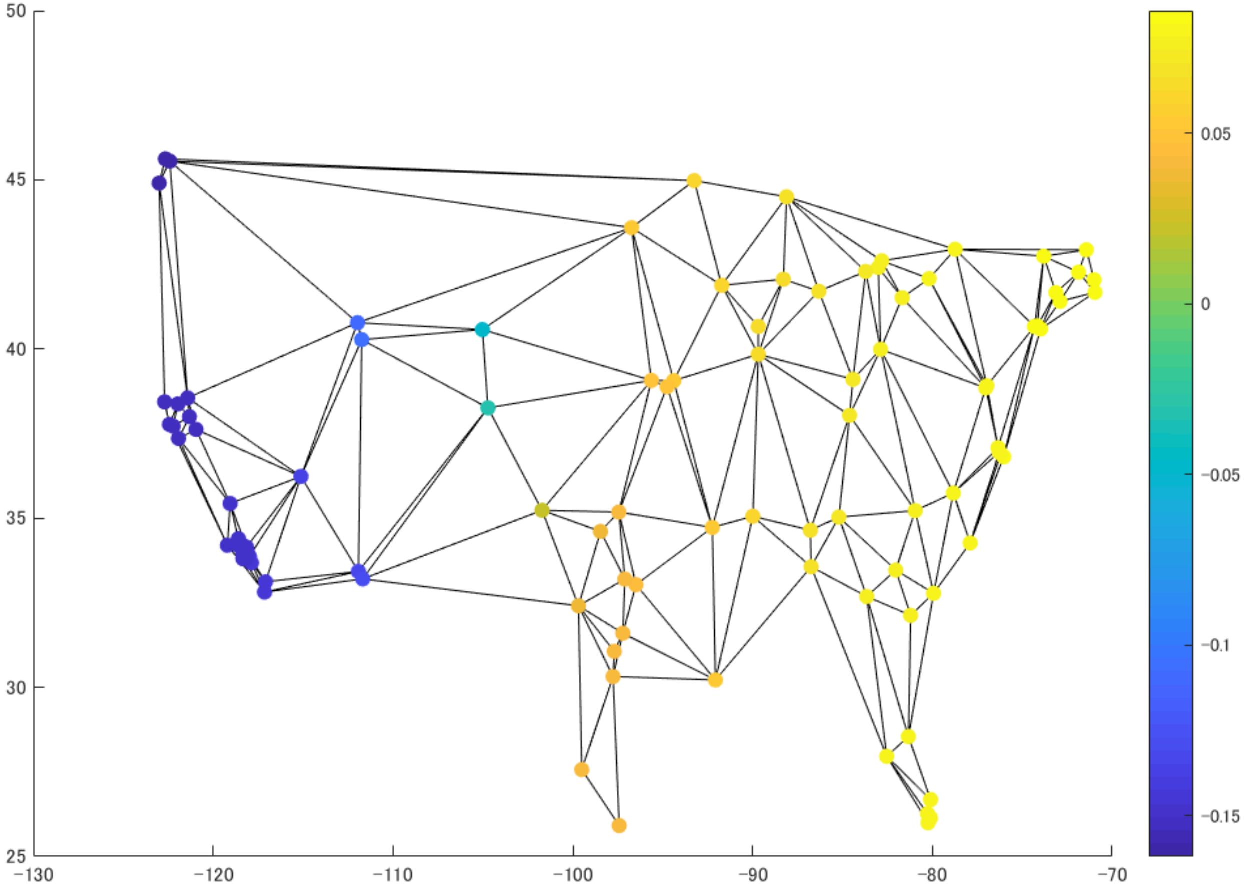
Graph Frequency Examples (US Temperature)

- 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

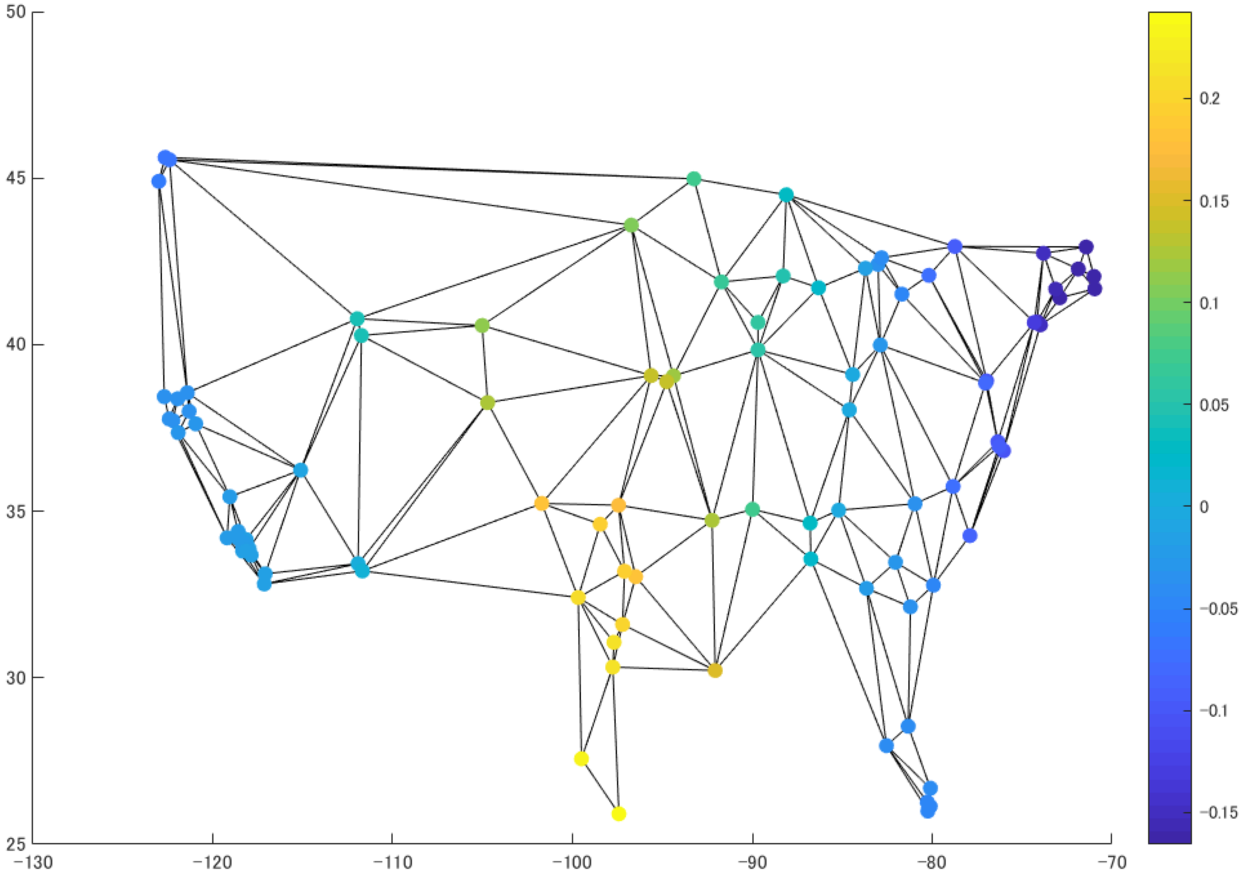
Graph Frequency Examples (US Temperature)

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

location diff.

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

Edge weights



V3: 2nd AC component

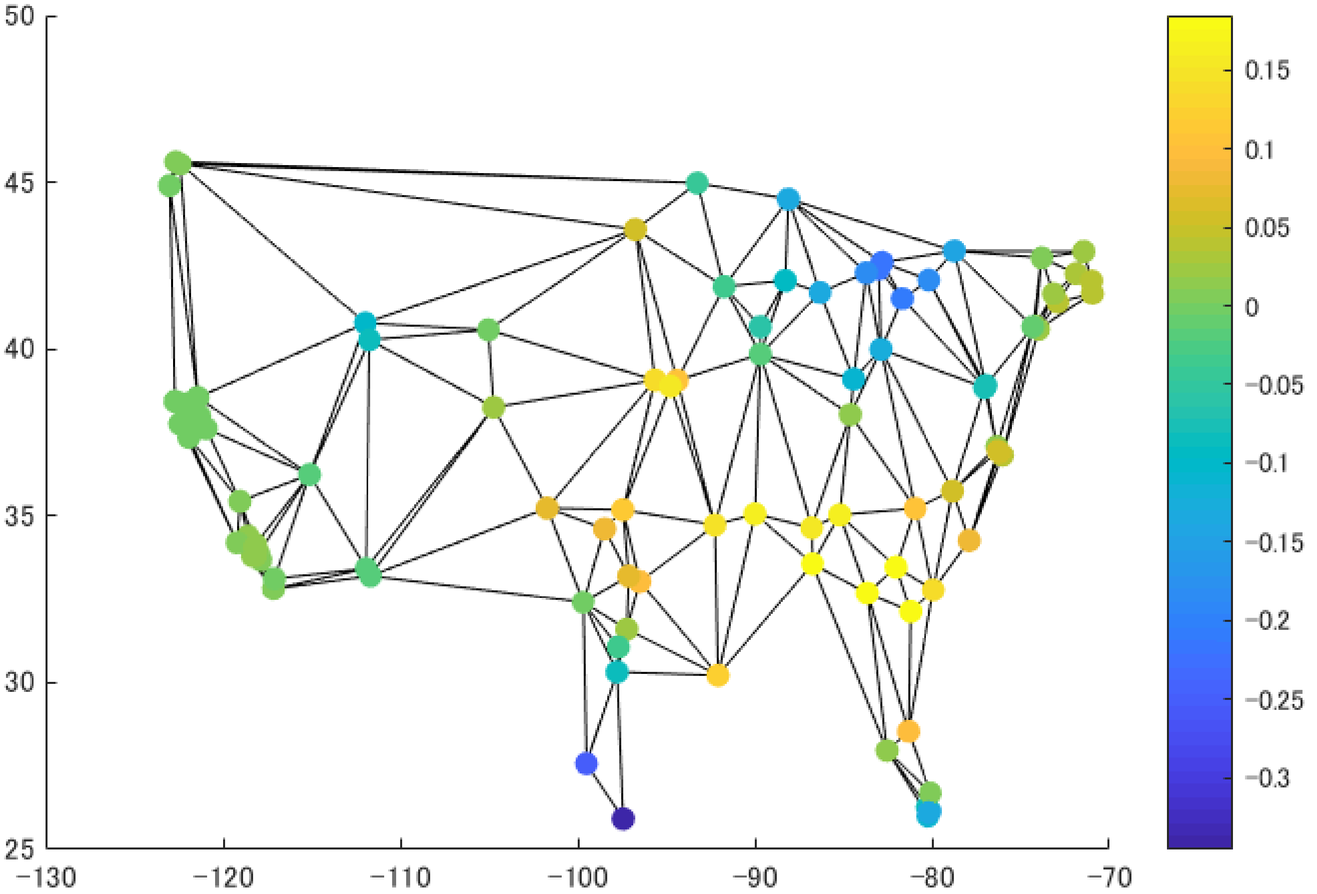
Graph Frequency Examples (US Temperature)

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

location diff.

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

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]:

$$L_g = L + D^*$$

Characteristics:

- 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.

Outline

- GSP Fundamentals

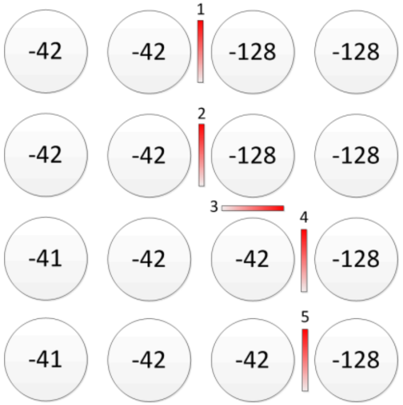
- GSP for Image Compression
  - Graph Fourier Transform

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

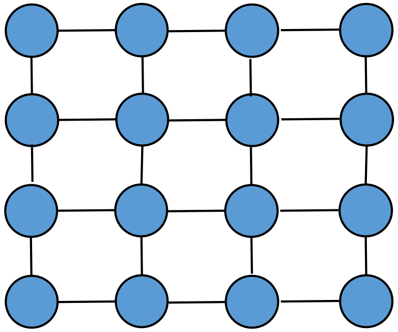
- Summary

GFT for Image Compression

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



-42	-42	-128	-128
-42	-42	-128	-128
-41	-42	-42	-128
-41	-42	-42	-128



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

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

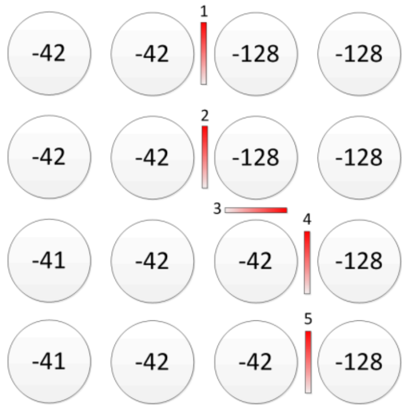
4. Transmit bits (*contour*) to identify chosen GFT to decoder (overhead of GFT).

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

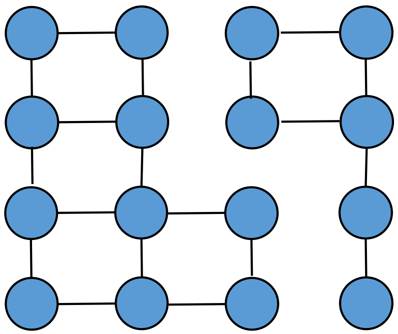
[2] W. Hu, G. Cheung, X. Li, O. Au, "**Depth Map Compression using Multi-resolution Graph-based Transform for Depth-image-based Rendering**," *IEEE International Conference on Image Processing*, Orlando, FL, September 2012.

GFT for Image Compression

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



-42	-42	-128	-128
-42	-42	-128	-128
-41	-42	-42	-128
-41	-42	-42	-128



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

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

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.

Edge Weight Assignment

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

0-mean r.v. with large var. σ²

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

0-mean r.v. with var. σ_k^2

$$\mathbf{F}\mathbf{x} = \mathbf{b},$$

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad \mathrm{b} = \begin{bmatrix} \eta \\ e_2 \\ \vdots \\ \\ \\ e_N \end{bmatrix}$$

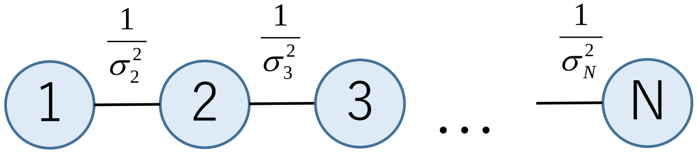
- Precision (inverse covariance) matrix is tridiagonal.

large

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

Graph Laplacian matrix!

$$\approx \mathbf{L}$$



Edge Weight Assignment

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

0-mean r.v. with large var. σ²

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

0-mean r.v. with var. σ_k^2

$$\mathbf{F}\mathbf{x} = \mathbf{b},$$

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad \mathrm{b} = \begin{bmatrix} \eta \\ e_2 \\ \vdots \\ \\ e_N \end{bmatrix}$$

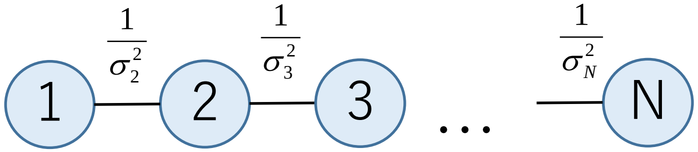
- Precision (inverse covariance) matrix is tridiagonal.

large

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

1. Eigenvectors of covariance matrix compose KLT, optimally decorrelating signal.

$\approx \mathbf{L}$



Edge Weight Assignment

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

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

0-mean r.v. with large var. σ^2

0-mean r.v. with var. σ$_k^2$

$$\mathbf{F}\mathbf{x} = \mathbf{b},$$

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad \mathrm{b} = \begin{bmatrix} \eta \\ e_2 \\ \vdots \\ \\ \\ e_N \end{bmatrix}$$

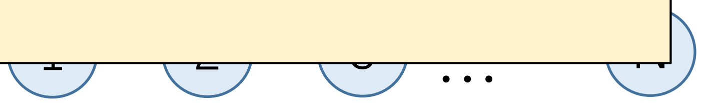
- Precision (inverse covariance) matrix is tridiagonal.

large

$$\mathrm{Q} = \begin{bmatrix} \frac{1}{\ } + \frac{1}{\ } & -\frac{1}{\ } & 0 & \cdots & 0 \\ \\ 0 & \cdots & 0 & -\frac{1}{\sigma_N^2} & \frac{1}{\sigma_N^2} \end{bmatrix}$$

1. Eigenvectors of covariance matrix compose KLT, optimally decorrelating signal.

2. Graph Laplacian matrix approximates precision matrix, hence GFT approximates KLT.



Multi-resolution-GFT Implementation

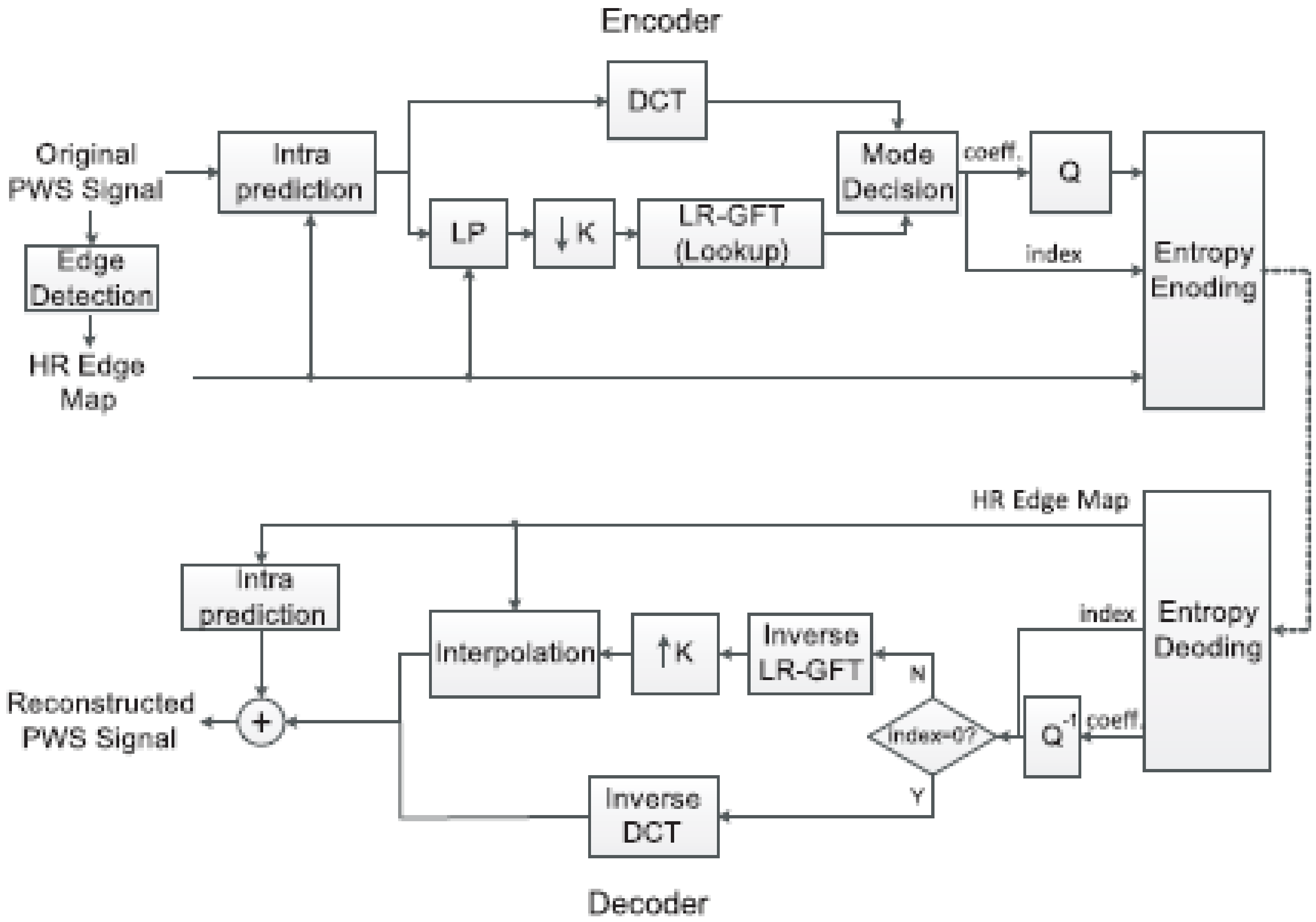
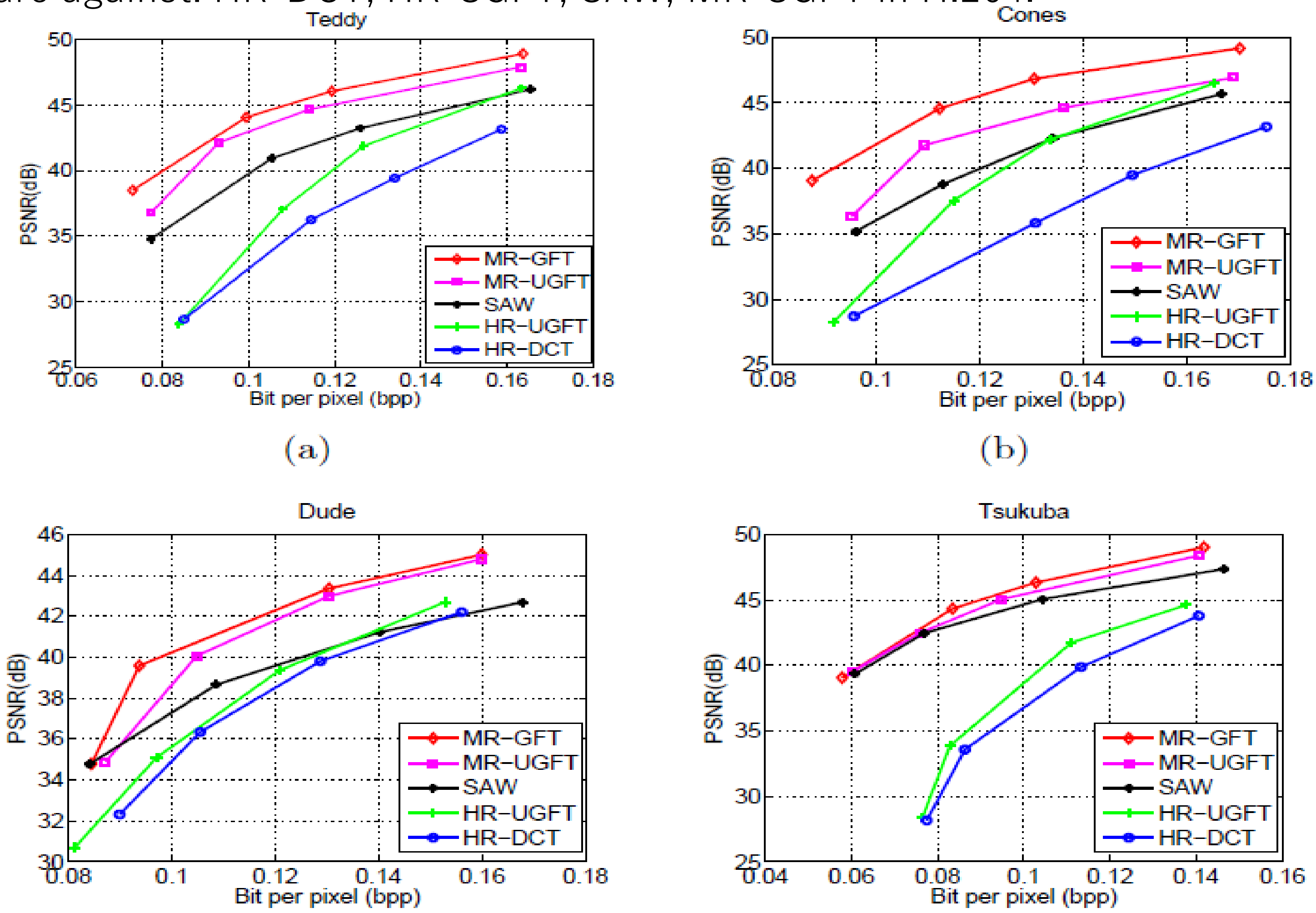


Fig. 2. MR-GFT coding system for PWS images.

[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)

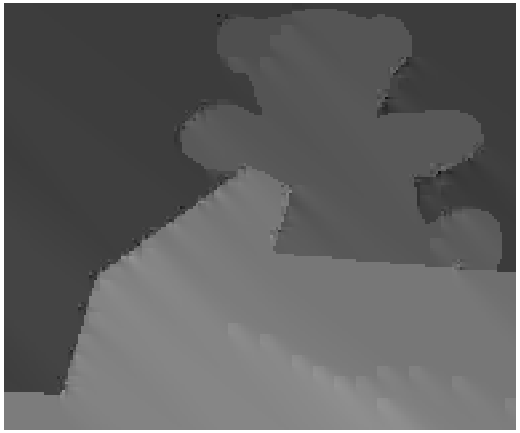
(b)

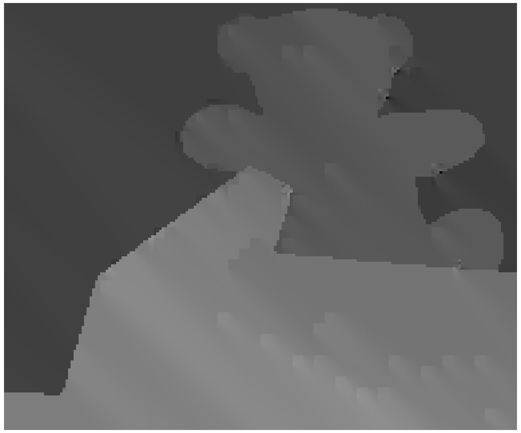
(c)

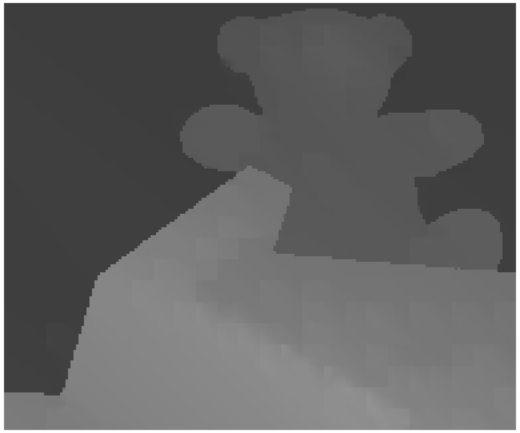
(d)

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

Subjective Results



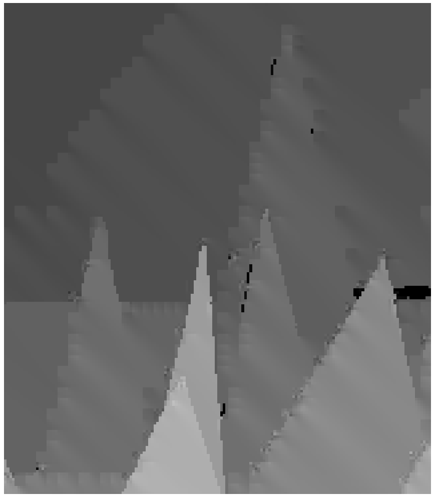


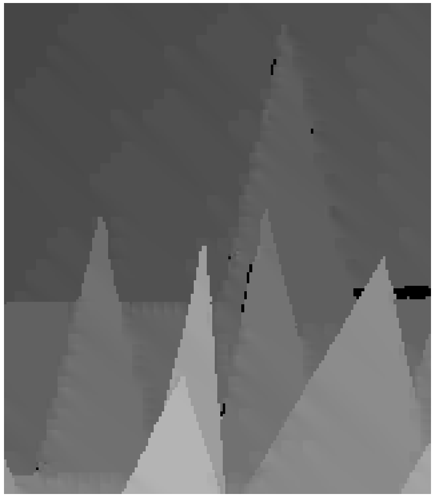


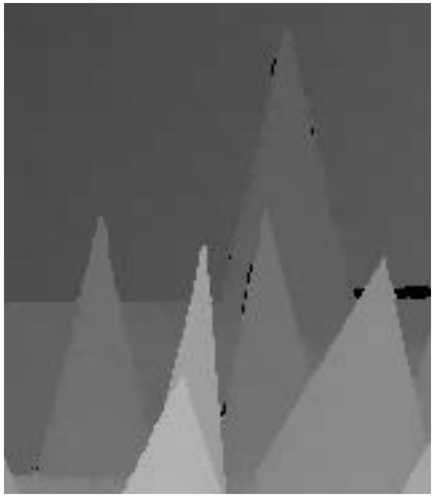
HR-DCT

HR-SGFT

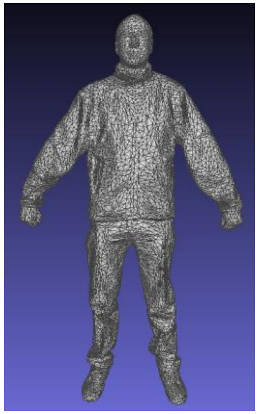
MR-GFT





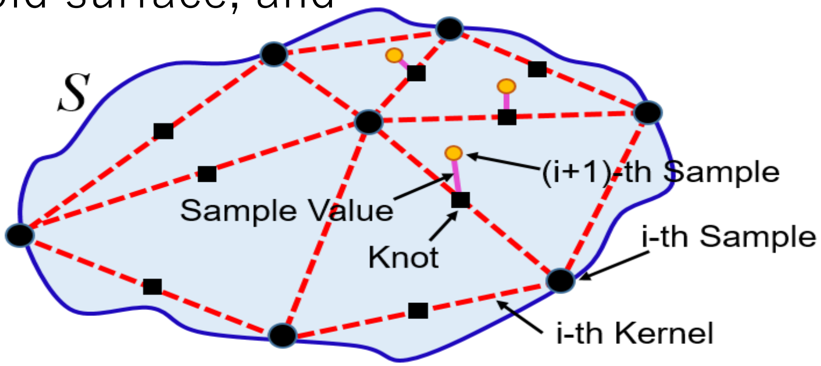


Graph-Signal Sampling / Encoding for 3D Point Cloud



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.

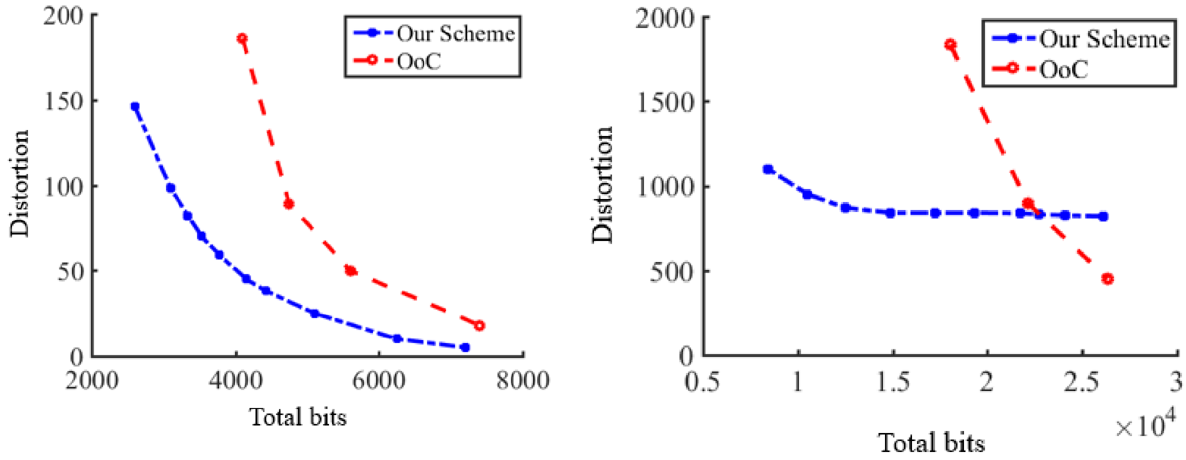


- **Example**:

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

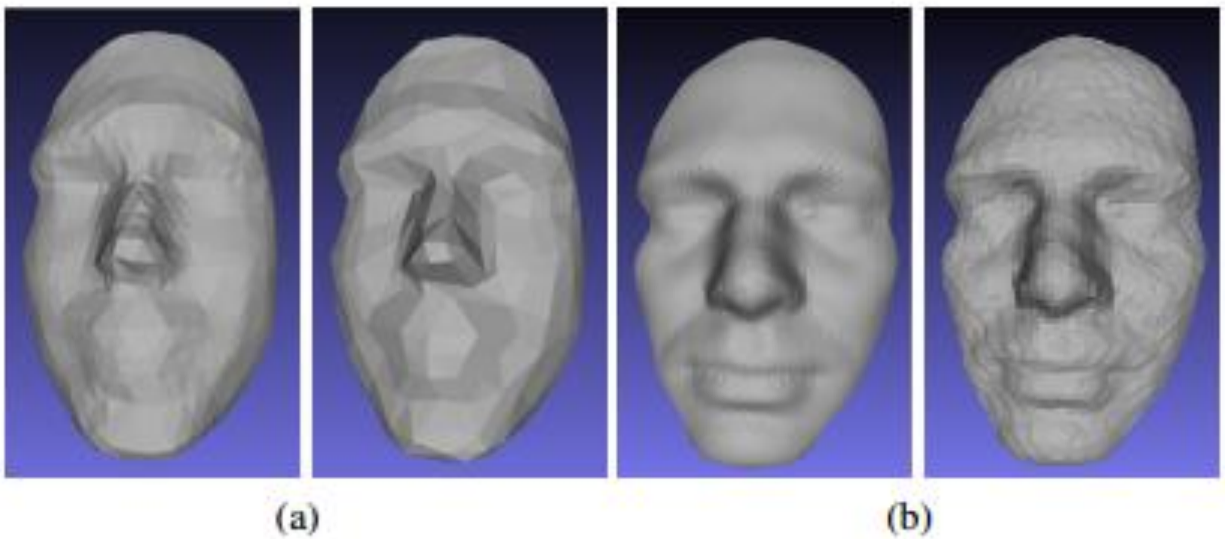
Graph-Signal Sampling / Encoding for 3D Point Cloud

- **Experimental Results**:



(a) Dataset1

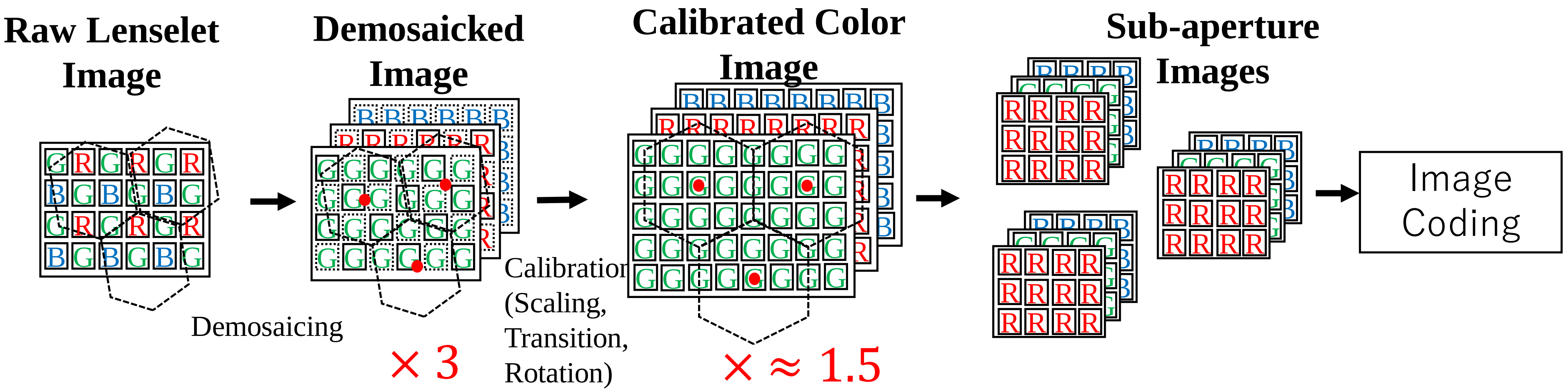
(b) Dataset2



[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.



Raw Lenselet Image

Demosaicked Image

Calibrated Color Image

Sub-aperture Images

Image Coding

Demosaicing

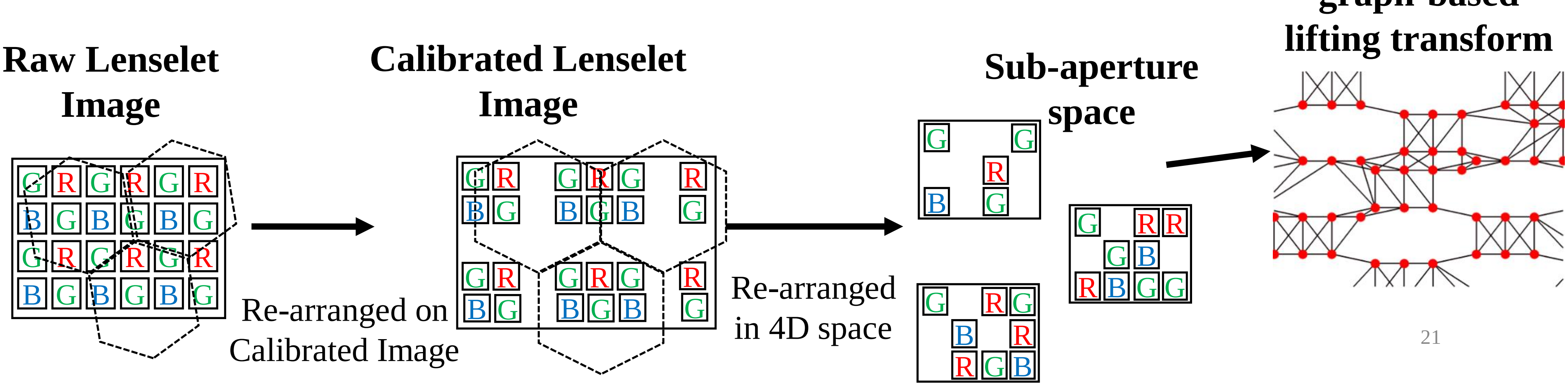
× 3

Calibration (Scaling, Transition, Rotation)

× ≈ 1.5

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

graph-based lifting transform



Raw Lenselet Image

Calibrated Lenselet Image

Sub-aperture space

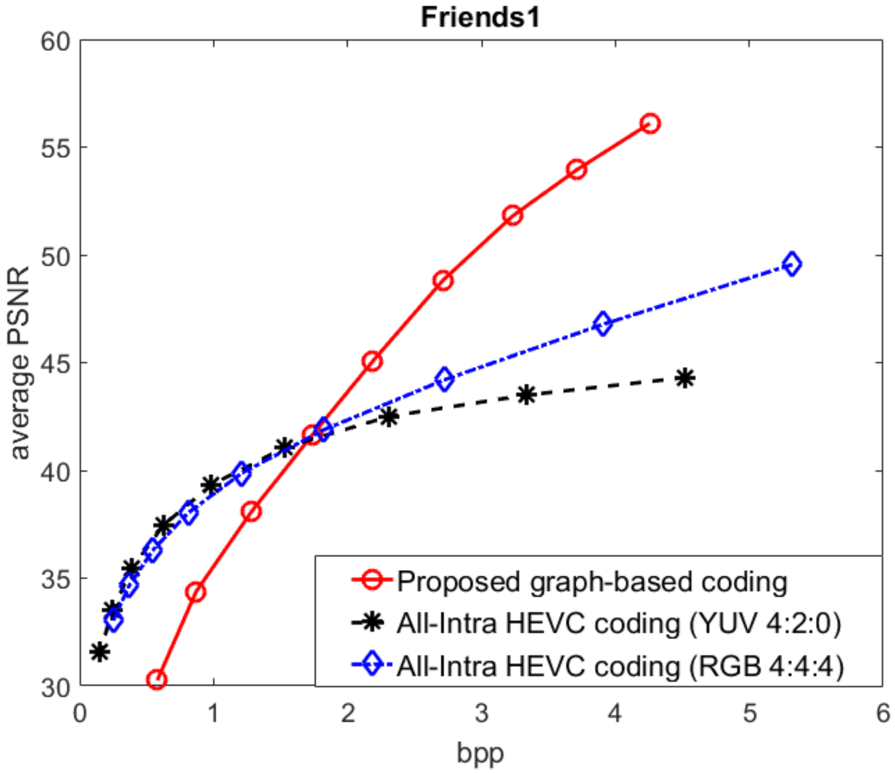
Re-arranged on Calibrated Image

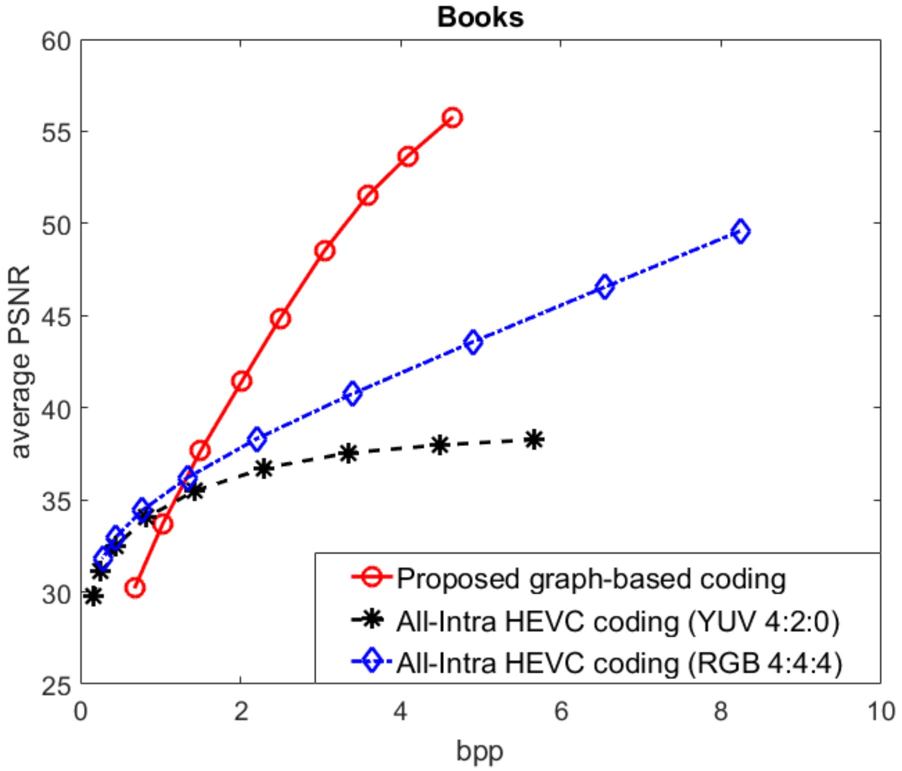
Re-arranged in 4D space

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
    - Graph Fourier Transform

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

- Summary

Graph Laplacian Regularizer

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

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

signal smooth in
nodal domain

signal contains
mostly low graph freq.

- **Signal Denoising**:

observation

desired signal

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

noise

- **MAP Formulation**:

fidelity term

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

smoothness prior

update edge
weights

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

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**:

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

- **MAP Formulation**:

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

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

update edge weights

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

pixel intensity diff.

pixel location diff.

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

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.

Graph Laplacian Regularizer

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

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

signal smooth in nodal domain

signal contains mostly low graph freq.

- **Signal Denoising**:

- **MAP Formulation**:

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

pixel intensity diff.

pixel location diff.

upd

wei

ghts

trix

1. Reweighted Graph Laplacian Regularizer (RGLR):

$$\mathrm{x}^T \mathrm{L}(\mathrm{x})\mathrm{x} = \frac{1}{2}\sum_{i,j} \exp\left[\frac{\left(x_i - x_j\right)^2}{\sigma^2}\right]\left(x_i - x_j\right)^2$$

[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:

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

  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

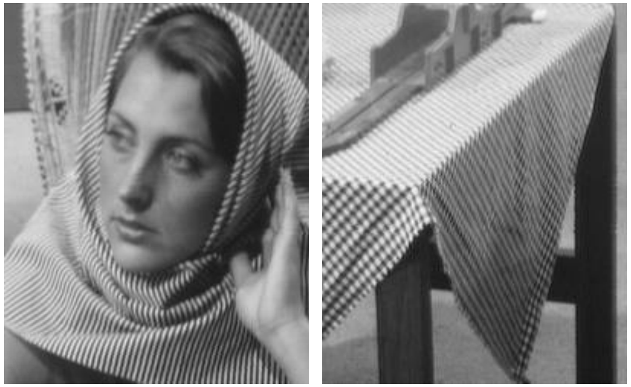
$$\mathbf{f}_3^D = \frac{1}{K + \sigma_e^2 / \sigma_g^2} \sum_{k=0}^{K-1} \mathbf{p}_k$$

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

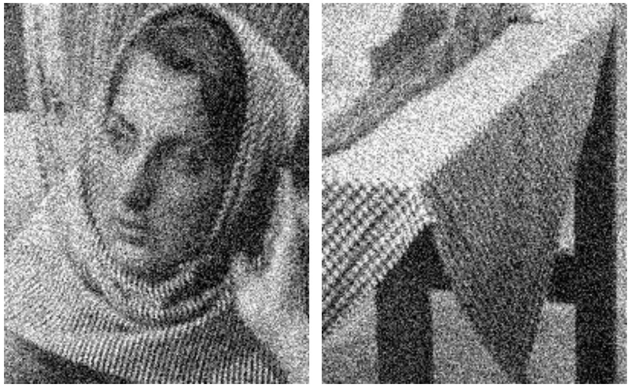
[1] J. Pang, G. Cheung, "**Graph Laplacian Regularization for Inverse Imaging: Analysis in the Continuous Domain**," *IEEE Transactions on Image Processing*, vol. 26, no.4, pp.1770-1785, April 2017.

Denoising Experiments (natural images)

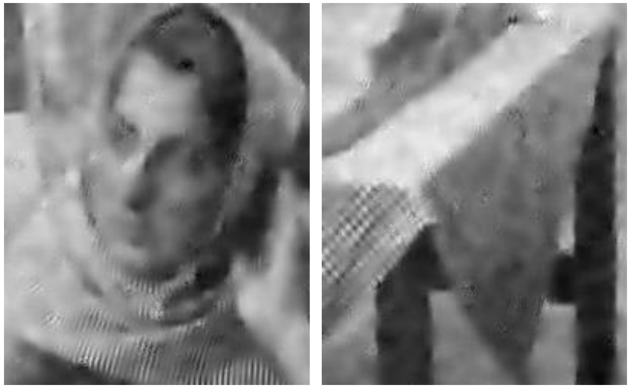
- Subjective comparisons ($\sigma_I = 40$)



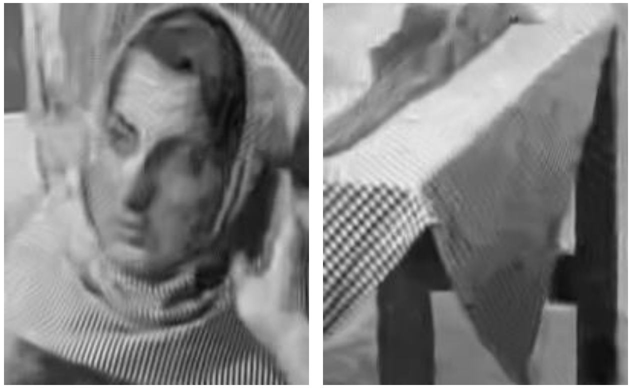
Original



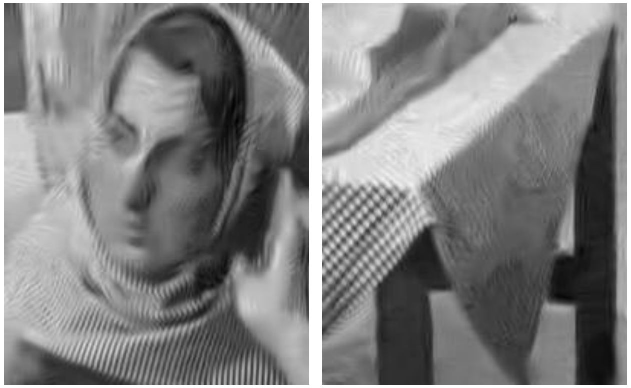
Noisy, 16.48 dB



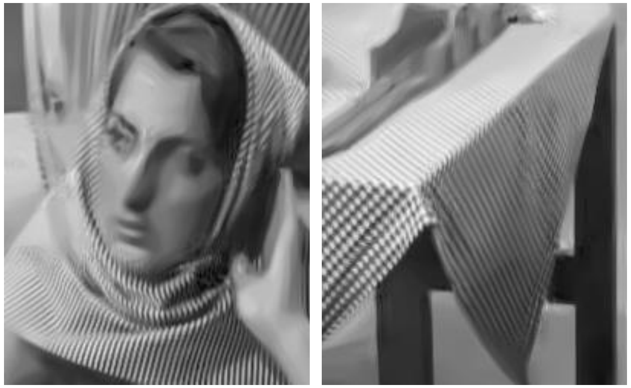
K-SVD, 26.84 dB



BM3D, 27.99 dB



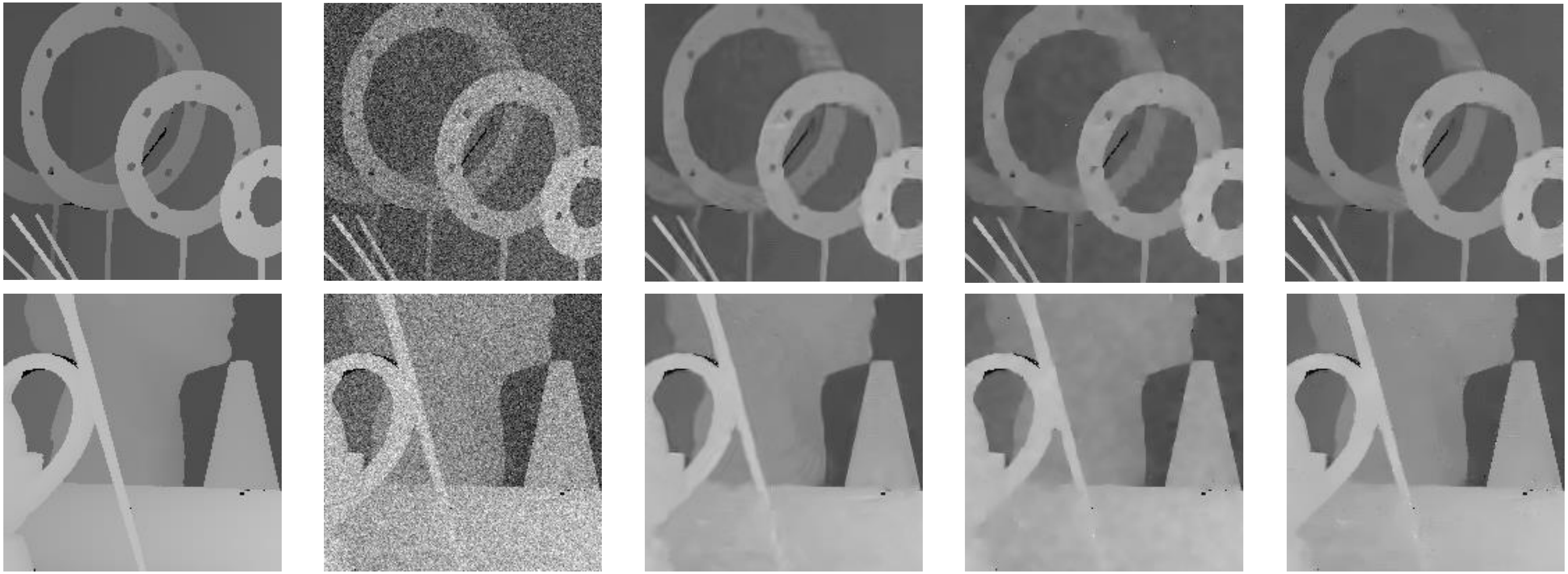
PLOW, 28.11 dB



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:

$$\mathbf{y} = \underset{\text{scalar}}{\tau}\,\underset{\text{illumination}}{\mathbf{l}} \odot \overset{\text{reflectance}}{\mathbf{r}} + \mathbf{z} \leftarrow \text{noise}$$

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

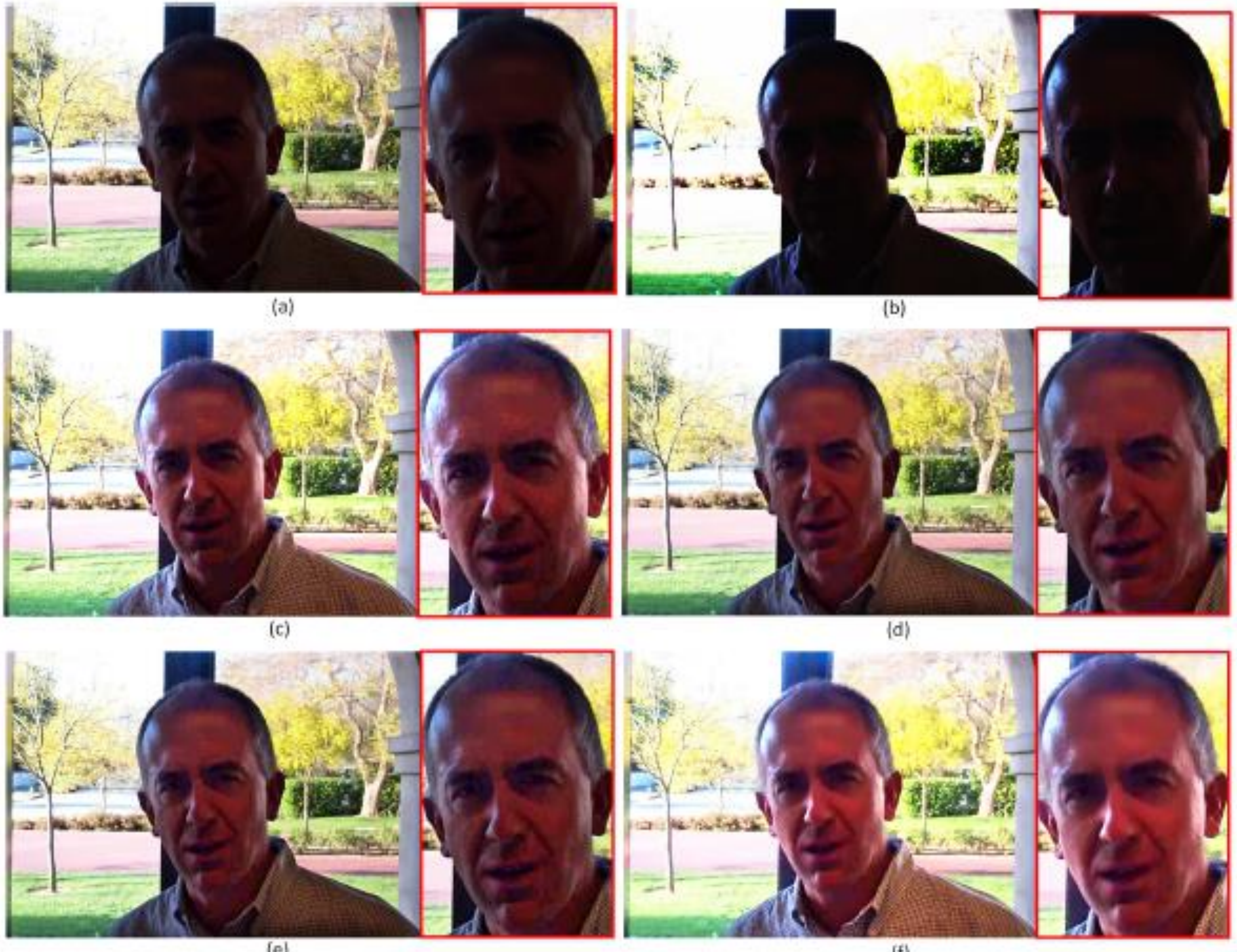
$$\min_{\mathbf{l},\mathbf{r}} \quad \mathbf{l}^\top \left(\mathbf{L}_l + \alpha \mathbf{L}_l^2\right) \mathbf{l} + \mu\, \mathbf{r}^\top \mathcal{L}_r \mathbf{r}$$
$$\text{s.t.} \quad \left(\mathbf{q} - \tfrac{1}{2}\right) \mathbf{Q} \preceq \mathbf{T} \tau\, \mathbf{l} \odot \mathbf{r} \prec \left(\mathbf{q}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

- **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



(a)

(b)

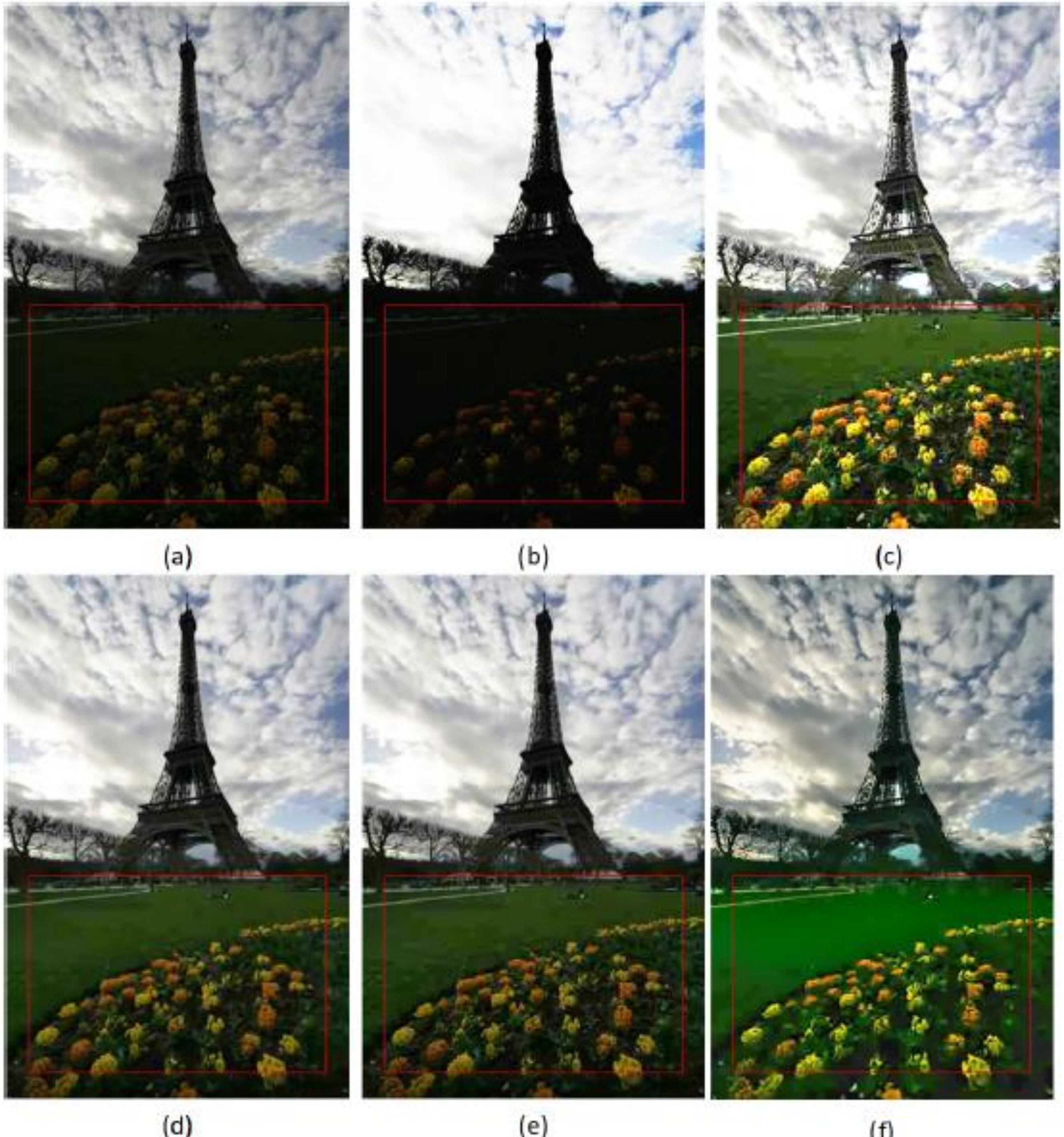
(c)

(d)

(e)

(f)

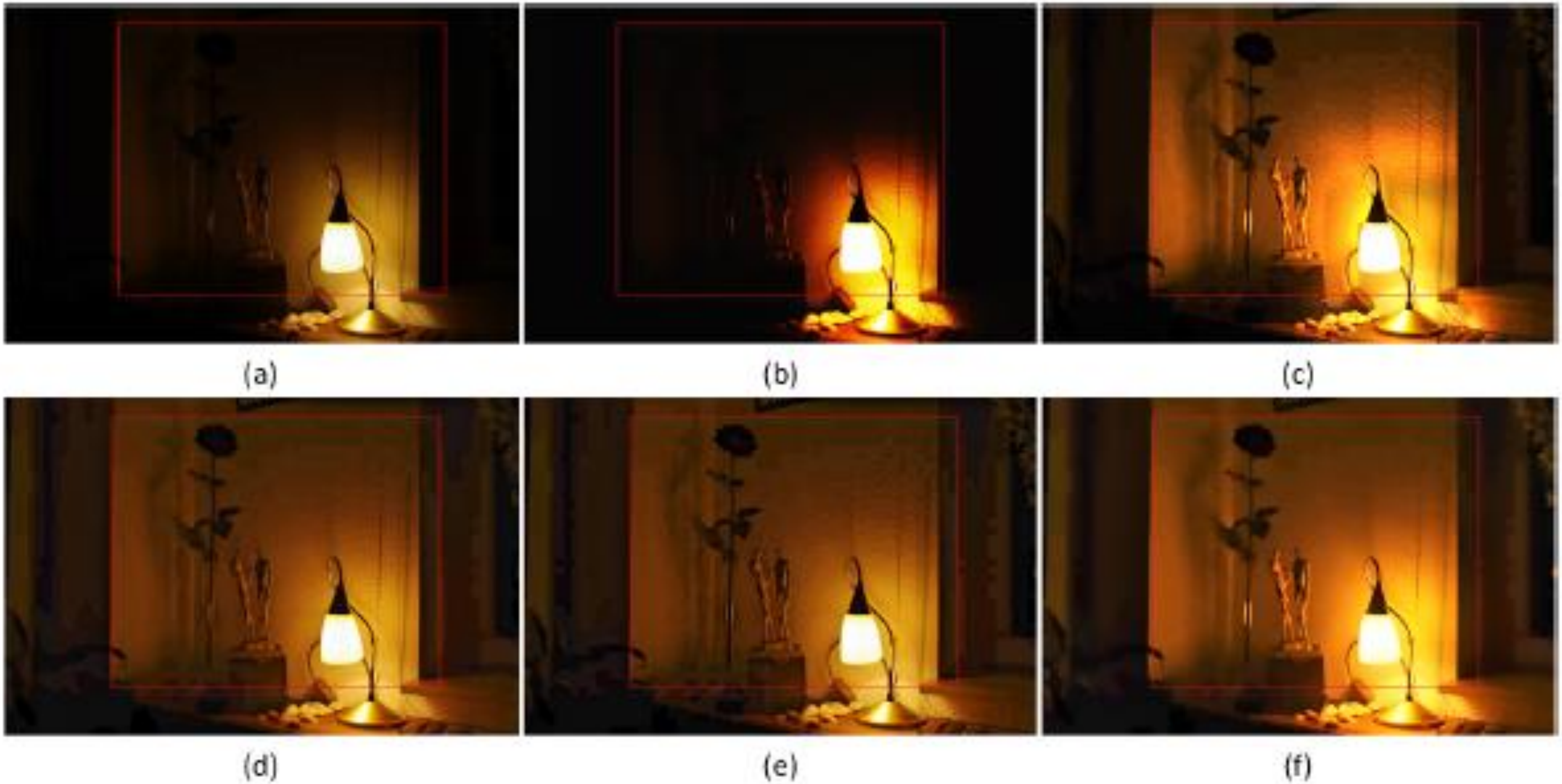
Experimental Results



(a) (b) (c)

(d) (e) (f)

Experimental Results



(a)

(b)

(c)

(d)

(e)

(f)

Graph Total Variation (GTV)

- **Graph Laplacian Regularizer** (GLR):

$$\mathrm{x}^{\mathrm{T}}\mathrm{L}\mathrm{x} = \sum_{i,j} w_{i,j}\left(x_i - x_j\right)^2$$

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

  - System of linear equations: $\left(\mathrm{I}+\mu\mathrm{L}\right)\mathrm{x}^* = \mathrm{y}$

- **Graph Total Variation** (GTV):

$$\min_{x} \|\mathrm{y}-\mathrm{x}\|_2^2 + \mu \sum_{i,j} w_{i,j}\left|x_i - x_j\right|$$

  - L2-L1 norm minimization: primal-dual algorithm, proximal gradient.

[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.

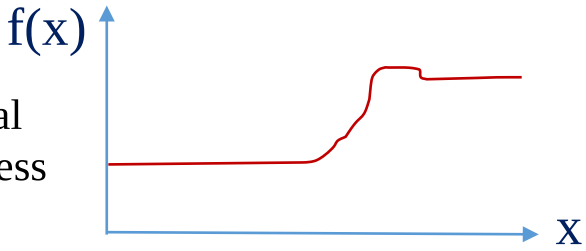
[2] P. Berger, G. Hannak, and G. Matz, “Graph signal recovery via primal-dual algorithms for total variation minimization,” in *IEEE Journal on Selected Topics in Signal Processing*, September 2017, vol. 11, no.6, pp. 842–855.

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**.

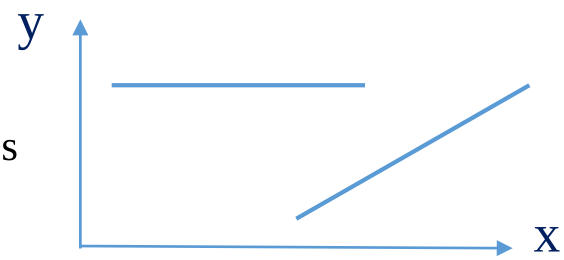
$$\sum_i \left| f(x_i) - f(x_{i-1}) \right|$$

functional smoothness



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

geometry smoothness



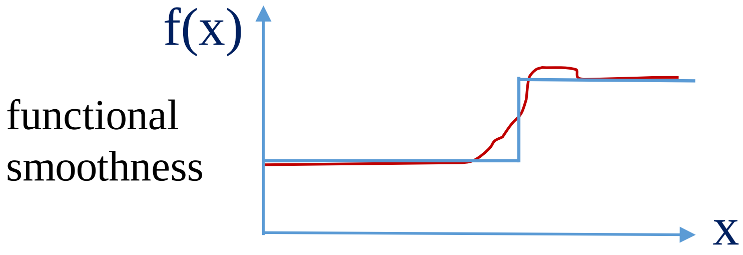
- **Proposal**: Apply GTV is to the surface normals of 3D point cloud—**a generalization of TV to 3D geometry.**

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

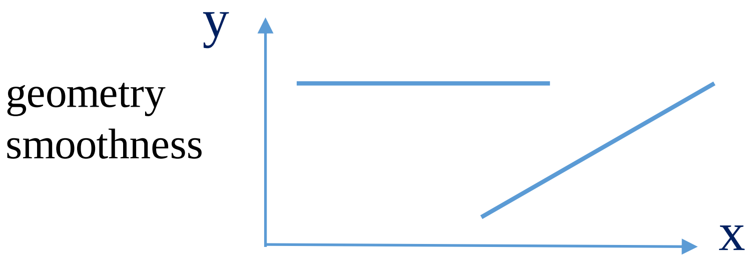
GTV for Point Cloud Denoising

- 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**.

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



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



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

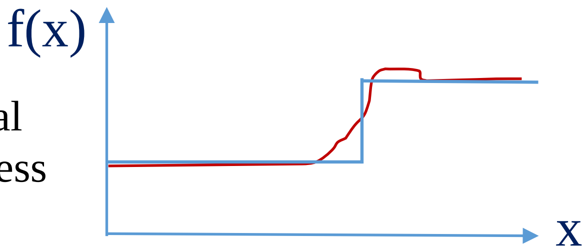
[1] Y. Schoenenberger, J. Paratte, and P. Vandergheynst, "**Graph-based denoising for time-varying point clouds**," in *IEEE 3DTV-Conference*, 2015, pp. 1–4

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**.

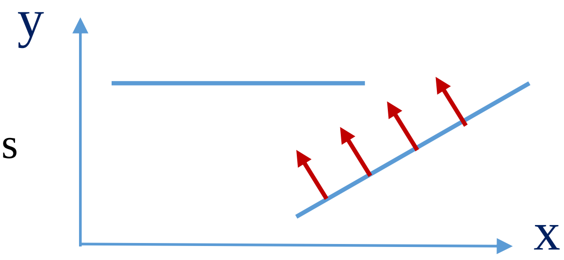
$$\sum_i |f(x_i) - f(x_{i-1})|$$

functional
smoothness



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

geometry
smoothness



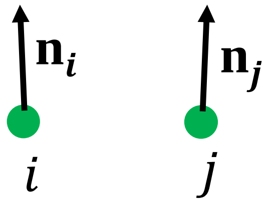
- **Proposal**: Apply GTV is to the surface normals of 3D point cloud—**a generalization of TV to 3D geometry.**

[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:

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



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

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

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

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

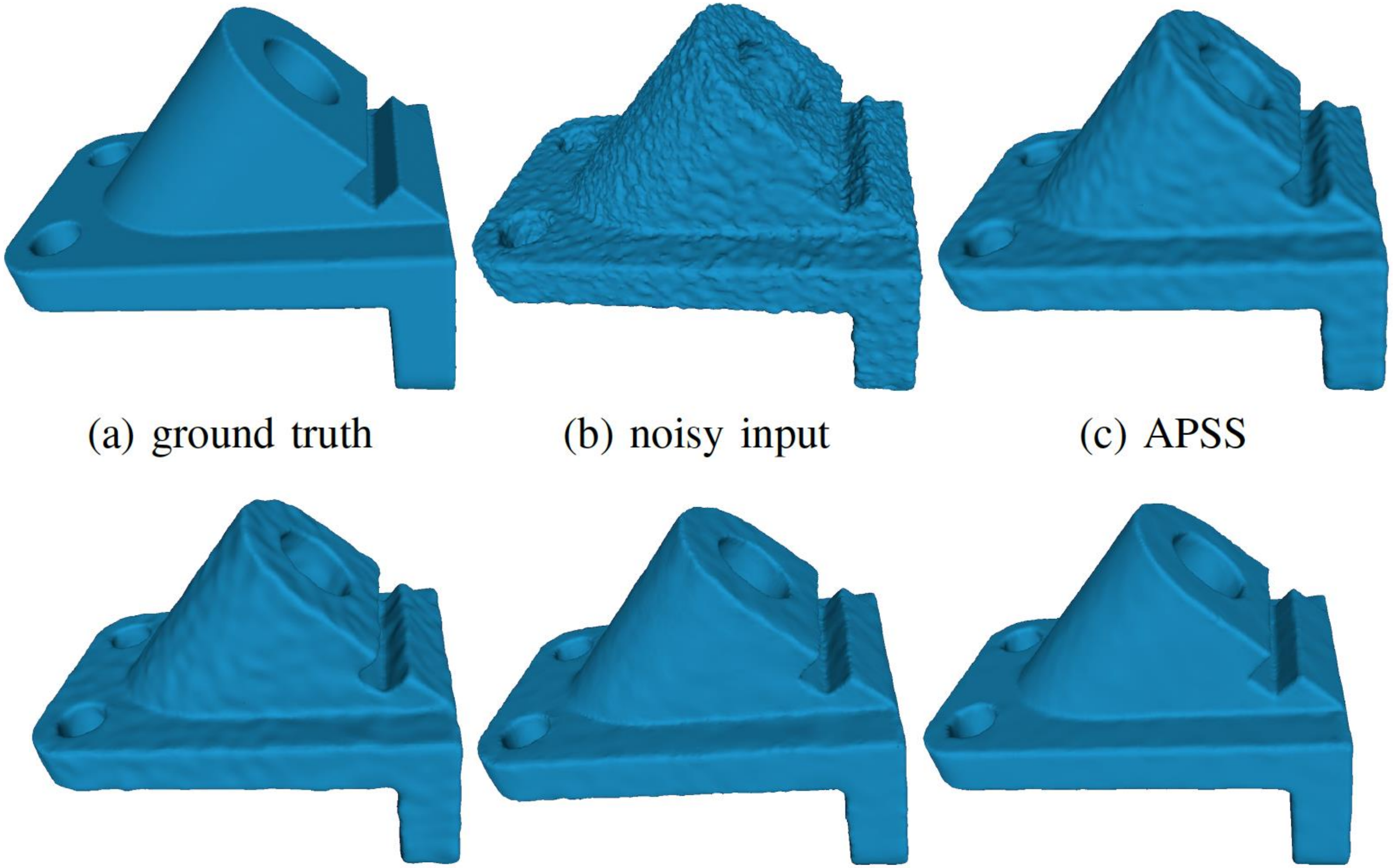
Proposal:

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

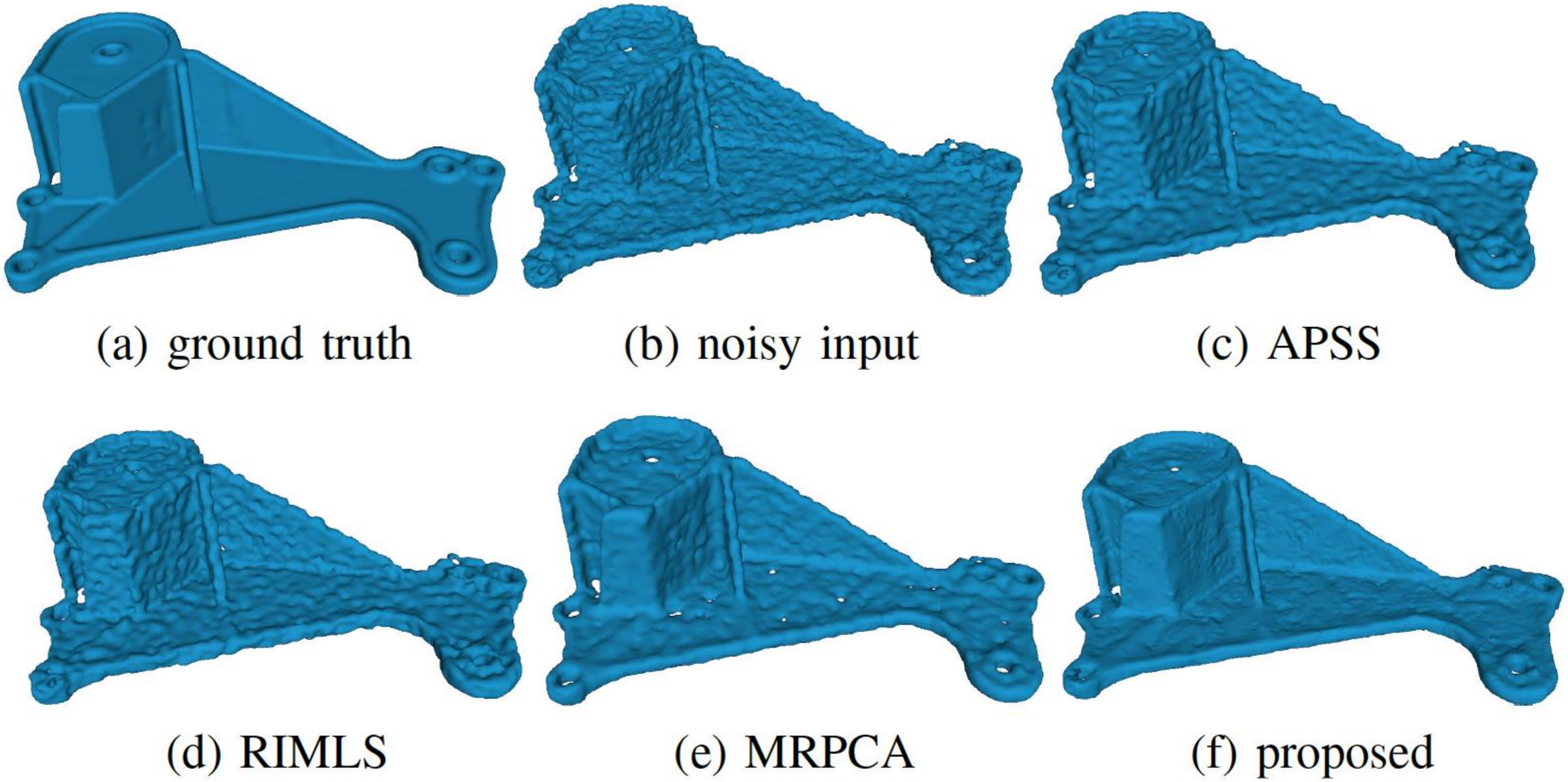
(d) RIMLS

(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 (RGTV)

- **Graph Total Variation** (GTV):

$$\min_{x} \|\mathrm{y} - \mathrm{x}\|_2^2 + \mu \sum_{i,j} \mathrm{w}_{i,j} \left| x_i - x_j \right|$$

- **Reweighted Graph Total Variation** (RGTV):

$$\min_{x} \|\mathrm{y} - \mathrm{x}\|_2^2 + \mu \sum_{i,j} \mathrm{w}_{i,j}\left(x_i, x_j\right) \left| x_i - x_j \right|$$

  - Fix edge weights $w_{i,j}$, solve L2-L1 norm minimization.
  - Update edge weights.

[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.

Reweighted Graph Total Variation (RGTV)

- **Graph Total Variation** (GTV):

$$\min_{x} \|\mathrm{y}-\mathrm{x}\|_2^2 + \mu \sum_{i,j} w_{i,j} \left| x_i - x_j \right|$$

pixel intensity difference

- **Reweighted Graph Total Variation** (RGTV):

$$\min_{x} \|\mathrm{y}-\mathrm{x}\|_2^2 + \mu \sum_{i,j} w_{i,j}\left(x_i, x_j\right) \left| x_i - x_j \right| \qquad w_{i,j} = \exp\left(\frac{-\|x_i - x_j\|_2^2}{\sigma_1^2} \right)$$

- Fix edge weights $w_{i,j}$, solve L2-L1 norm minimization.
- Update edge weights.

[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.

Why RGTV?

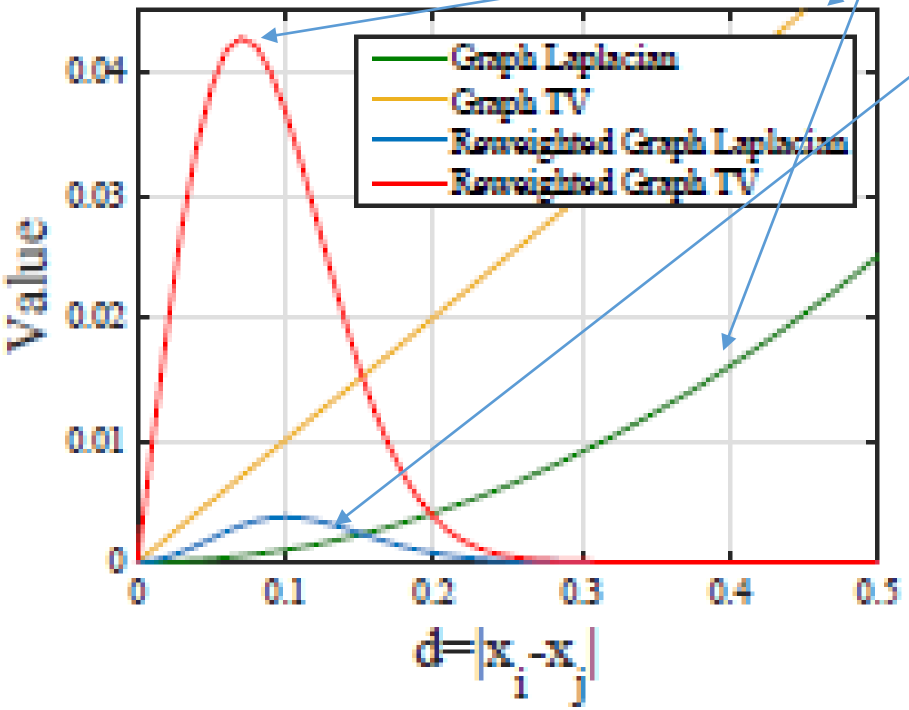
- RGTV promotes PWS signals.
- RGTV has better convergence.

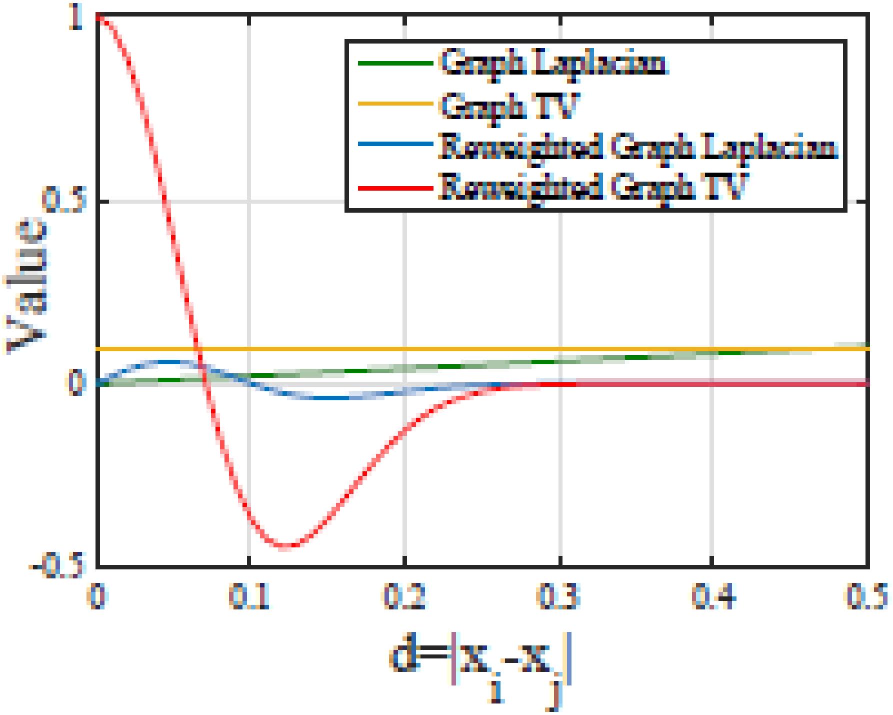
GLR $w_{i,j}\left(x_i - x_j\right)^2$

GTV $w_{i,j}\left|x_i - x_j\right|$

RGLR $\exp\left[-\frac{\left(x_i - x_j\right)^2}{\sigma^2}\right]\left(x_i - x_j\right)^2$

RGTV $\exp\left[-\frac{\left(x_i - x_j\right)^2}{\sigma^2}\right]\left|x_i - x_j\right|$





[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.

Background for Image Deblurring

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

$$y = k \otimes x$$

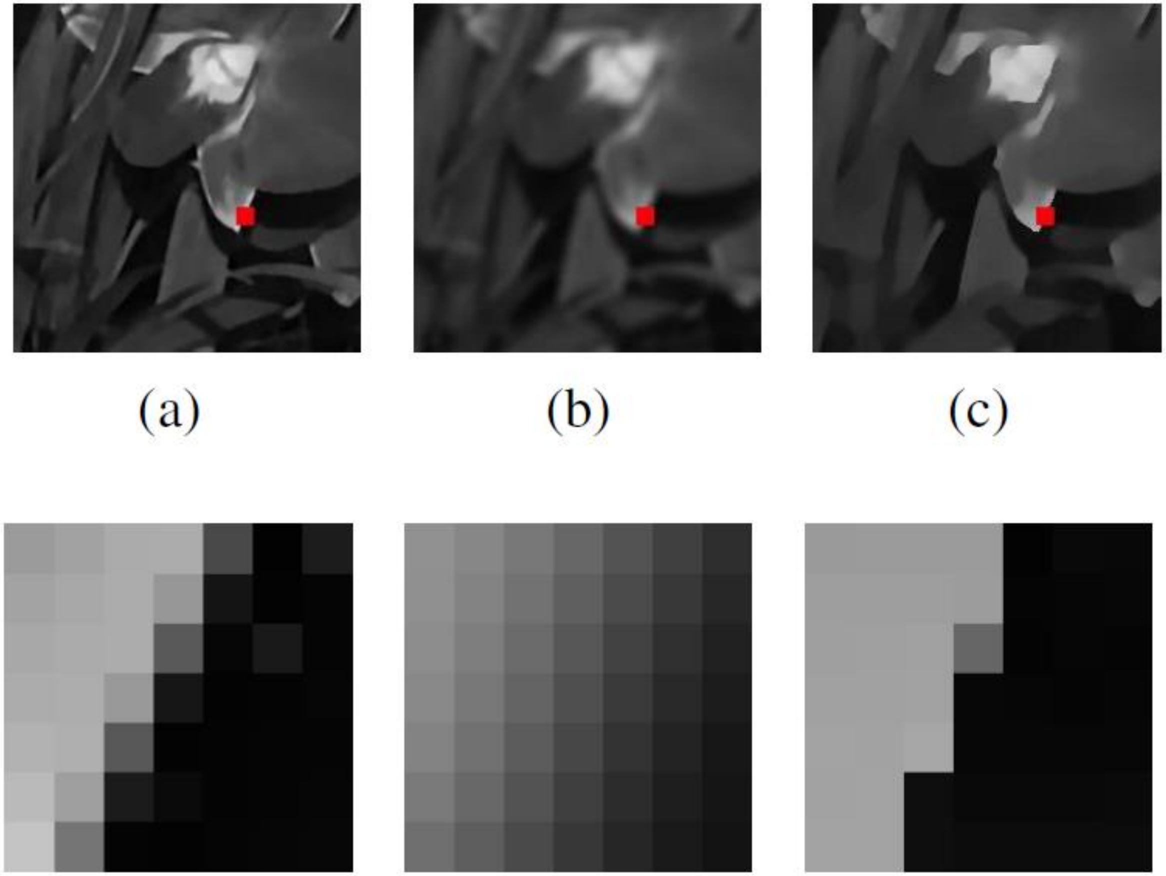
where y is the blurry image, k is the blur kernel, x is the original sharp image.

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

[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.

Observation

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

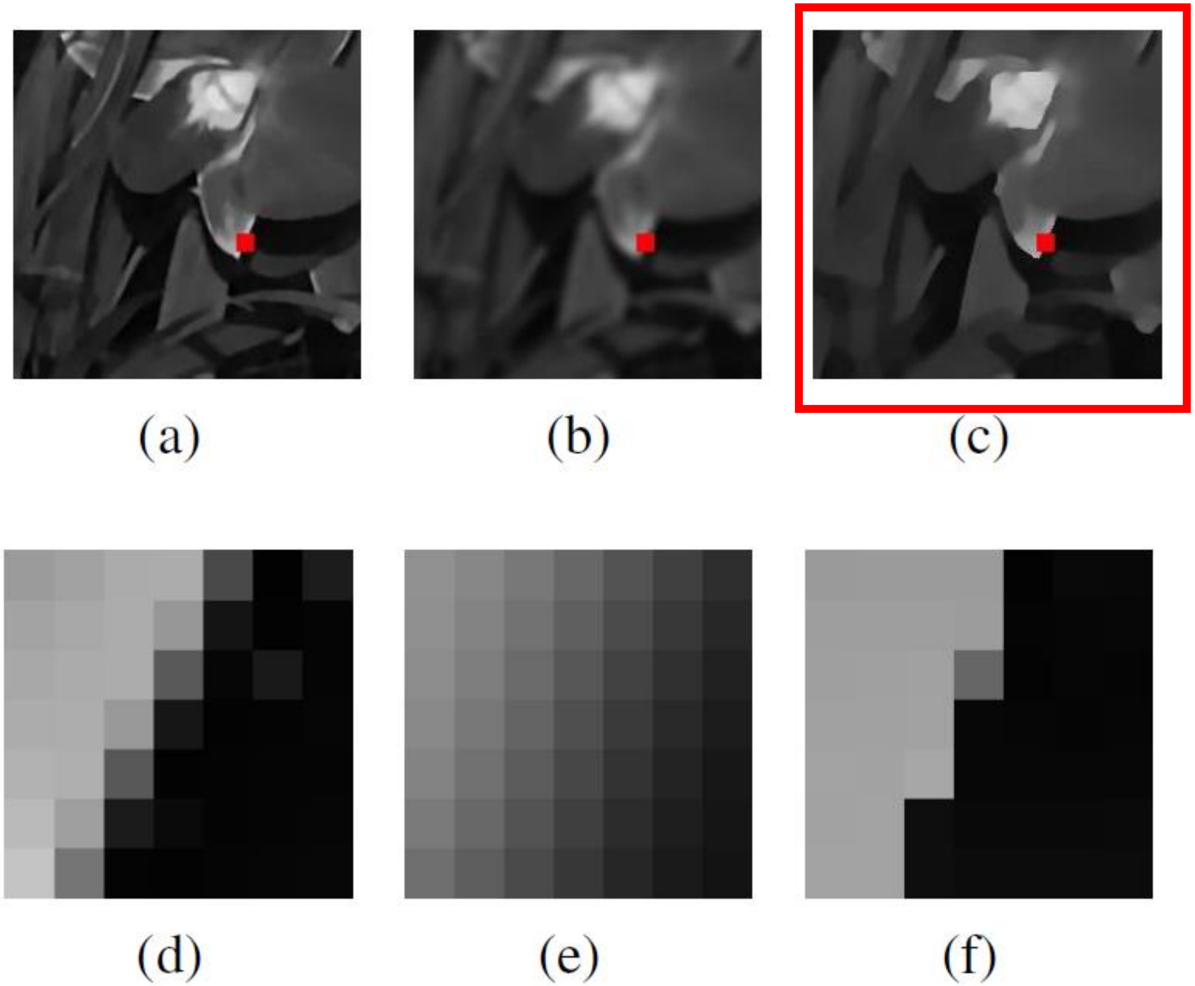


(a) (b) (c)

(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,

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

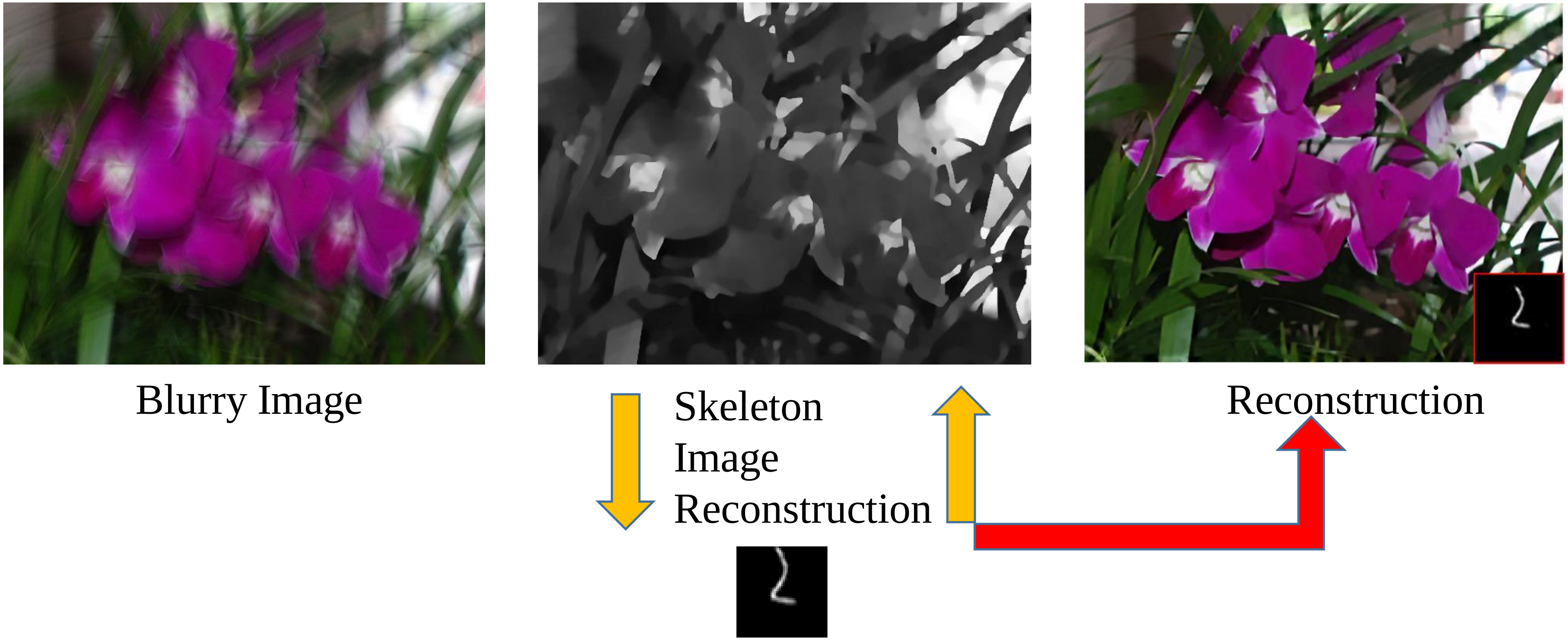
- Assume L_2 norm for fidelity term $\varphi(\cdot)$.
- $\theta_x(\cdot) = RGTV(\cdot)$.
- $\theta_k(\cdot) = ||\cdot||_2$, assuming zero mean Gaussian distribution of k.

- RGTV is non-differentiable and non-convex.

Solution:

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

Workflow



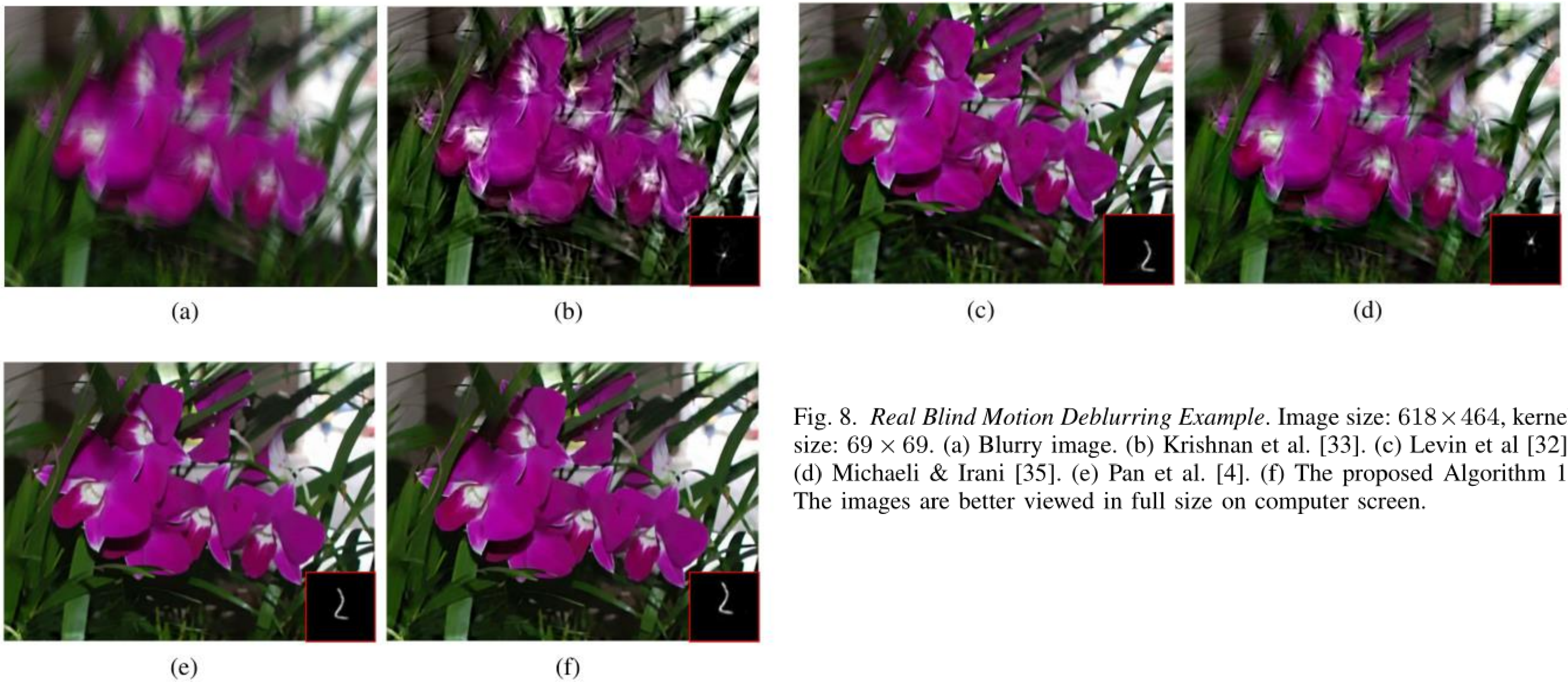
Blurry Image

Skeleton Image Reconstruction

Reconstruction

Kernel Estimation

Experimental Results



(a)

(b)

(c)

(d)

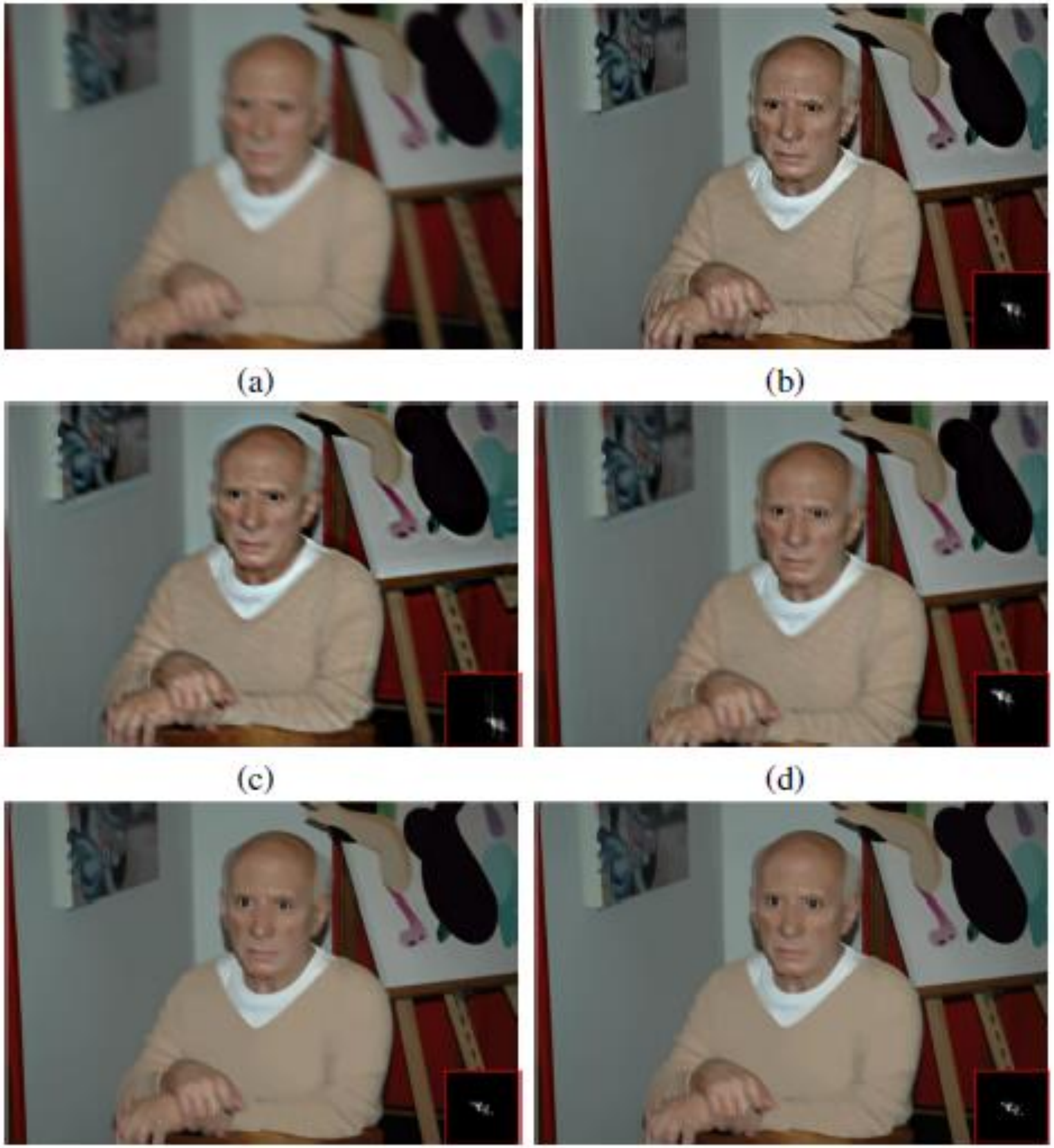
(e)

(f)

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



(a)

(b)

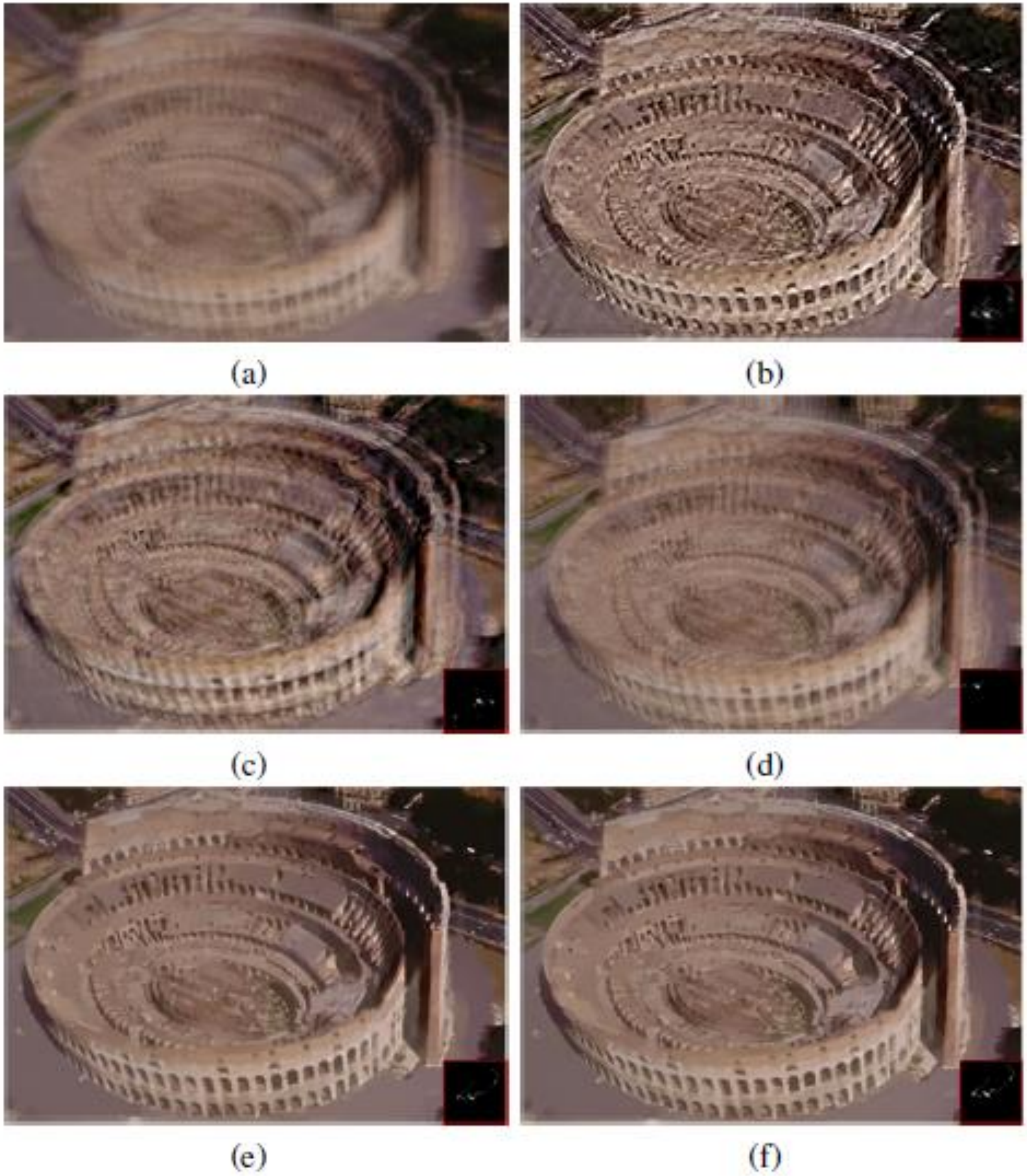
(c)

(d)

(e)

(f)

Experimental Results



(a)

(b)

(c)

(d)

(e)

(f)

Outline

- GSP Fundamentals
- GSP for Image Compression
  - Graph Fourier Transform
- GSP for Inverse Imaging
  - Graph Laplacian Regularizer (GLR)
  - Reweighted Graph TV
- Summary

Summary

- Frequencies for Graphs
  - GFT 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

Ongoing Work:

- Hybrid graph-based / data-driven approach [1]

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

Exercise 1: Codec Design for Image Compression

1. Identify Image Compression Application
   - e.g., image, video, point cloud, light field, hyperspectral image

2. Graph Design
   - Connectivity
   - Edge Weight Assignment

3. Compression Tool
   - Implementation of graph transform
   - Coding of side information(?)

Exercise 2: Problem Formulation for Inverse Imaging

1. Identify Inverse Imaging Application
   - e.g., denoising, super-resolution, soft-decoding of JPEG images, demosaicking, colorization, inpainting

2. Graph Design
   - Connectivity
   - Edge Weight Assignment

3. Problem Formulation
   - Graph-signal prior: GLR, GTV, RGTV
   - Constraints: quantization bin, 8-bit pixel representation
   - How to solve it?

Exercise 1: Codec Design for Image Compression

1. Identify Image Compression Application
   - e.g., image, video, point cloud, light field, hyperspectral image

2. Graph Design
   - Connectivity
   - Edge Weight Assignment

3. Compression Tool
   - Implementation of graph transform
   - Coding of side information(?)

Exercise 2: Problem Formulation for Inverse Imaging

1. Identify Inverse Imaging Application
   - e.g., denoising, super-resolution, soft-decoding of JPEG images, demosaicking, colorization, inpainting

2. Graph Design
   - Connectivity
   - Edge Weight Assignment

3. Problem Formulation
   - Graph-signal prior: GLR, GTV, RGTV
   - Constraints: quantization bin, 8-bit pixel representation
   - How to solve it?