Gene Cheung
National Institute of Informatics
21st May, 2015

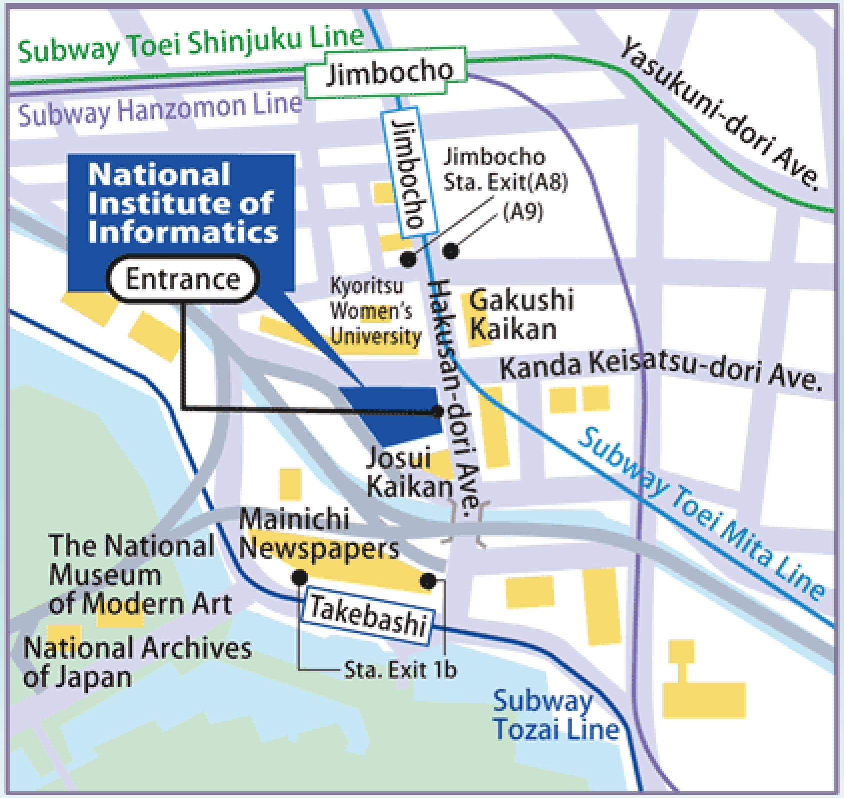
Graph-based Depth Image Processing

.

NII Overview

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

- **National Institute of Informatics**
- Chiyoda-ku, Tokyo, Japan.
- Government-funded research lab.



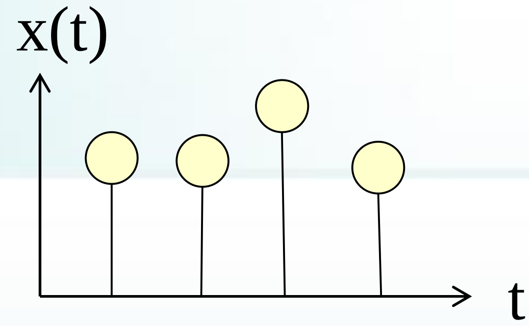
- Offers graduate courses & degrees through **The Graduate University for Advanced Studies**.
- 60+ faculty in "**informatics**": quantum computing, discrete algorithms, database, machine learning, computer vision, speech & audio, image & video processing.

- **Get involved!**
  - 2-6 month Internships.
  - Short-term visits via MOU grant.
  - Lecture series, Sabbatical.

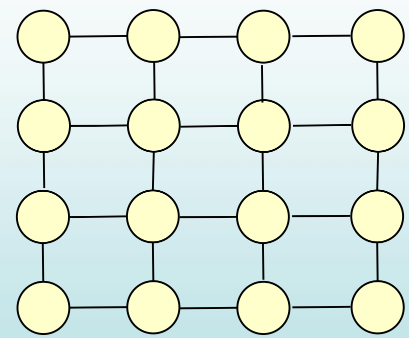
Outline

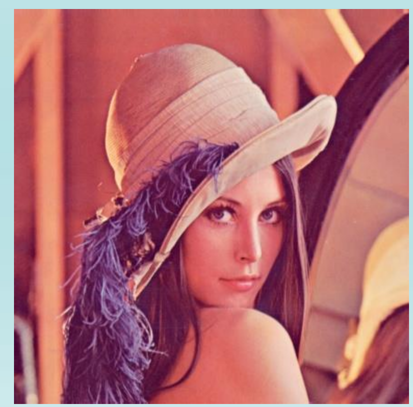
- Traditional vs. Graph Signal Processing
- Graph Fourier Transform (GFT)
- Depth Map Compression
- Depth Map Denoising
- Depth Map Interpolation

Traditional Signal Processing



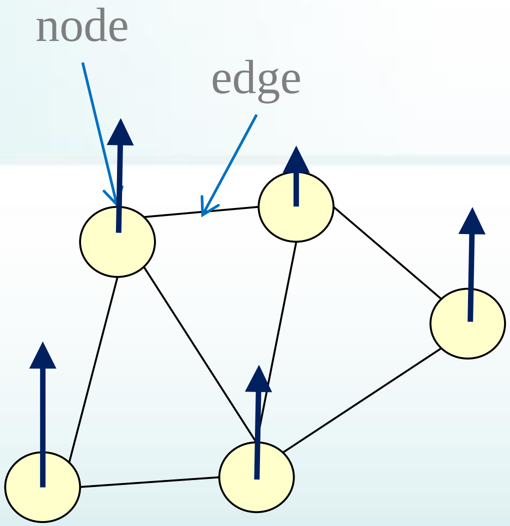
- Traditional discrete signals live on regular data kernels (**unstructured**).
  - Ex.1: audio / music / speech on regularly sampled timeline.
  - Ex.2: image on 2D grid.
  - Ex.3: video on 3D grid.





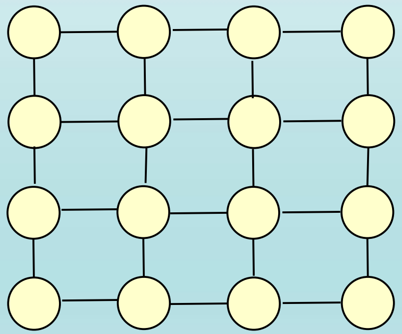
- Wealth of SP tools (transforms, wavelets, dictionaries, etc) for tasks such as:
  - compression, denoising, classification.

Graph Signal Processing



- Signals live on graph.
  - Graph is sets of nodes and edges.
  - Edges reveals *node-to-node relationships*.
- Data kernel itself is **structured**.

1. Data domain is naturally a graph.
   - Ex.1: posts on social networks.
   - Ex.2: temperatures on sensor networks.
2. **Embed signal structure in graph.**
   - Ex.1: images: 2D grid → structured graph.

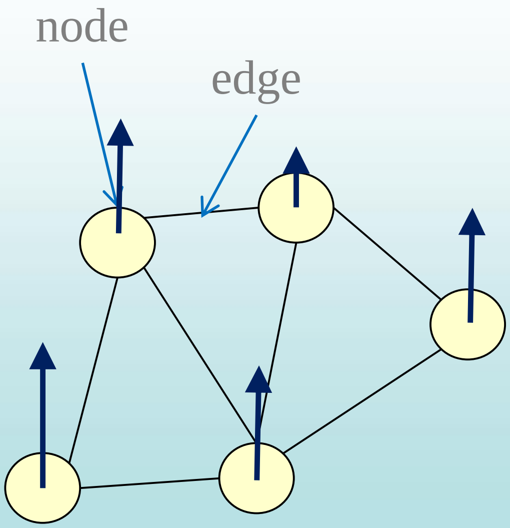


Graph Signal Processing (GSP) addresses the problem of processing signals that live on graphs.

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.

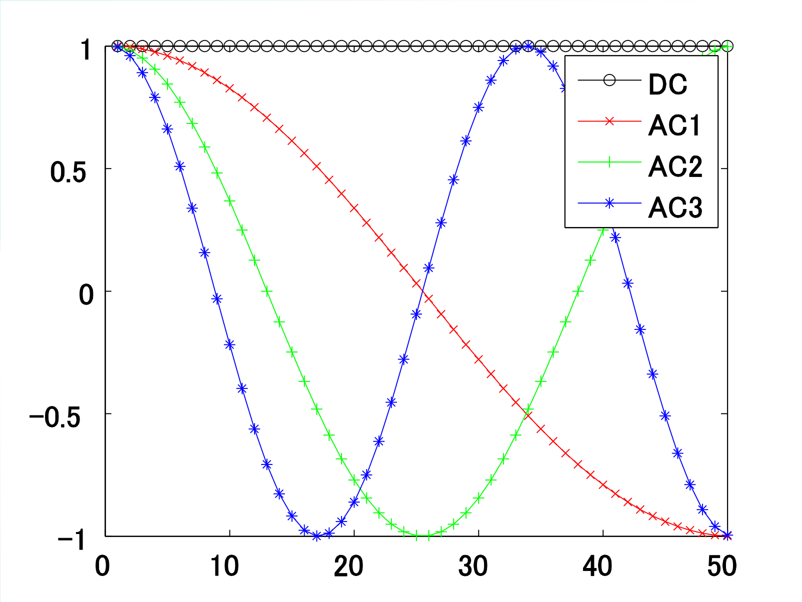


Outline

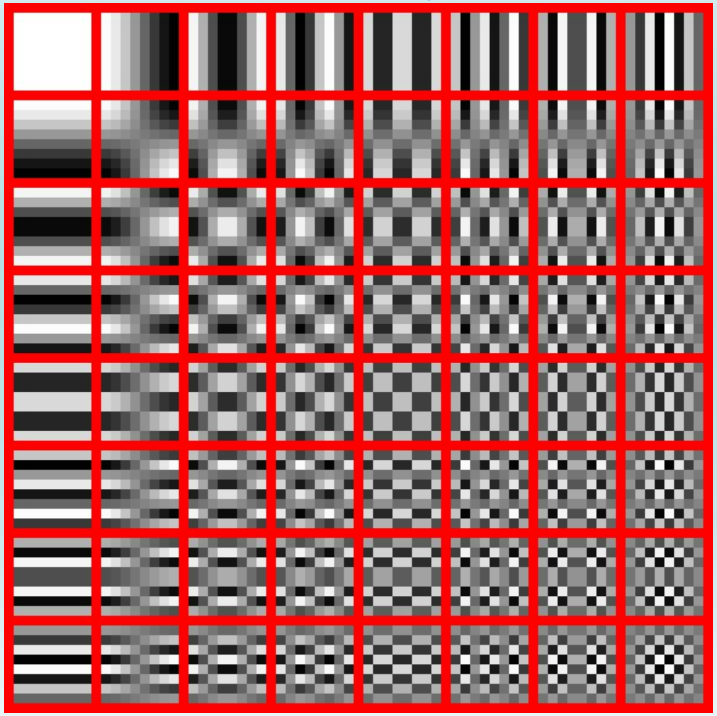
- Traditional vs. Graph Signal Processing
- Graph Fourier Transform (GFT)
- Depth Map Compression
- Depth Map Denoising
- Depth Map Interpolation

Transforms for Signals

- **Transform**: Set of orthogonal basis.
- Notion of *frequency*.
- Ex.1: 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 is set of outer-product of 1D DCT basis in x- and y-dimension.



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

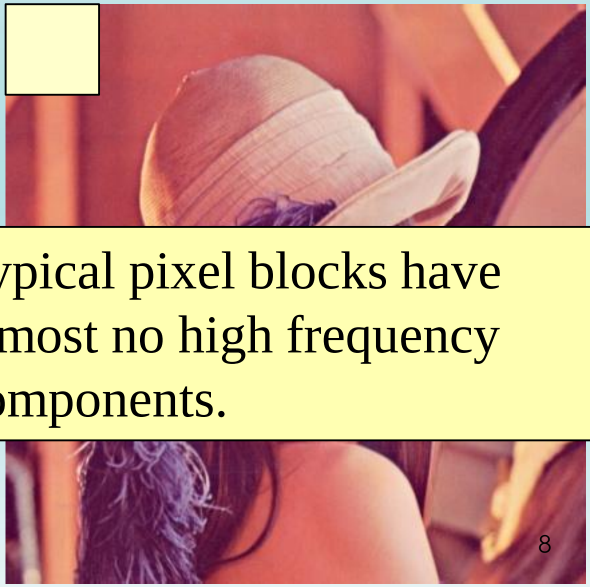
desired signal

transform

transform coeff.

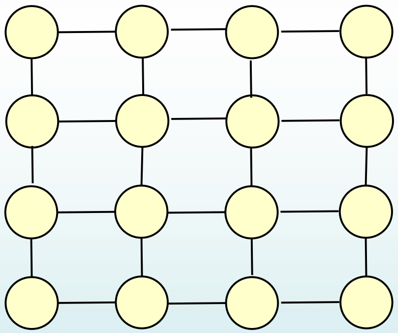
$$x = a_1\phi_1 + a_2\phi_2 + \cdots$$

$$x = \sum_i a_i \phi_i$$



Typical pixel blocks have almost no high frequency components.

Graph Fourier Transform (GFT) for Graph-signals

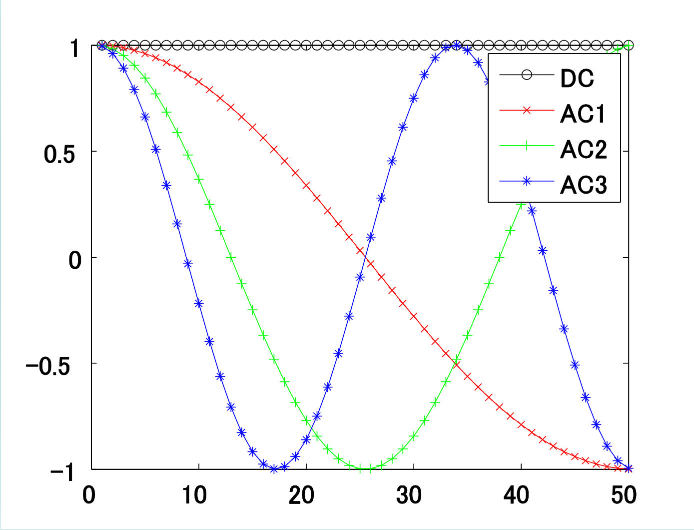


Graph Fourier Transform:

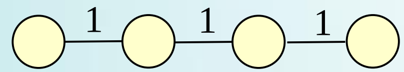
- *Signal-adaptive* transform:

  1. If two connected pixels are "similar", then edge weight is large → adjacency matrix A.
  2. Compute ***graph Laplacian*** L = D-A.
  3. Perform eigen-decomposition on L for GFT.

$$x = \sum_i a_i \varphi_{\mathrm{i}}$$



- **Intuition**: Embed geometric structure of signal as edge weights in graph.



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

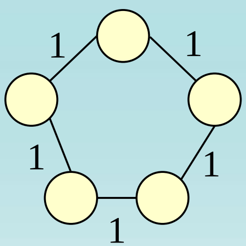
L for 4-node un-weighted line graph

Facts of Graph Laplacian & GFT

- L is a high-pass filter.
- $\mathrm{x}^T \mathrm{L} \mathrm{x}$ is one measure of variation in signal.

$$\mathrm{x}^T \mathrm{L} \mathrm{x} = \frac{1}{2} \sum_{i,j} w_{i,j} \left(x_i - x_j\right)^2 = \sum_i \lambda \alpha_i^2$$

- L is *positive semi-definite*; eigenvalues λ's ≥ 0 → eigenvalues are ***graph frequencies***.
- L = D – A; λ = 0 must be eigenvalue w/ vector $[1 \ldots 1]^T$.
- Use eigenvectors for spectral decomposition of signal.
  - GFT defaults to ***DCT*** for un-weighted connected line.
  - GFT defaults to ***DFT*** for un-weighted connected circle.

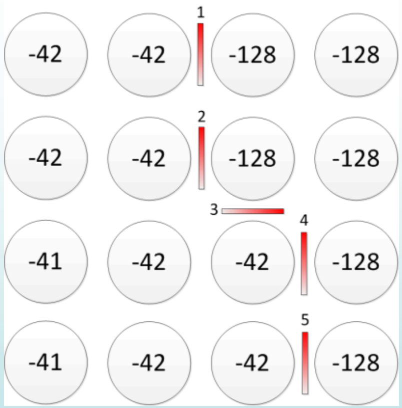


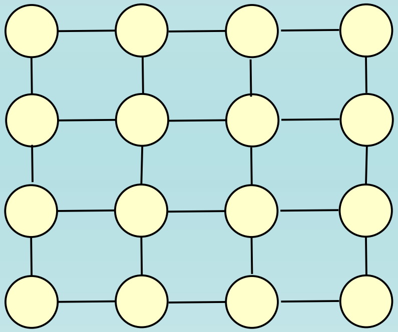
Usage Example: first non-zero eigenvalue → *spectral clustering* (Shi & Malik'00).

Outline

- Traditional vs. Graph Signal Processing
- Graph Fourier Transform (GFT)
- Depth Map Compression
- Depth Map Denoising
- Depth Map Interpolation

Depth Map Compression





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

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

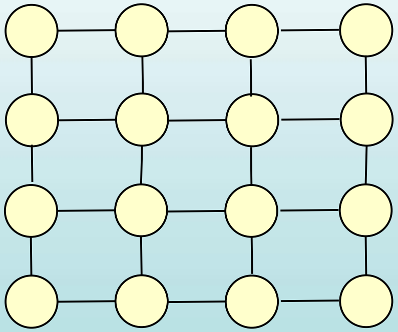
$$\alpha = \underbrace{\Psi}_{\text{GFT}} x$$

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

Depth Map Compression

Q: Why GFT leads to sparseness?

Ans 1: **Capture statistical structure of signal** in edge weights of graph.



- Adjacent pixel correlation 0 or 1 for piecewise smooth (PWS) signal.
- Can be shown GFT approximates KLT given *Gaussian Random Markov Field (GRMF)* model.

Ans 2: Avoid filtering across sharp edges.

a 4x4 block

GFT $\alpha_1 = \begin{bmatrix} 237 & 0 & 0 & 0 \\ 163 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

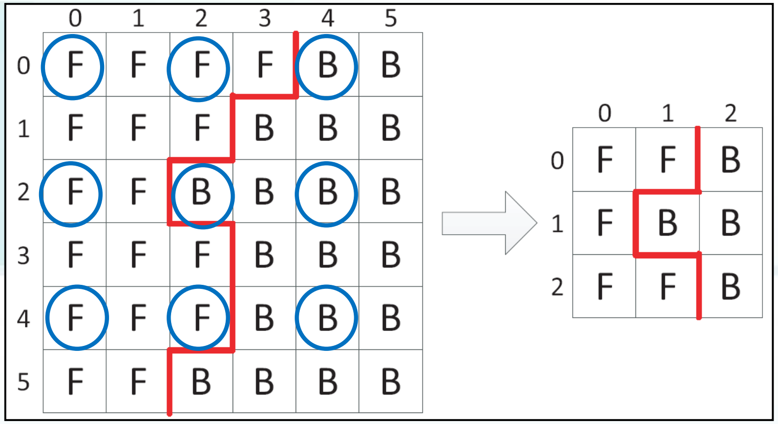
DCT $\alpha_2 = \begin{bmatrix} 285 & -29 & -5 & -4 \\ 16 & 1 & -16 & -4 \\ -5 & 3 & 5 & -7 \\ -1 & -4 & 1 & 9 \end{bmatrix}$

filtering operation

$$\alpha = \Psi x$$

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

Depth Map Compression using Multi-resolution GFT



DCT

GFT

- **Idea**:
  - LP-filter & down-sample signal before GFT.

$$y = DHx$$

LP & down-sample operators

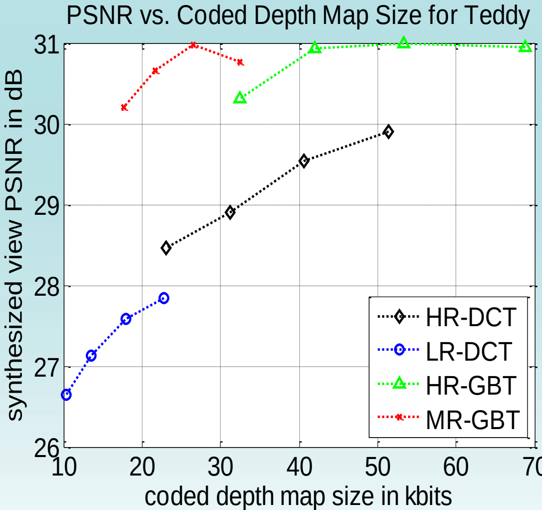
  - GFT on LR block.

$$\alpha = \Psi y$$

LR-GFT

- **Details**:

1. **Enc**: Detect & encode HR edges (***structure***).
2. **Enc**: Encode down-sampled LR block using GFT (***texture***).
3. **Dec**: Deduce LR edges from HR edges (***structure***).
4. **Dec**: Decode LR block in GFT, up-sample & interpolate using HR edges (smooth ***texture***).



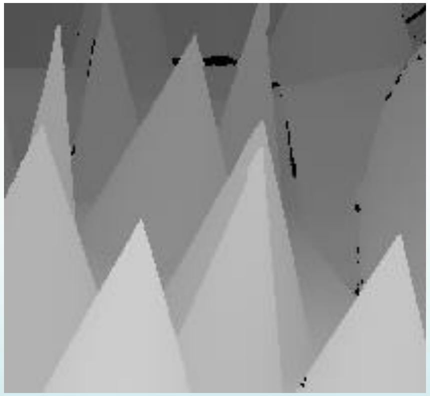
Results: up to 68% bitrate reduction compared to HR-DCT.

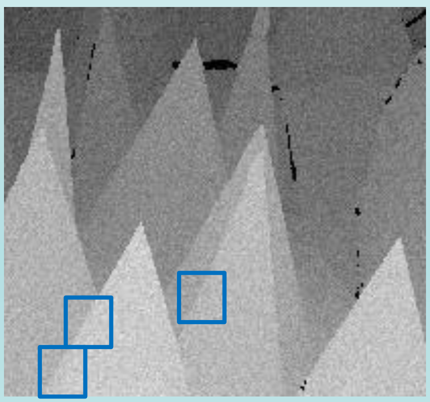
Wei Hu, Gene Cheung, Antonio Ortega, Oscar Au, "**Multiresolution Graph Fourier Transform for Compression of Piecewise Smooth Images**," *IEEE Transactions on Image Processing*, vol.24, no.1, pp.419-433, January 2015.

Outline

- Traditional vs. Graph Signal Processing
- Graph Fourier Transform (GFT)
- Depth Map Compression
- Depth Map Denoising
- Depth Map Interpolation

Depth Image Denoising





- **Problem**:
  - Acquired depth images are noisy.

desired signal

observation $\;\; y = x + v \;\;$ noise

- **Strong signal prior**: piecewise smooth (*sparsity in GFT*).
- **Self-similarity** in images (non-local means#).

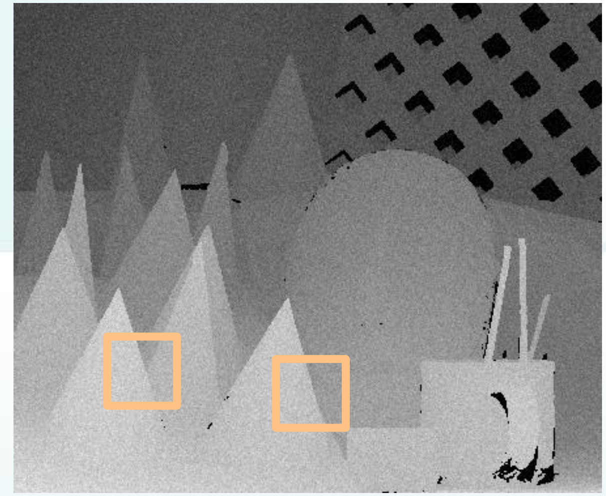
- **Our algorithm** (in a nutshell):
  1. Identify similar patches (*same structure*).
  2. Compute average patch for cluster, compute GFT.

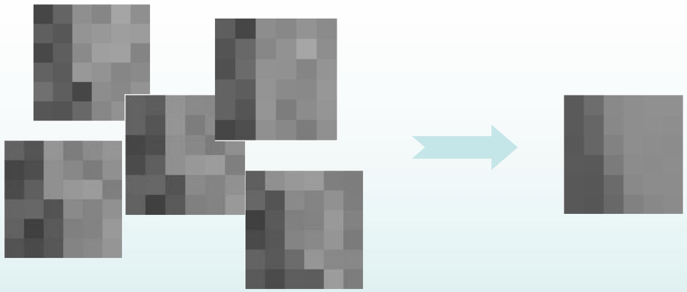
up sparsification.

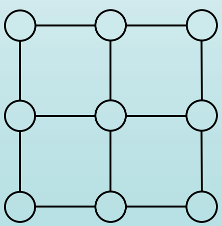
Intuition:
Sparsely represented signal = Denoised signal

A. Buades, B. Coll, J. M. Morel, "**A review of image denoising algorithms, with a new one**," *Multiscale Modeling Simulation (SIAM Interdisciplinary J.)*, vol.4., no.2, pp.490-530, 2005.

Denoising Algorithm







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

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

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

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

common GFT from avg. patch

observation *i*

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

code vector for observation *i*

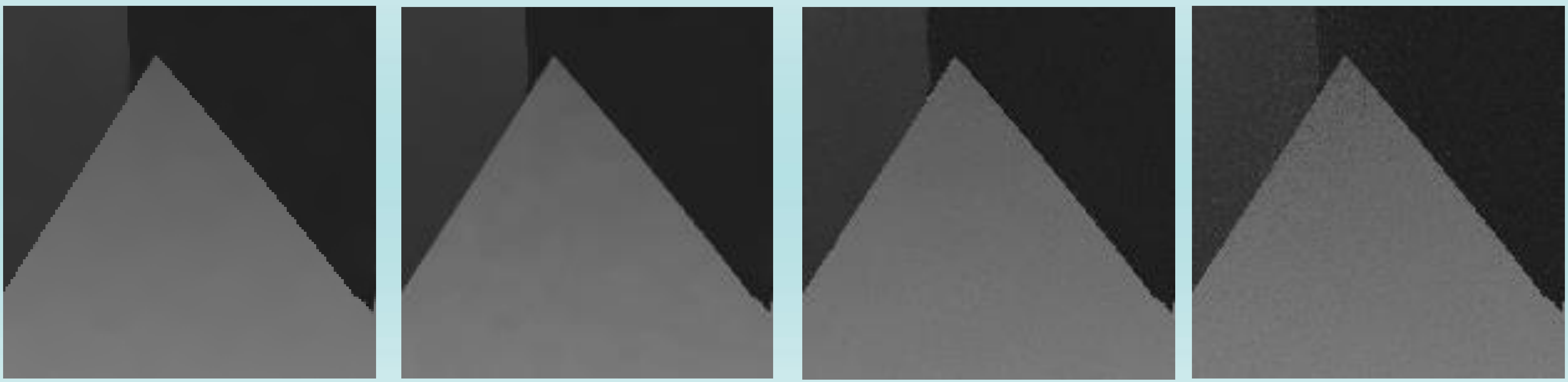
Algorithm:

1. Identify similar patches, compute average patch. (**self-similarity**)
2. Given average patch, compute "similarity" between adjacent pixels. Construct graph.
3. Compute graph transform (GFT).
4. Given GFT, seek sparse representation.

*Wei Hu, Xin Li, Gene Cheung, Oscar Au, "**Depth Map Denoising using Graph-based Transform and Group Sparsity**," *IEEE International Workshop on Multimedia Signal Processing*, Pula (Sardinia), Italy, October, 2013. (**Top 10% paper award**.)

Depth Image Denoising

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

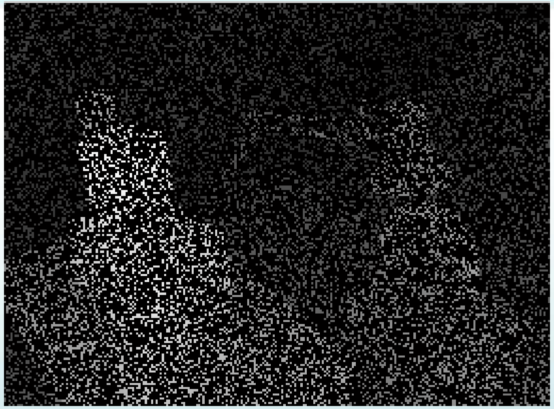


NLGBT BM3D NLM BF

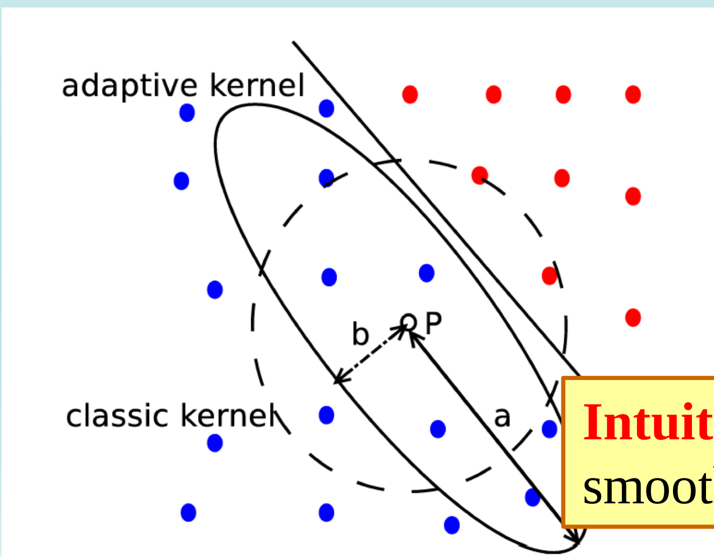
Outline

- Traditional vs. Graph Signal Processing
- Graph Fourier Transform (GFT)
- Depth Map Compression
- Depth Map Denoising
- Depth Map Interpolation

Depth Image Interpolation



Depth image missing 85% pixels



- **Problem**:
  - Fill holes in sparsely sampled depth images.
- **Idea**:*
  1. Find right graph for missing pixels.
     - Adaptive kernel
  2. Compute edge weights using initial values.

$$\mathrm{w}_{i,j} = \exp\left\{ -\frac{\left| y_i - y_j \right|^2}{\sigma^2} \right\}$$

  3. Find smooth graph-signal given observations.

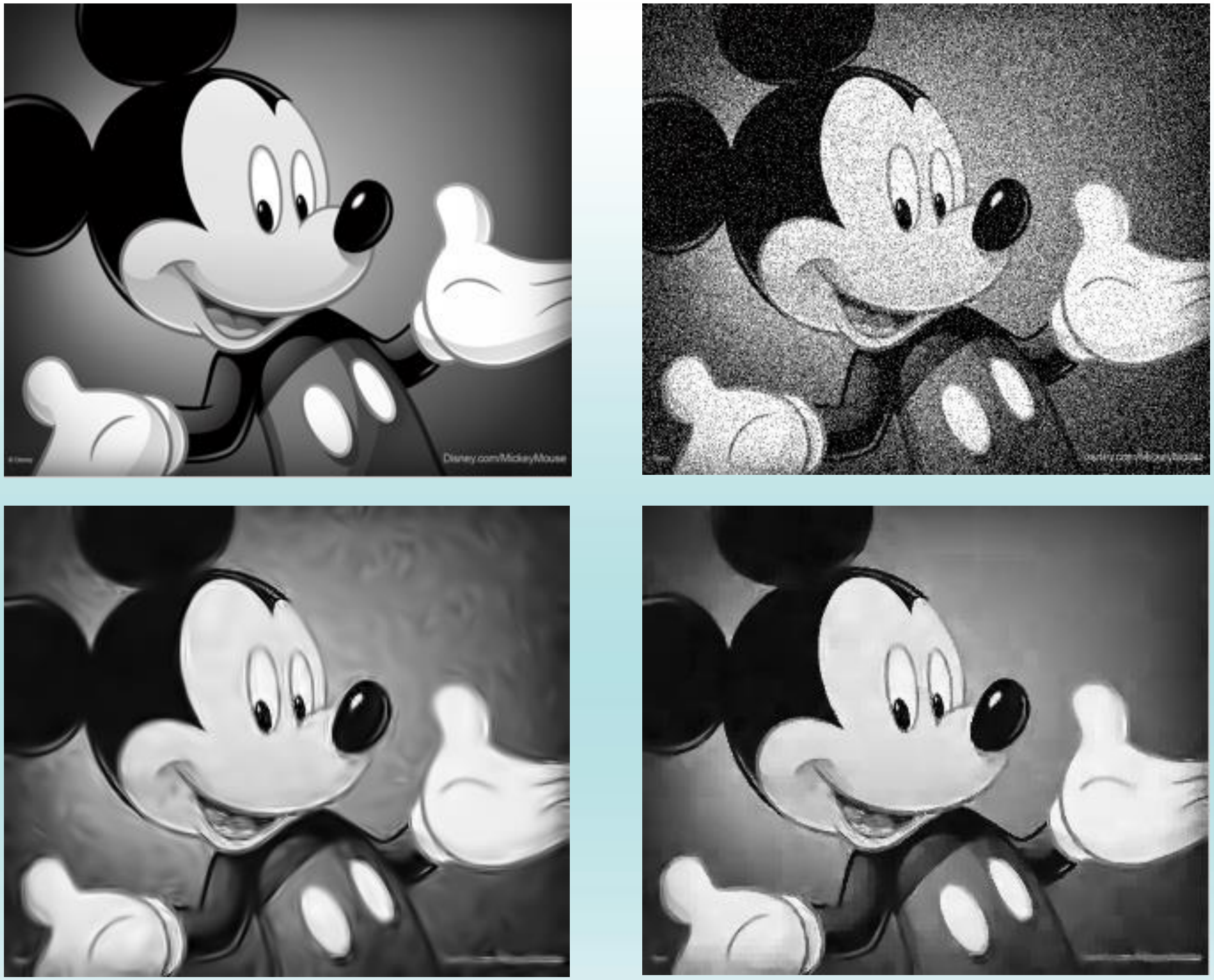
$$\ldots y_i \|_2^2 + \lambda \mathrm{x}^t \mathrm{L} \mathrm{x}$$

i

Intuition:
smooth signal = interpolated signal

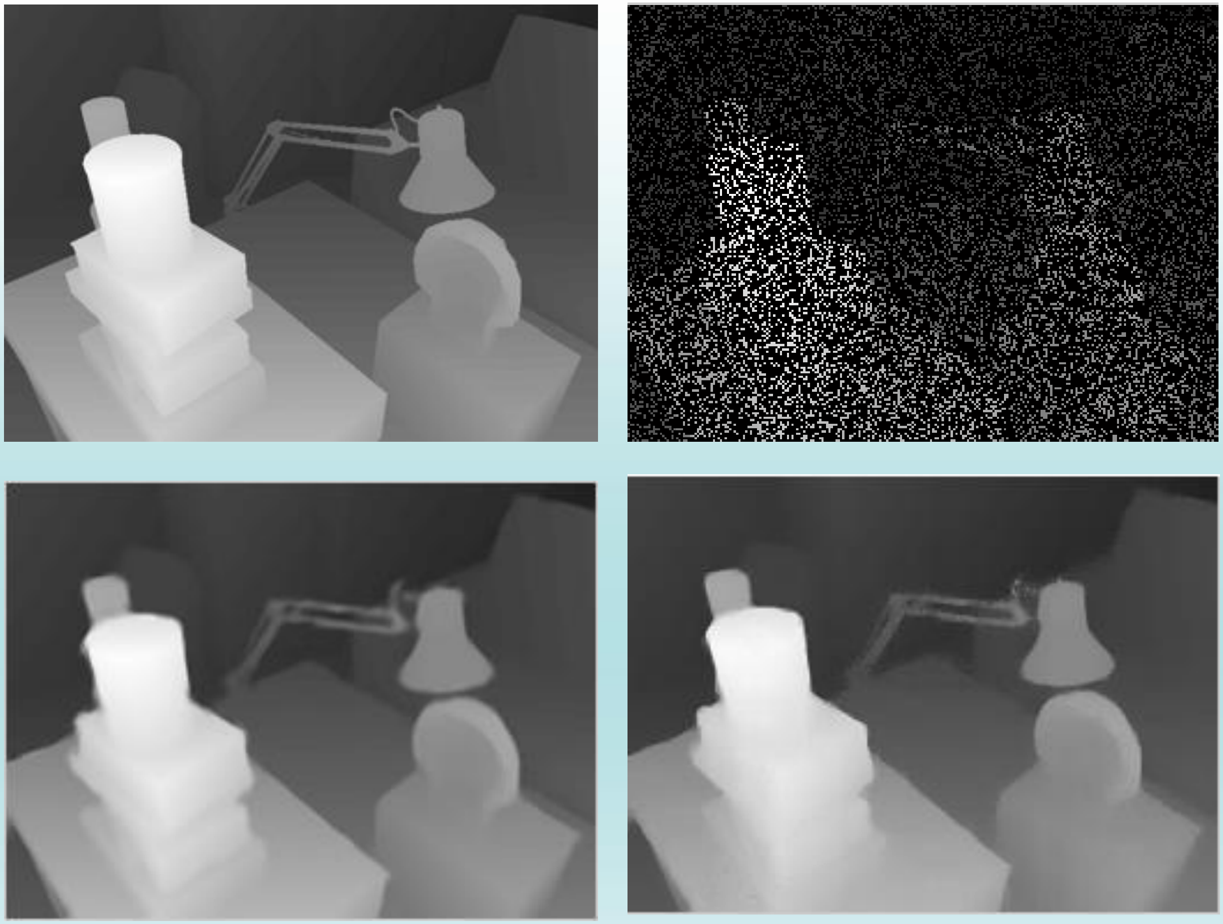
*Yu Mao, Gene Cheung, Yusheng Ji, "**Image Interpolation During DIBR View Synthesis Using Graph Fourier Transform**," *3DTV-Conference 2014*, Budapest, Hungary, July, 2014.

Denoising Results



Original	Noised (theta = 20)
LARK (26.99dB)	Our Proposal (27.36dB)

Interpolation Results



Original	Partial sample
LARK (34.82dB)	Our Proposal (35.31dB)

Summary & Open Problems

- **By embedding image structure onto a graph, signal is smooth wrt graph.**
- **Current Work**:
  - GSP for joint denoising / SR of GPWS images (ICIP'14).
  - GSP for bit-depth enhancement in images (ICIP'14).
  - Variants of graph transforms (in submitted SPL).
- **Open Questions**:
  - Appropriate graph?
    - Optimize desired signal and graph simultaneously.
  - Given graph, appropriate set of basis?
    - Other transforms, (biorthogonal) wavelets, dictionaries.

Acknowledgement

Collaborators:

- Y. Mao, Prof. Yusheng Ji (NII, Japan)
- W. Sun, W. Hu, P. Wan, W. Dai, J. Pang, J. Zeng, A. Zheng, Prof O. Au (HKUST, HK)
- Y.-H. Chao, Prof. A. Ortega (USC, USA)
- D. Florencio, C. Zhang, P. Chou (MSR, USA)
- Y. Gao, Prof. J. Liang (SFU, Canada)
- T. Maugey, L. Toni, B. Motz, A. De Abreu, P. Frossard (EPFL, Switzerland)
- B. Machiavello, C. Dorea, M. Hung (University of Brasilia, Brazil)
- W.-t. Tan (formerly HP Labs, now Cisco, USA)
- C. Yang, V. Stankovic (U of Strathclyde, UK)