Graph Signal Processing for Image Coding & Restoration
Acknowledgement

Collaborators:

• M. Kaneko (NII, Japan)
• A. Ortega (USC, USA)
• D. Florencio (MSR, USA)
• P. Frossard (EPFL, Switzerland)
• J. Liang, I. Bajic (SFU, Canada)
• V. Stankovic (U of Strathclyde, UK)
• X. Wu (McMaster U, Canada)
• P. Le Callet (U of Nantes, France)
• X. Liu (HIT, China)
• W. Hu, J. Liu, Z. Guo (Peking U., China)
• L. Fang (Tsinghua, HK)
• C.-W. Lin (National Tsing Hua University, Taiwan)
NII Overview

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

- Offers graduate courses & degrees through The Graduate University for Advanced Studies (Sokendai).
- 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.
Introduction to APSIPA and APSIPA DL

**APSIPA Mission**: To promote broad spectrum of research and education activities in signal and information processing in Asia Pacific

**APSIPA Conferences**: APSIPA Annual Summit and Conference

**APSIPA Publications**: Transactions on Signal and Information Processing in partnership with Cambridge Journals since 2012; APSIPA Newsletters

**APSIPA Social Network**: To link members together and to disseminate valuable information more effectively

**APSIPA Distinguished Lectures**: An APSIPA educational initiative to reach out to the community
Outline

• Graph Signal Processing
  • Graph spectrum, GFT
• PWS Image Coding using GFT
• Prediction Residual Coding using GGFT
• Image Denoising using Graph Laplacian Regularizer
• Soft Decoding of JPEG Images w/ LERaG
• GSP for 3D Imaging
• Summary & Ongoing Work
Outline

• Graph Signal Processing
  • Graph spectrum, GFT
• PWS Image Coding using GFT
• Prediction Residual Coding using GGFT
• Image Denoising using Graph Laplacian Regularizer
• Soft Decoding of JPEG Images w/ LERaG
• GSP for 3D Imaging
• Summary & Ongoing 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 is set of outer-product of 1D DCT basis in x- and y-dimension.

\[ a = \Phi x \]

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*.

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.

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 Fourier Transform (GFT)

Graph Laplacian:

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

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

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

\[ D_{i,i} = \sum_j A_{i,j} \]

\[ L = D - A \]

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

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

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

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

*https://en.wikipedia.org/wiki/Second_derivative*
Graph Spectrum from GFT

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

  \[ Lu_i = \lambda_i u_i \]

1. Edge weights affect shapes of eigenvectors.
2. Eigenvalues ($\geq 0$) as *graph frequencies*.
   - Constant eigenvector is DC.
   - # zero-crossings increases as $\lambda$ increases.
- GFT defaults to **DCT** for un-weighted connected line.
- GFT defaults to **DFT** for un-weighted connected circle.
Variants of Graph Laplacians

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

  \[ Lu_i = \lambda_i u_i \]

  eigenvalue

  eigenvector

- Other definitions of graph Laplacians:
  - **Normalized** graph Laplacian:
    \[ L_n = D^{-1/2}LD^{-1/2} = I - D^{-1/2}AD^{-1/2} \]
  - **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.

GSP: SP framework that unifies concepts from multiple fields.

- Partial Differential Eq’ns
- Machine Learning
- Combinatorial Graph Theory
- Spectral Graph Theory
- Computer Graphics
- Computer Vision

Graph Signal Processing* (GSP)

- Laplace equation
- Laplace-Beltrami operator
- Graphical model, manifold learning, classifier learning
- Max cut, graph transformation
- Spectral clustering
- Eigen-analysis of graph Laplacian, adjacency matrices

DSP
Outline

• Graph Signal Processing
  • Graph spectrum, GFT

• PWS Image Coding using GFT

• Prediction Residual Coding using GGFT

• Image Denoising using Graph Laplacian Regularizer

• Soft Decoding of JPEG Images w/ LERaG

• GSP for 3D Imaging

• Summary & Ongoing Work
**PWS Image Compression using GFT**

- DCT are **fixed** basis. Can we do better?
- **Idea:** use **adaptive** GFT to improve sparsity [3].
  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.
  4. Transmit bits (**contour**) to identify chosen GFT to decoder (**overhead of GFT**).

\[ \alpha = \Psi x \]


## Transform Comparison

<table>
<thead>
<tr>
<th>Transform Representation</th>
<th>Transform Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karhunen-Loeve Transform (KLT)</td>
<td>“Sparsest” signal representation given available statistical model</td>
</tr>
<tr>
<td>Discrete Cosine Transform (DCT)</td>
<td><em>non-sparse</em> signal representation across sharp boundaries</td>
</tr>
<tr>
<td>Graph Fourier Transform (GFT)</td>
<td>minimizes the total rate of signal’s transform representation &amp; transform description</td>
</tr>
</tbody>
</table>

MR-GFT: Definition of the Search Space for Graph Fourier Transforms

- In general, weights could be any number in $[0,1]$
- To limit the description cost $R_T$
  - Restrict weights to a small discrete set $\mathcal{C} = \{1, 0, c\}$

$\min_W R_\alpha(x, W) + R_T(W)$

- "1": strong correlation in smooth regions
- "0": zero correlation in sharp boundaries
- "c": weak correlation in slowly-varying parts
MR-GFT: Derivation of Optimal Edge Weights for Weak Correlation

- Assume a 1D 1st-order autoregressive (AR) process \( x = [x_1, \ldots, x_N]^T \) where,

\[
x_k = \begin{cases} 
\eta, & k = 1 \\
 1 < k \leq N, & [k - 1, k] \in S \\
 1 < k \leq N, & [k - 1, k] \in \mathcal{P}
\end{cases}
\]

\( x_k = x_{k-1} + e_k \)

- Assuming the only weak correlation exists between \( x_{k-1} \) and \( x_k \)

\[
\begin{align*}
x_1 &= \eta \\
x_2 - x_1 &= e_2 \\
&\vdots \\
x_k - x_{k-1} &= g + e_k \\
&\vdots \\
x_N - x_{N-1} &= e_N
\end{align*}
\]

mean vector \( \mu = [0 \cdots 0 m_g \cdots m_g]^T \)

k-th

\[
F = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 & 0 \\
0 & \ddots & \ddots & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & \ddots & \ddots & 0 \\
0 & 0 & 0 & 0 & -1 & 1
\end{bmatrix}, \quad b = \begin{bmatrix}
0 \\
e_2 \\
\vdots \\
e_k \\
e_N \\
0
\end{bmatrix}, \quad \eta
\]

\[ Fx = b, \]

\[ x = F^{-1}b \]
MR-GFT: Derivation of Optimal Edge Weights for Weak Correlation (cont’d)

- **Covariance matrix**
  \[
  \mathbf{C} = \mathbb{E}[(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T] = \mathbb{E}[\mathbf{xx}^T] - \mu \mu^T = \mathbb{E}[\mathbf{F}^{-1}\mathbf{bb}^T(\mathbf{F}^T)^{-1}] - \mu \mu^T = \mathbf{F}^{-1}\mathbb{E}[\mathbf{bb}^T](\mathbf{F}^T)^{-1} - \mu \mu^T
  \]

- **Precision matrix** (tri-diagonal)
  \[
  \mathbf{Q} = \mathbf{C}^{-1} = \begin{bmatrix}
    1 + \frac{1}{\sigma_1^2} & -1 & & \\
    -1 & 2 & -1 & \\
    & \ddots & \ddots & \\
    & & -1 & 2
  \end{bmatrix}
  \]
  \[
  c = W_{k-1,k} = \frac{1}{\sigma_g^2 + 1}
  \]

- **Graph Laplacian matrix**
  \[
  \mathbf{L} \approx \begin{bmatrix}
    1 & & & & \\
    1 & 1 & \frac{1}{\sigma_g^2 + 1} & & \\
    & \ddots & \ddots & \ddots & \\
    & & \ddots & \ddots & \frac{1}{\sigma_g^2 + 1} \\
    & & & 1 & 1
  \end{bmatrix}
  \]
MR-GFT: Adaptive Selection of Graph Fourier Transforms
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

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

HR-DCT

HR-SGFT

MR-GFT
Mode Selection

red: WGFT
blue: UGFT
Edge Coding for PWS Image Compression

- **Arithmetic Edge Coding** [1,2]:
  - Coding of sequence of *between-pixel edges*, or chain code with symbols {l, s, r}.
  - Design a *variable-length context tree (VCT)* to compute symbol probabilities for arithmetic coding.

\[ P(x_i | x_1^{i-1}) = P(x_i | w) \]

---


Outline

- Graph Signal Processing
  - Graph spectrum, GFT
- PWS Image Coding using GFT
- Prediction Residual Coding using GGFT
- Image Denoising using Graph Laplacian Regularizer
- Soft Decoding of JPEG Images w/ LERaG
- GSP for 3D Imaging
- Summary & Ongoing Work
Motivation

- **Intra-prediction**

  ![Intra-prediction Diagram](image)

  - Boundary pixel (predictor)
  - Predicted pixels $x_i$

  $x_i - x_0$: prediction residuals

- Discontinuities at block boundaries
  - intra-prediction will not be chosen or bad prediction

---

Optimal 1D Intra prediction

Assume a 1D 1st-order autoregressive (AR) process

\[ x_n = x_{n-1} + \hat{\mu}_i(\mu_n) + \hat{g}_i(\mu_n) \]

- **Optimal prediction** in terms of resulting zero-mean prediction residual
- Default to conventional intra-prediction when \( \hat{\mu}_a = \hat{\mu}_b = 0 \), i.e.,

\[ [x_0, \ldots, x_0]^T \]
Generalized Graph Fourier Transform

• The precision matrix of the prediction residual

\[
\begin{bmatrix}
\alpha_a & -1 & 2 & -1 \\
-1 & 1 + \alpha_b & -\alpha_b & -\alpha_b \\
-\alpha_b & \alpha_b + 1 & -1 \\
1 & -1 & 2 & -1 \\
-1 & 1 + \alpha_b & -\alpha_b & -\alpha_b \\
-\alpha_b & \alpha_b + 1 & -1
\end{bmatrix}
= \begin{bmatrix}
\alpha_a & 0 & \cdot & \cdot & \cdot \\
-1 & 1 + \alpha_b & -\alpha_b & -\alpha_b & 0 \\
-\alpha_b & \alpha_b + 1 & -1 & \cdot & \cdot \\
1 & -1 & 2 & -1 & \cdot \\
-1 & 1 + \alpha_b & -\alpha_b & -\alpha_b & 0 \\
-\alpha_b & \alpha_b + 1 & -1 & \cdot & \cdot \\
\end{bmatrix}
+ \begin{bmatrix}
\alpha_a \\
0 \\
\cdot \\
\cdot \\
\cdot \\
\cdot \\
0
\end{bmatrix}
\]

Generalized Laplacian
Combinatorial Laplacian
Degree matrix for boundary vertices

• Default to the DCT if \( \alpha_a = 0 \) and \( \alpha_b = 1 \)

• Default to the ADST [1] if \( \alpha_a = 1 \) and \( \alpha_b = 1 \)

\[
\alpha_a = \frac{\sigma_g^2}{\sigma_o^2} \quad \alpha_b = \frac{\sigma_g^2}{\sigma_b^2}
\]

inaccuracy of intra-prediction
discontinuities within signal
Variance of approx. error

Experimental Results

- Test images: PWS images and natural images
- Compare proposed intra-prediction ($p_{\text{Intra}}$) + GGFT against:
  - edge-aware intra-prediction ($e_{\text{Intra}}$) + DCT
  - $e_{\text{Intra}}$ + ADST
  - $e_{\text{Intra}}$ + GFT

![Graphs showing PSNR vs. Bit per pixel for Teddy and Tsukuba images.](image_url)
Spectral Folding & Critical Sampling

- **Spectral Folding:**
  - (Sub)sampling a bandlimited signal at freq. $f_s$ → freq. content replication at $f_s$.

- **Nyquist Sampling Theorem:**
  - To avoid aliasing, sample at 2x max. freq. of bandlimited signal.

- **Multirate Wavelet Filterbank:**
  - System of “perfect reconstruction” bandpass filters

A Multirate two-channel system

- Analysis filter
- Synthesis filter

BJTU 11/25/2017
Bipartite Graph Approximation

**Problem:** GraphBior [1,2] (critically sampled, perfect reconst. wavelet) for *bipartite graph* only!

**Idea** [3]:
- Successively find *bipartite graph approximation*.
- Criteria for graph approx [1]:
  \[
  \min_{L^b} D_{KL}(L \| L^b) - \gamma \text{ rank}(L^b_{1,2})
  \]
- Preserve graph structure, minimize eigenvalue=1.


Outline

• Graph Signal Processing
  • Graph spectrum, GFT
• PWS Image Coding using GFT
• Prediction Residual Coding using GGFT
• Image Denoising using Graph Laplacian Regularizer
• Soft Decoding of JPEG Images w/ LERaG
• GSP for 3D Imaging
• Summary & Ongoing Work
Graph Laplacian Regularizer

- $x^T L x$ (graph Laplacian quadratic form) [1]) is one variation measure → graph-signal smoothness prior.

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

- Signal Denoising:

observation $\rightarrow$ $y = x + v$ $\leftarrow$ noise

- MAP formulation:

$$\min_x \|y - x\|_2^2 + \mu x^T L x$$

signal smooth in nodal domain

desired signal

signal contains mostly low graph freq.

smoothness prior

fidelity term

Graph Laplacian Regularizer for Denoising

1. Choose graph:
   - Connect neighborhood graph.
   - Assign edge weight:
     \[ w_{i,j} = \exp \left( \frac{-\|x_i - x_j\|^2}{\sigma_1^2} \right) \exp \left( \frac{-\|l_i - l_j\|^2}{\sigma_2^2} \right) \]

2. Solve obj. in closed form:
   \[ \min_x \|y - x\|^2 + \mu x^T L x \]
   - Iterate until convergence.

Analysis of Graph Laplacian Regularizer

• Show $S_G(u) = u^T Lu$ converges to continuous functional $S_\Omega$, analysis of $S_\Omega$ explains how $u^T Lu$ penalizes candidates:

$$\text{prior}(x) = x^T L x \rightarrow S_\Omega(x) = \int_\Omega \nabla x^T G^{-1} \nabla x \left(\sqrt{\det G}\right)^{2\gamma-1} ds$$

• Derive optimal $S_G(u) = u^T Lu$ for denoising: graph is discriminant for small noise, robust when very noisy.

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

Denoising Experiments (natural images)

- Subjective comparisons ($\sigma = 40$)

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 ($\sigma_i = 30$)

Outline

• Graph Signal Processing
  • Graph spectrum, GFT
• PWS Image Coding using GFT
• Prediction Residual Coding using GGFT
• Image Denoising using Graph Laplacian Regularizer
• Soft Decoding of JPEG Images w/ LERaG
• GSP for 3D Imaging
• Summary & Ongoing Work
Soft Decoding of JPEG Images

**Setting:** JPEG compresses natural images:

1. Divide image into 8x8 blocks, DCT.
2. Perform DCT transform per block and quantize:
   \[ q_i = \text{round}(Y_i / Q_i), \quad Y = Ty \]
   - **DCT**: DCT Coefficients
   - 8x8 pixel block
   - **Quantization parameter**
   - **DCT**
3. Quantized DCT coeff entropy coded.

**Decoder:** uncertainty in signal reconstruction:

\[ q_i Q_i \leq Y_i \leq (q_i + 1) Q_i, i = 1, 2, \ldots, 64. \]


Graph Laplacian Regularizer for Denoising

1. Choose graph:
   - Connect neighborhood graph.
   - Assign edge weight:
     \[ w_{i,j} = \exp \left( -\frac{\|x_i - x_j\|^2}{\sigma_1^2} \right) \exp \left( -\frac{\|l_i - l_j\|^2}{\sigma_2^2} \right) \]

2. Solve obj. in closed form:
   \[
   \min_x \|y - x\|^2_2 + \mu x^T L x
   \]
   - Iterate until convergence.

Comments:
1. \( L \) is NOT normalized.
2. Why works well for PWS signals?

Spectral Clustering

• **Normalized Cut** [1]:

\[
\min_{A,B} Ncut(A, B) := \left( \text{cut}(A, B) \left( \frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)} \right) \right)
\]

\[
\text{cut}(A, B) = \sum_{i \in A, j \in B} W_{i,j} \quad \text{vol}(A) = \sum_{i \in A} D_{i,i}
\]

• Problem is **NP-hard**, so:

1. Rewrite as:

\[
\min_f \quad f^T L f \quad \text{s.t. } f_i = \begin{cases} 
1 & \text{if } i \in A \\
\frac{1}{\text{vol}(A)} - 1 & \text{if } i \in B \\
\frac{1}{\text{vol}(B)} & \text{if } i \in B
\end{cases}
\]

2. Relax to:

\[
\min_f \quad f^T L f \quad \text{s.t. } f^T D 1 = 0
\]

Eigenvectors of Normalized graph Laplacian

• Define:
\[ v := D^{1/2}f \quad v_1 := D^{1/2}1 \]

• Problem rewritten as:
\[ v^* = \arg \min_v \frac{v^T L_n v}{v^T v} \quad \text{s.t. } v^T v_1 = 0 \]

• \( v_1 \) minimizes obj \( \rightarrow \) Sol'n is 2nd eigenvector of \( L_n \).
• If \( f^* \) optimal to norm. cut, \( v^* \) is PWS \( \rightarrow \) well rep. PWS signals!
• \( f^* \) optimal when nodes easy to cluster:
  • Easy-to-cluster graph has small Fiedler number.

• Disadvantage:
  • \( v_1 \) not constant vector (DC) \( \rightarrow \) cannot well rep. smooth patch.
Left E-vector random walk graph Laplacian (LERaG)

- **Disadvantage:**
  - $L_{rw}$ is asymmetric → no orthogonal e-vectors w/ real e-values.
- So, **left Eigenvector Random Walk Graph Laplacian (LERaG)** [1]:

\[
x^T L_r^T L_r x = (x^T D^{1/2} L_n) D^{-1} (L_n D^{1/2} x)
\]

\[
\gamma = L_n D^{1/2} x
\]

\[
x^T L_r^T L_r x = \gamma^T D^{-1} \gamma
\]

\[
\frac{\gamma^T \gamma}{d_{\text{max}}} \leq \gamma^T D^{-1} \gamma \leq \frac{\gamma^T \gamma}{d_{\text{min}}}
\]

projection of signal $x$ to $D^{1/2}$, then $L_n$

Comparison of Graph-signal Smoothness Priors

• Different graph Laplacian matrices
  • Combinatorial graph Laplacian: \( L = D - W \)
  • Symmetrically normalized graph Laplacian: \( L_n = D^{-1/2} LD^{-1/2} \)
  • Random walk graph Laplacian: \( L_r = D^{-1} L \)
  • Doubly stochastic graph Laplacian [1]: \( L_d = I - C^{-1/2} WC^{-1/2} \)

<table>
<thead>
<tr>
<th>Graph Laplacian</th>
<th>Symmetric</th>
<th>Normalized</th>
<th>DC e-vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combinatorial</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Symmetrically Normalized</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Random Walk</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Doubly Stochastic [1]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

LERaG for Soft Decoding of JPEG Images

- **Problem**: reconstruct image given indexed quant. bin in 8x8 DCT.

- **Procedure**:
  1. Initialize per-block MMSE sol’n via Laplacian prior.
  2. Solve per-patch signal restoration problem w/ 2 priors:
     1. Sparsity prior
     2. Graph-signal smoothness prior

Soft Decoding Algorithm w/ Prior Mixture

• **Objective:**

\[
\arg\min_{\{x, \alpha\}} \|x - \Phi \alpha\|_2^2 + \lambda_1 \|\alpha\|_0 + \lambda_2 x^T (d_{\text{min}}^{-1}) L D^{-1} L x,
\]

s.t. \( qQ \leq T M x < (q + 1) Q \)

• **Optimization:**

1. Laplacian prior provides an initial estimation;
2. Fix \( x \) and solve for \( \alpha \);
3. Fix \( \alpha \) and solve for \( x \).
Evolution of 2\textsuperscript{nd} Eigenvector

- 2\textsuperscript{nd} Eigenvector becomes more PWS:

  - PWS means:
    1. better pixel clusters,
    2. smaller Fidler number (2\textsuperscript{nd} eigenvalue),
    3. Smaller smoothness penalty term.
Experimental Setup

- **Compared methods**
  - **BM3D**: well-known denoising algorithm
  - **KSVD**: with a large enough over-complete dictionary (100x4000); our method uses a much smaller one (100x400).
  - **ANCE**: non-local self similarity [Zhang et al. TIP14]
  - **DicTV**: Sparsity + TV [Chang et al, TSP15]
  - **SSRQC**: Low rank + Quantization constraint [Zhao et al. TCSVT16]
## PSNR / SSIM Comparison

**Quality Comparison with Respect to PSNR (in dB) and SSIM at QF = 40**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>SSIM</td>
<td>PSNR</td>
<td>SSIM</td>
<td>PSNR</td>
<td>SSIM</td>
<td>PSNR</td>
</tr>
<tr>
<td>Butterfly</td>
<td>29.97</td>
<td>0.9244</td>
<td>31.35</td>
<td>0.9555</td>
<td>31.57</td>
<td>0.9519</td>
<td>31.38</td>
</tr>
<tr>
<td>Leaves</td>
<td>30.67</td>
<td>0.9438</td>
<td>32.55</td>
<td>0.9749</td>
<td>33.04</td>
<td>0.9735</td>
<td>32.74</td>
</tr>
<tr>
<td>Hat</td>
<td>32.78</td>
<td>0.9022</td>
<td>33.89</td>
<td>0.9221</td>
<td>33.62</td>
<td>0.9149</td>
<td>33.69</td>
</tr>
<tr>
<td>Boat</td>
<td>33.42</td>
<td>0.9195</td>
<td>34.77</td>
<td>0.9406</td>
<td>34.28</td>
<td>0.9301</td>
<td>34.64</td>
</tr>
<tr>
<td>Bike</td>
<td>28.98</td>
<td>0.9131</td>
<td>29.96</td>
<td>0.9356</td>
<td>30.19</td>
<td>0.9323</td>
<td>30.31</td>
</tr>
<tr>
<td>House</td>
<td>35.07</td>
<td>0.8981</td>
<td>36.09</td>
<td>0.9013</td>
<td>36.05</td>
<td>0.9055</td>
<td>36.12</td>
</tr>
<tr>
<td>Flower</td>
<td>31.62</td>
<td>0.9112</td>
<td>32.81</td>
<td>0.9357</td>
<td>32.63</td>
<td>0.9271</td>
<td>32.67</td>
</tr>
<tr>
<td>Parrot</td>
<td>34.03</td>
<td>0.9291</td>
<td>34.92</td>
<td>0.9397</td>
<td>34.91</td>
<td>0.9371</td>
<td>35.02</td>
</tr>
<tr>
<td>Pepper512</td>
<td>34.21</td>
<td>0.8711</td>
<td>34.94</td>
<td>0.8767</td>
<td>34.89</td>
<td>0.8784</td>
<td>34.99</td>
</tr>
<tr>
<td>Fishboat512</td>
<td>32.76</td>
<td>0.8763</td>
<td>33.61</td>
<td>0.8868</td>
<td>33.36</td>
<td>0.8809</td>
<td>33.60</td>
</tr>
<tr>
<td>Lena512</td>
<td>35.12</td>
<td>0.9089</td>
<td>36.03</td>
<td>0.9171</td>
<td>35.82</td>
<td>0.9146</td>
<td>36.04</td>
</tr>
<tr>
<td>Airplane512</td>
<td>33.36</td>
<td>0.9253</td>
<td>34.38</td>
<td>0.9361</td>
<td>34.36</td>
<td>0.9341</td>
<td>34.53</td>
</tr>
<tr>
<td>Bike512</td>
<td>29.43</td>
<td>0.9069</td>
<td>30.47</td>
<td>0.9299</td>
<td>30.66</td>
<td>0.9258</td>
<td>30.71</td>
</tr>
<tr>
<td>Statue512</td>
<td>32.78</td>
<td>0.9067</td>
<td>33.61</td>
<td>0.9188</td>
<td>33.55</td>
<td>0.9149</td>
<td>33.55</td>
</tr>
<tr>
<td>Average</td>
<td>32.44</td>
<td>0.9097</td>
<td>33.52</td>
<td>0.9264</td>
<td>33.50</td>
<td>0.9229</td>
<td>33.57</td>
</tr>
</tbody>
</table>

BJTU 11/25/2017
Subjective Quality Evaluation

(a) BM3D (23.91, 0.8266)  
(b) KSVD (24.55, 0.8549)  
(c) ANCE (24.34, 0.8532)  
(d) DicTV (23.42, 0.8176)  
(e) SSRQE (25.31, 0.8864)  
(f) Proposed (25.82, 0.8861)
Subjective Quality Evaluation

(a) BM3D (23.78, 0.8408)
(b) KSVD (24.39, 0.8684)
(c) ANCE (24.18, 0.8551)
(d) DicTV (23.27, 0.8245)
(e) SSRQC (23.01, 0.8601)
(f) Proposed (25.57, 0.8979)
Other Comparisons

- Computation complexity:

<table>
<thead>
<tr>
<th>TIME</th>
<th>BM3D</th>
<th>KSVD</th>
<th>ANCE</th>
<th>DicTV</th>
<th>SSRQC</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>373.35</td>
<td>209.71</td>
<td>307.43</td>
<td>39.53</td>
<td>70.32</td>
<td>143.73</td>
</tr>
</tbody>
</table>

- Comparisons w/ other graph regularizers:

<table>
<thead>
<tr>
<th>Images</th>
<th>Combinatorial</th>
<th>Normalized</th>
<th>Doubly Stochastic</th>
<th>LERaG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Butterfly</td>
<td>25.42</td>
<td>24.70</td>
<td>25.15</td>
<td>25.57</td>
</tr>
<tr>
<td>Leaves</td>
<td>24.99</td>
<td>24.54</td>
<td>24.84</td>
<td>25.17</td>
</tr>
<tr>
<td>Hat</td>
<td>27.53</td>
<td>27.42</td>
<td>27.43</td>
<td>27.56</td>
</tr>
<tr>
<td>Bike</td>
<td>23.12</td>
<td>23.01</td>
<td>23.09</td>
<td>23.17</td>
</tr>
<tr>
<td>House</td>
<td>29.87</td>
<td>29.83</td>
<td>29.86</td>
<td>29.89</td>
</tr>
<tr>
<td>Flower</td>
<td>25.84</td>
<td>25.78</td>
<td>25.82</td>
<td>25.87</td>
</tr>
<tr>
<td>Parrot</td>
<td>27.97</td>
<td>27.95</td>
<td>27.97</td>
<td>28.02</td>
</tr>
<tr>
<td>Average</td>
<td>26.46</td>
<td>26.27</td>
<td>26.39</td>
<td>26.53</td>
</tr>
</tbody>
</table>
Outline

• Graph Signal Processing
  • Graph spectrum, GFT
• PWS Image Coding using GFT
• Prediction Residual Coding using GGFT
• Image Denoising using Graph Laplacian Regularizer
• Soft Decoding of JPEG Images w/ LERaG
• GSP for 3D Imaging
• Summary & Ongoing Work
Graph-Signal Sampling / Encoding for 3D Point Cloud

- **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**.

MIT dataset*
Graph-Signal Sampling / Encoding for 3D Point Cloud

- **Experimental Results:**

![Graph-Signal Sampling / Encoding for 3D Point Cloud](image)

(a) Dataset1  
(b) Dataset2

Pre-Demosiac Light Field Image Compression Using Graph Lifting Transform

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

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

---

**Raw Lenselet Image**  
Demosaicing  
Sub-aperture images  
Demosaiced Image  
Calibrated Color Image  
Calibration (Scaling, Transition, Rotation)  
× 3  
× \( \approx 1.5 \)  
Re-arranged in 4D space  
Sub-aperture space  
Calibrated Image  
Re-arranged on Calibrated Image  
Raw Lenselet Image  
Sub-aperture Images  
Image Coding  
Demosaiced Image  
Calibrated Color Image  
graph-based lifting transform
Pre-Demosiak 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

![Graphs showing experimental results](image)

Outline

• Graph Signal Processing
  • Graph spectrum, GFT
• PWS Image Coding using GFT
• Prediction Residual Coding using GGFT
• Image Denoising using Graph Laplacian Regularizer
• Soft Decoding of JPEG Images w/ LERaG
• GSP for 3D Imaging
• Summary & Ongoing Work
Summary

- **Graph Signal Processing (GSP)**
  - Spectral analysis tools to process signals on graphs.
- **PWS Image Compression**
  - Graph Fourier Transform
  - Generalized GFT
  - Arithmetic Edge Coding
- **Graph-signal Smoothness for Inverse Problems**
  - Image denoising w/ graph Laplacian regularizer
  - New regularizer LERaG soft decoding of JPEG Images
- **GSP for 3D Imaging**
  - 3D point cloud compression, light field image compression
Other GSP Works: Semi-Supervised Graph Classifier Learning

- **Binary Classifier**: given feature vector \( x_i \) of dimension \( K \), compute \( f(x_i) \in \{0,1\} \).
- **Classifier Learning**: given partial / noisy labels \((x_i, y_i)\), train classifier \( f(x_i) \).

**GSP Approach** [1]:

1. Construct *similarity graph* with +/- edges.
2. Pose MAP graph-signal restoration problem.
3. Perturb graph Laplacian to ensure PSD.
4. Solve num. stable MAP as sparse lin. system.

---


Other GSP Works

- Coding of spectral image [1], 3D point cloud w/ GFT.
- Coding of graph data w/ graph wavelets.
- Political leaning estimation [2].
- Wireless signal / power estimation [3].


Q&A

- Email: cheung@nii.ac.jp
- Homepage: http://research.nii.ac.jp/~cheung/