Continuous Domain Analysis of Graph Laplacian Regularization for Image Denoising

Presenter: Gene Cheung
National Institute of Informatics

Outline

• Introduction
• Convergence of the Graph Laplacian Regularizer
• Justification of the Graph Laplacian Regularizer
• Formulation and Algorithm
• Experimental Results
• Towards the Optimal Graph Laplacian Regularizer
• Conclusion

Lena, $\sigma = 30$
Outline

- Introduction
- Convergence of the Graph Laplacian Regularizer
- Justification of the Graph Laplacian Regularizer
- Formulation and Algorithm
- Experimental Results
- Towards the Optimal Graph Laplacian Regularizer
- Conclusion

"Lena, σ = 30"
Motivation (I)

- Image denoising—a basic restoration problem:

\[ y = x + e \]

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

\[
\min_x \|y - x\|_2^2 + \lambda \text{ prior}(x)
\]

- Graph Laplacian regularizer: should be small for target patch \( x \)

\[
S_G(x) = x^T L x \quad L = D - A
\]

- Many works use Gaussian kernel to compute graph weights [2]:

\[
w_{ij} = \exp\left(\frac{\text{dist}(i, j)^2}{\sigma^2}\right)
\]

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

Motivation (II)

• However...
  a. Why is $S_G(x) = x^T L x$ a good prior?
  b. Why using Gaussian kernel for edge weights?
  c. How to design a discriminant $x^T L x$ for restoration?

• We answer these by viewing
  • discrete graph as samples of high-dimensional manifold.
Our Contributions

1. Using Gaussian kernel to compute graph weights, $S_G(x) = x^T L x$ converges to a continuous functional $S_\Omega$, which can be interpreted as regularizer in continuous domain.

2. Analysis of functional $S_\Omega$ provides understanding of what signals are being discriminated and to what extent, on a point-by-point basis in the continuous domain.

3. We design a discriminant $S_\Omega$ for regularization in continuous domain, then obtain the graph Laplacian regularizer $S_G$. The corresponding $S_G$ design discriminant $S_\Omega$ obtain
Outline

• Introduction
• Convergence of the Graph Laplacian Regularizer
• Justification of the Graph Laplacian Regularizer
• Formulation and Algorithm
• Experimental Results
• Towards the Optimal Graph Laplacian Regularizer
• Conclusion

Lena, $\sigma = 30$
Road Map

**Continuous Domain**
- Choose the continuous feature functions $\{f_n\}_{n=1}^N$
- Get metric space $\mathbf{G} \in \mathbb{R}^{2 \times 2}$ on point-by-point basis
- Obtain continuous functional $S_\Omega(h)$

**Discrete Domain**
- Sample $\{f_n\}_{n=1}^N$ to obtain the discrete $\{f_n^D\}_{n=1}^N$
- Compute the weights and Laplacian $\mathbf{L} \in \mathbb{R}^{M \times M}$
- Graph Laplacian Regularizer $S_{G}(\mathbf{h}^D)$

- Different $\{f_n\}_{n=1}^N$ leads to different regularization behavior!
Graph Construction (I)

• First, define:
  • 2D domain \( \Omega \subset R^2 \) — the shape of an image
  • \( \Gamma = \{ s_i = [x_i, y_i]^T | s_i \in \Omega, 1 \leq i \leq M \} \) — a set of \( M \) random samples uniformly distributed on \( \Omega \), construed as pixel locations

• (Freely) choose \( N \) continuous functions

\[
f_n(x, y) : \Omega \to R, \ 1 \leq n \leq N\]

called feature functions, can be
  • intensity for gray-scale image \((N = 1)\)
  • \( R, G, B \) channels for color image \((N = 3)\)
Graph Construction (II)

- Sampling $f_n$ at positions in $\Gamma$ gives $N$ discretized feature functions
  \[ f^D_n = [f_n(x_1, y_1) \ f_n(x_2, y_2) \ldots \ f_n(x_M, y_M)]^T \]

- For each pixel location $s_i \in \Gamma$, define a length $N + 2$ vector
  \[ v_i = [x_i \ y_i \ \beta f^1_D(i) \ \beta f^2_D(i) \ldots \ \beta f^N_D(i)]^T \]
  \( \beta \) is a tunable constant

- Build a graph $G$ with $M$ vertices, each pixel location $s_i \in \Gamma$ have a vertex $V_i$
Graph Construction (III)

- **Weight between vertices** $V_i$ and $V_j$
  
  degree before normalization
  
  $$\rho_i = \sum_{j=1}^{m} \psi(d_{ij})$$

  normalization factor $\gamma$

  clipped Gaussian kernel
  
  $$\psi(d) = \begin{cases} \exp\left(-\frac{d^2}{2\sigma^2}\right) & |d| \leq r, \\ 0 & \text{otherwise} \end{cases}$$

  where $r = \varepsilon C_r$ and $C_r$ is a constant

  distance
  
  $$d_{ij}^2 = \|v_i - v_j\|_2^2$$

  $$= \|s_i - s_j\|_2^2 + \beta^2 \sum_{n=1}^{N} (f_n^D(i) - f_n^D(j))^2$$

- **Roadmap**

  - **Features** $\{f_n\}_{n=1}^{N}$
  - **Samples** $\{f_n^D\}_{n=1}^{N}$
  - **Matrix** $G \in \mathbb{R}^{2 \times 2}$
  - **Graph weights, and** $L \in \mathbb{R}^{M \times M}$
  - **Functional** $S_\Omega(h)$
  - **Regularizer** $S_G(h^D)$

  converge

- **$G$ is an $r$-neighborhood graph**, i.e., no edge connecting two vertices with distance greater than $r$
Graph Construction (IV)

- Our graph $G$ is very general
  - *e.g.*, choose a small $\beta$ with proper $r$, obtain the 2D grid graph

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

- $h(x, y) : \Omega \rightarrow R$ is a continuous candidate function
  - $h^D = [h(x_1, y_1), h(x_2, y_2), \ldots, h(x_M, y_M)]^T$ — samples of $h(x, y)$
  - $S_G(h^D) = (h^D)^T L h^D$ — graph Laplacian regularizer, a functional on $R^M$
Convergence of the Graph Laplacian Regularizer (I)

- The continuous counterpart of \( S_G \) is a functional \( S_\Omega \) on domain \( \Omega \)

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

\( \nabla h = [\partial_x h \, \partial_y h]^T \) is the gradient of \( h \)

- \( G \) is a 2-by-2 matrix:

\[
G = I + \beta^2 \begin{bmatrix}
\sum_{n=1}^{N} \left( \partial_x f_n \right)^2 & \sum_{n=1}^{N} \partial_x f_n \cdot \partial_y f_n \\
\sum_{n=1}^{N} \partial_x f_n \cdot \partial_y f_n & \sum_{n=1}^{N} \left( \partial_y f_n \right)^2
\end{bmatrix} = I + \beta^2 \sum_{n=1}^{N} \nabla f_n \cdot (\nabla f_n)^T
\]

\( 2 \times 2 \) identity matrix

- \( G \) is computed from \( \{\nabla f_n\}_{n=1}^{N} \) on a point-by-point basis

---

Convergence of the Graph Laplacian Regularizer (II)

- **Theorem:** convergence of $S_G$ to $S_\Omega$

\[
\lim_{M \to \infty} \frac{M^{2\gamma - 1}}{\varepsilon^{4(1-\gamma)}(M-1)} S_G(h^D) \sim S_\Omega(h)
\]

number of samples $M$ increases
neighborhood $r = \varepsilon C_r$ shrinks

“\~” means there exist a constant such that equality holds.

- With results of [4], we proved it by viewing a graph as proxy of an $N + 2$-dimensional Riemannian manifold

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Coordinate on $\Omega$</th>
<th>Coordinate on $(N+2)$-D manifold</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_i$</td>
<td>$s_i = (x_i, y_i)$</td>
<td>$v_i = [x_i \ y_i \ \beta f_1^D(i) \ \beta f_2^D(i) \ \ldots \ \beta f_N^D(i)]^T$</td>
</tr>
</tbody>
</table>

Outline

• Introduction
• Convergence of the Graph Laplacian Regularizer
• Justification of the Graph Laplacian Regularizer
• Formulation and Algorithm
• Experimental Results
• Towards the Optimal Graph Laplacian Regularizer
• Conclusion

\[ \text{Lena, } \sigma = 30 \]
Justification of Graph Laplacian Regularizer (I)

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

\[ G = I + \beta^2 \sum_{n=1}^{N} \nabla f_n : (\nabla f_n)^T \]

\[ S_G(h^D) = (h^D)^T L h^D \]

Roadmap

Features \( \{f_n\}_{n=1}^{N} \) \hspace{1cm} Samples \( \{f^D_n\}_{n=1}^{N} \)

Matrix \( G \in \mathbb{R}^{2 \times 2} \) \hspace{1cm} Graph weights, and \( L \in \mathbb{R}^{M \times M} \)

Functional \( S_{\Omega}(h) \) \hspace{1cm} Regularizer \( S_G(h^D) \)

- \( S_G \) converges to \( S_{\Omega} \), With \( S_{\Omega} \), any new insights we can gain on \( S_G \) ??

- The eigen-space of \( G \) reflects statistics of \( \{\nabla f_n\}_{n=1}^{N} \)
- \( (\nabla h)^T G^{-1} (\nabla h) \) measures length of \( \nabla h \) in a metric space established by \( G \)!
- \( S_{\Omega} \) integrates the gradient norm
Justification of Graph Laplacian Regularizer (II)

- **Metric space defined by** $G$

Ellipses are norm-balls, reflects how concentration of $\{\nabla f_n\}_{n=1}^N$

Green dots are $\{\nabla f_n(x, y)\}_{n=1}^N$

$l$: Eigenvector corresponds to the largest eigenvalue of $G$, goes through the cluster of $\{\nabla f_n\}_{n=1}^N$

\[
S_\Omega(h) = \int_\Omega (\nabla h)^T G^{-1} (\nabla h) \left(\sqrt{\det G}\right)^{2\gamma-1} \, dxdy
\]

\[
G = I + \beta^2 \sum_{n=1}^N \nabla f_n \cdot (\nabla f_n)^T
\]
Justification of Graph Laplacian Regularizer (III)

- The 2D metric space provides a clear picture of what signals are being discriminated and to what extent, on a point-by-point basis in the continuous domain!

\[ \frac{\partial x}{\partial y} \]

\[ \frac{\partial x}{\partial y} \]

(a) is more skewed, or discriminant, than (b)

- (a) is more skewed, or discriminant, than (b)
- In (a), a small distance away from the direction orthogonal to \( l \) brings large metric distance
Justification of Graph Laplacian Regularizer (IV)

• **Lesson**: Select feature functions properly!

• Suppose $A$ is the truth gradient, choose $\{f_n\}_{n=1}^N$ such that
  
  • (i) $l$ goes through $A$; (ii) Ellipses stretched flat along $l$.

(a) A **good** scheme, $\{\nabla f_n\}_{n=1}^N$ are similar to the ground-truth $A$

(b) A **bad** scheme...

• For the case of discrete images, one can seek for similar patches in terms of gradient!
Outline

- Introduction
- Convergence of the Graph Laplacian Regularizer
- Justification of the Graph Laplacian Regularizer
- Formulation and Algorithm
- Experimental Results
- Towards the Optimal Graph Laplacian Regularizer
- Conclusion

Lena, $\sigma = 30$
Problem Formulation and Algorithm Development

- Adopt a patch-based recovery framework to denoise the image
- For a noisy patch \( p_0 \) on the image
  1. Assume a “self-similar-in-gradient” image model, search for \( K - 1 \) patches similar to \( p_0 \) in terms of gradient in pre-filtered image.
  2. Compute graph Laplacian from the similar patches.
  3. Solve the unconstrained quadratic optimization iteratively:
     \[
     q^* = \arg \min_q \| p_0 - q \|_2^2 + \lambda q^T L q
     \]
     to obtain the denoised patch \( q^* \).
- Aggregate denoised patches to form an updated image.
- Denoise the given image iteratively to gradually enhance its quality.
- Our denoising method is named
  Graph-based Denoising using Gradient-based Self-similarity (GDGS)
Outline

- Introduction
- Convergence of the Graph Laplacian Regularizer
- Justification of the Graph Laplacian Regularizer
- Formulation and Algorithm
- Experimental Results
- Towards the Optimal Graph Laplacian Regularizer
- Conclusion

*Lena, $\sigma = 30$*
Experimental Results (I)

- Test images: Lena, Barbara, Boats and Peppers
- i.i.d. Additive White Gaussian Noise (AWGN)
- Non-Local GBT (NLGBT) – an existing graph-based denoising method [5]
- Compared to BF, NLM and NLGBT

Performance comparisons in PSNR (dB)

<table>
<thead>
<tr>
<th>Image</th>
<th>Method</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>Lena</td>
<td>GDGS</td>
<td>33.47</td>
</tr>
<tr>
<td></td>
<td>NLM</td>
<td>30.61</td>
</tr>
<tr>
<td></td>
<td>BF</td>
<td>27.00</td>
</tr>
<tr>
<td></td>
<td>NLGBT</td>
<td>21.52</td>
</tr>
<tr>
<td>Barbara</td>
<td>GDGS</td>
<td>31.71</td>
</tr>
<tr>
<td></td>
<td>NLM</td>
<td>28.33</td>
</tr>
<tr>
<td></td>
<td>BF</td>
<td>25.78</td>
</tr>
<tr>
<td></td>
<td>NLGBT</td>
<td>21.03</td>
</tr>
<tr>
<td>Boats</td>
<td>GDGS</td>
<td>31.59</td>
</tr>
<tr>
<td></td>
<td>NLM</td>
<td>28.55</td>
</tr>
<tr>
<td></td>
<td>BF</td>
<td>26.42</td>
</tr>
<tr>
<td></td>
<td>NLGBT</td>
<td>22.19</td>
</tr>
<tr>
<td>Peppers</td>
<td>GDGS</td>
<td>33.30</td>
</tr>
<tr>
<td></td>
<td>NLM</td>
<td>30.83</td>
</tr>
<tr>
<td></td>
<td>BF</td>
<td>28.96</td>
</tr>
<tr>
<td></td>
<td>NLGBT</td>
<td>21.49</td>
</tr>
</tbody>
</table>

1.4 dB better than NLM!

Experimental Results (II)

- GDGS vs NLGBT

- GDGS vs NLM

Noise standard deviation $\sigma = 25$
Outline

- Introduction
- Convergence of the Graph Laplacian Regularizer
- Justification of the Graph Laplacian Regularizer
- Formulation and Algorithm
- Experimental Results
- Towards the Optimal Graph Laplacian Regularizer
- Conclusion

$Lena, \sigma = 30$
Towards Optimal Graph Laplacian Regularization

- Our latest work [6] derives the optimal metric space $G^\star$, leading to optimal graph Laplacian regularization for denoising.

- Metric space should be discriminant to the extent that estimates of ground-truth gradient are reliable.

$$G^\star = \arg\min_G \int \int_{\Delta} \|G - G_0(g)\|_F^2 \ Pr \left( g \left| \{g_k\}_{k=0}^{K-1} \right. \right) dg$$

$\Delta$—whole gradient domain

ideal metric space given ground truth $g$

Outline

• Introduction
• Convergence of the Graph Laplacian Regularizer
• Justification of the Graph Laplacian Regularizer
• Formulation and Algorithm
• Experimental Results
• Towards the Optimal Graph Laplacian Regularizer
• Conclusion

Lena, $\sigma = 30$
• Image denoising is an ill-posed problem and requires good priors for regularization.

• graph Laplacian regularizer with Gaussian kernel weights converges to a continuous functional.

• Analysis of the continuous functional provides theoretical justification of why and under what conditions the graph Laplacian regularizer can be discriminant.

• Our denoising algorithm with graph Laplacian regularizer and gradient-based similarity out-performs NLM by up to 1.4 dB.

• Our latest work obtains the optimal graph Laplacian, which is discriminant when the estimates are accurate, and robust when the estimates are not.
Thank You!

Contact: Gene Cheung (cheung@nii.ac.jp)
Jiahao Pang (jpang@ust.hk)