Graph signal processing

Concepts, tools and applications

Xiaowen Dong

Department of Engineering Science

UNIVERSITY OF OXFORD

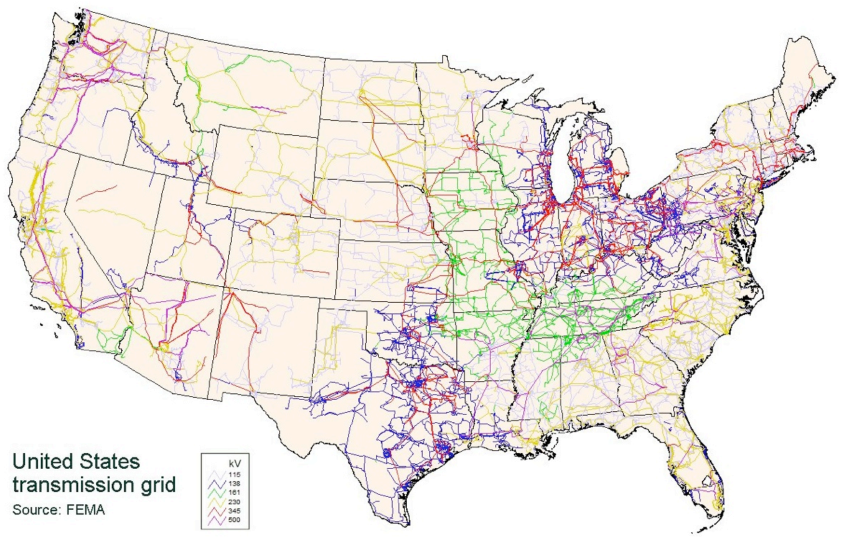
Outline

- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Applications and perspectives

Outline

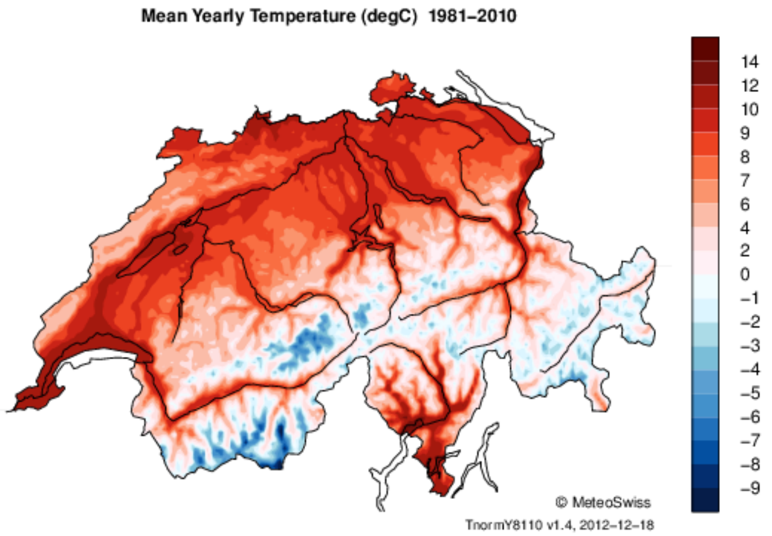
- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Applications and perspectives

Data are often structured

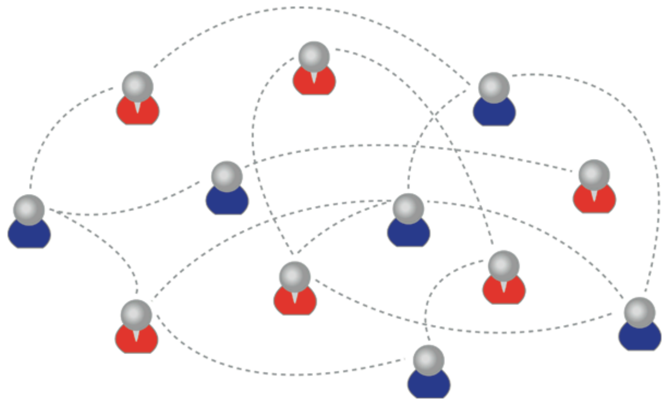


Electrical data

Traffic data

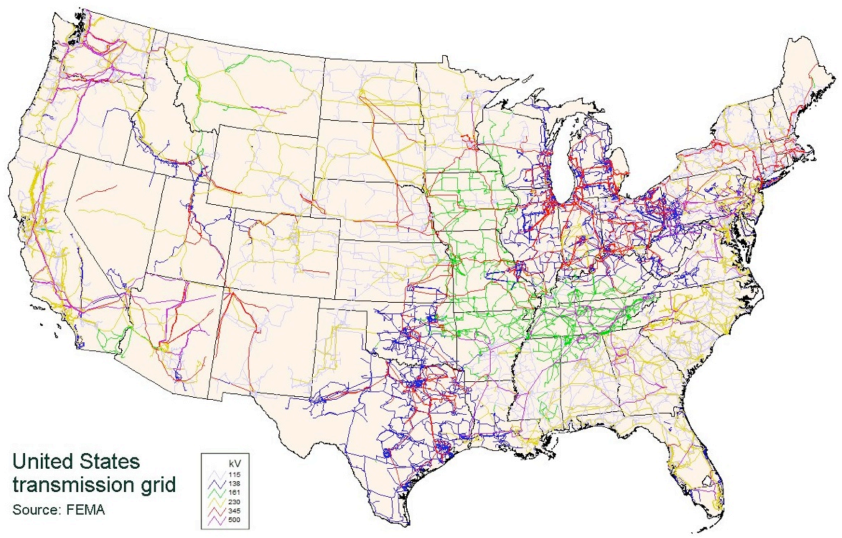


Temperature data



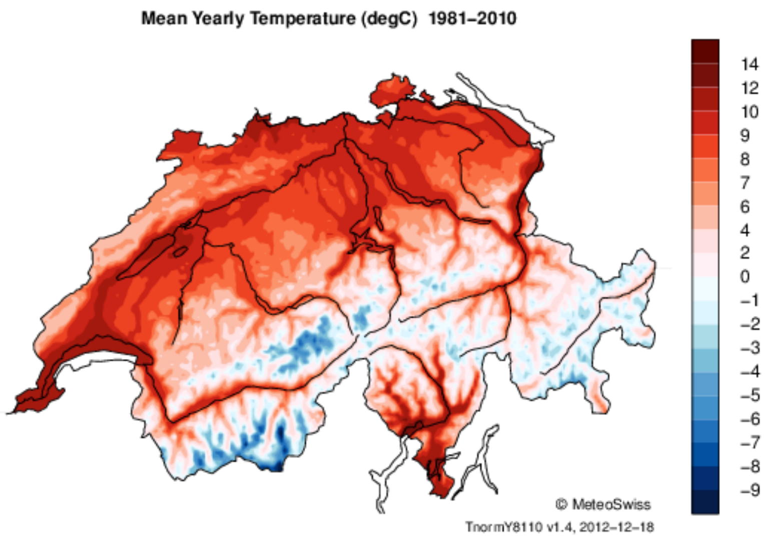
Social network data

Data are often structured

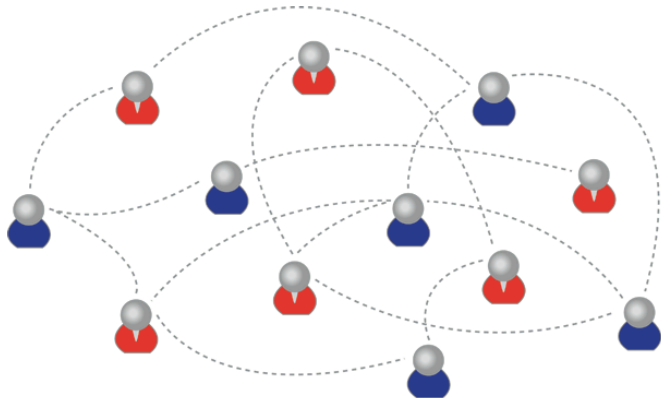


Electrical data

Traffic data



Temperature data



Social network data

We need to take into account the structure behind the data

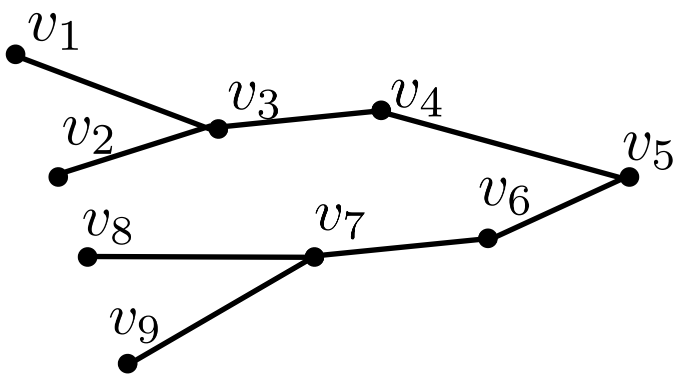
Graphs are appealing tools

- Efficient representations for **pairwise relations** between entities

The Königsberg Bridge Problem
[Leonhard Euler, 1736]

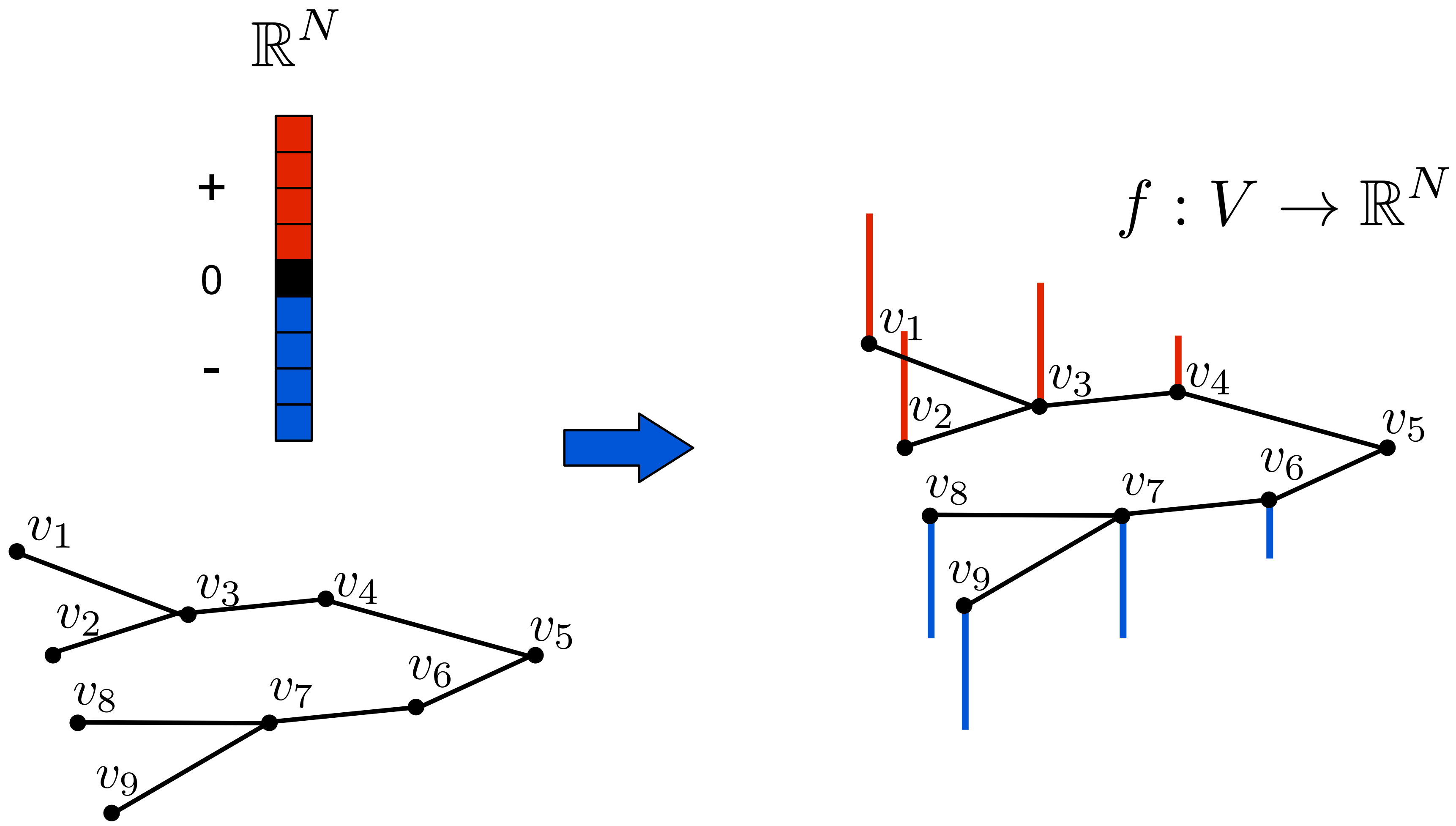
Graphs are appealing tools

- Efficient representations for pairwise relations between entities



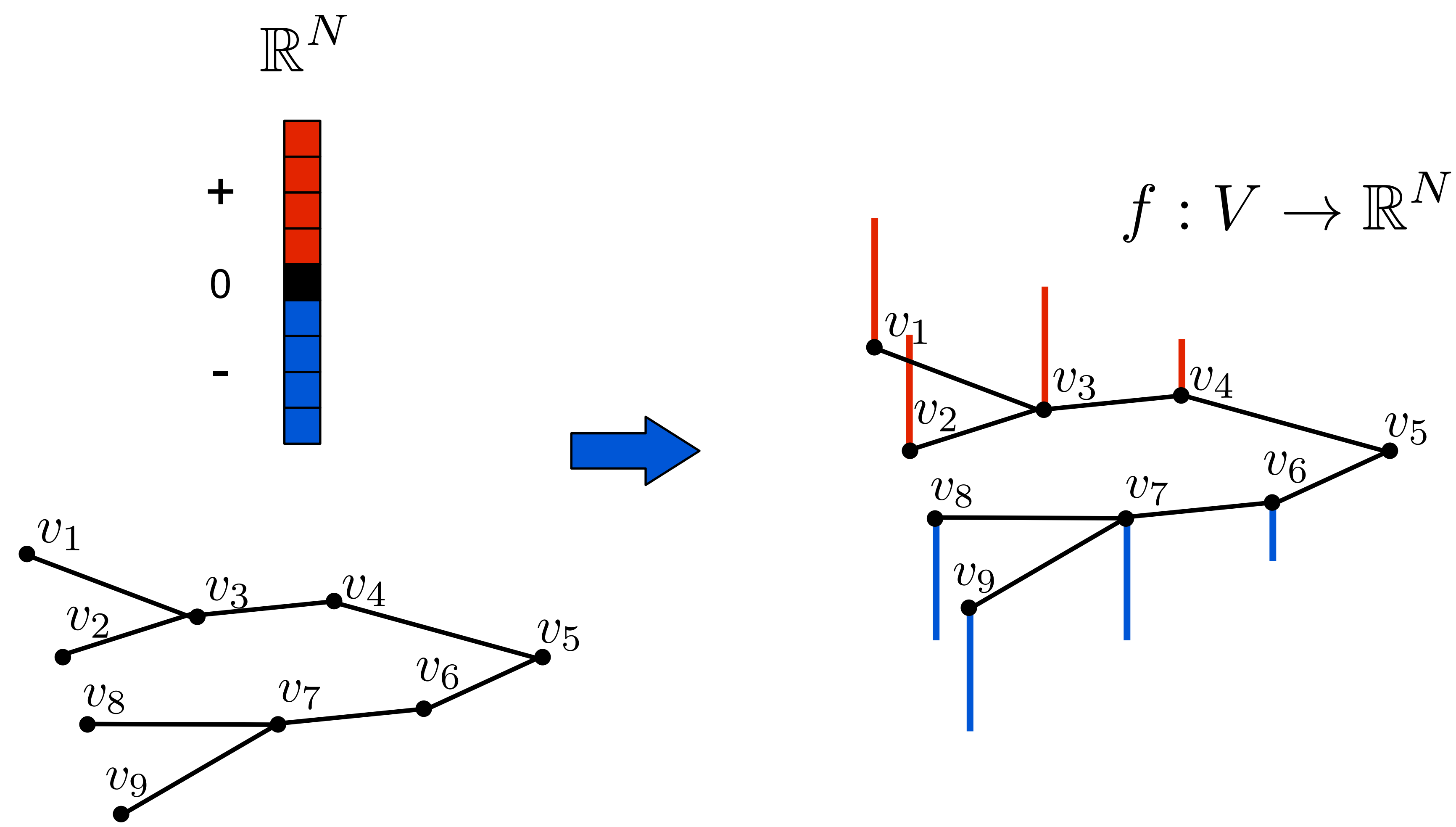
Graphs are appealing tools

- Efficient representations for pairwise relations between entities
- Structured data can be represented by **graph signals**



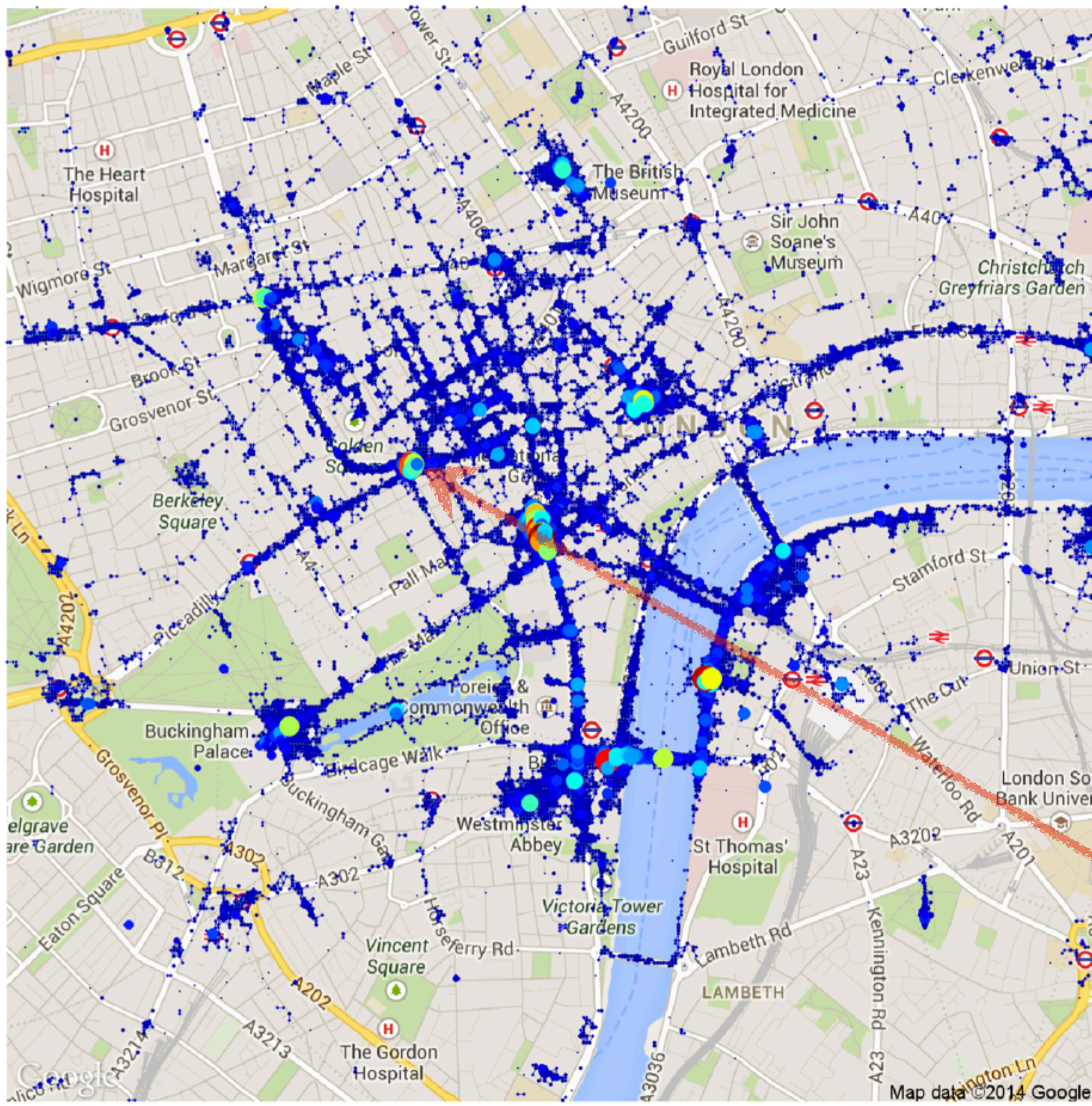
Graphs are appealing tools

- Efficient representations for pairwise relations between entities
- Structured data can be represented by **graph signals**

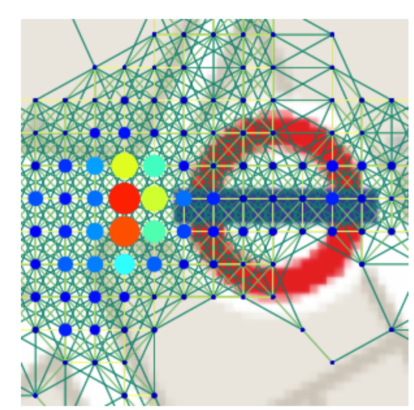


Takes into account both structure (edges) and data (values at vertices)

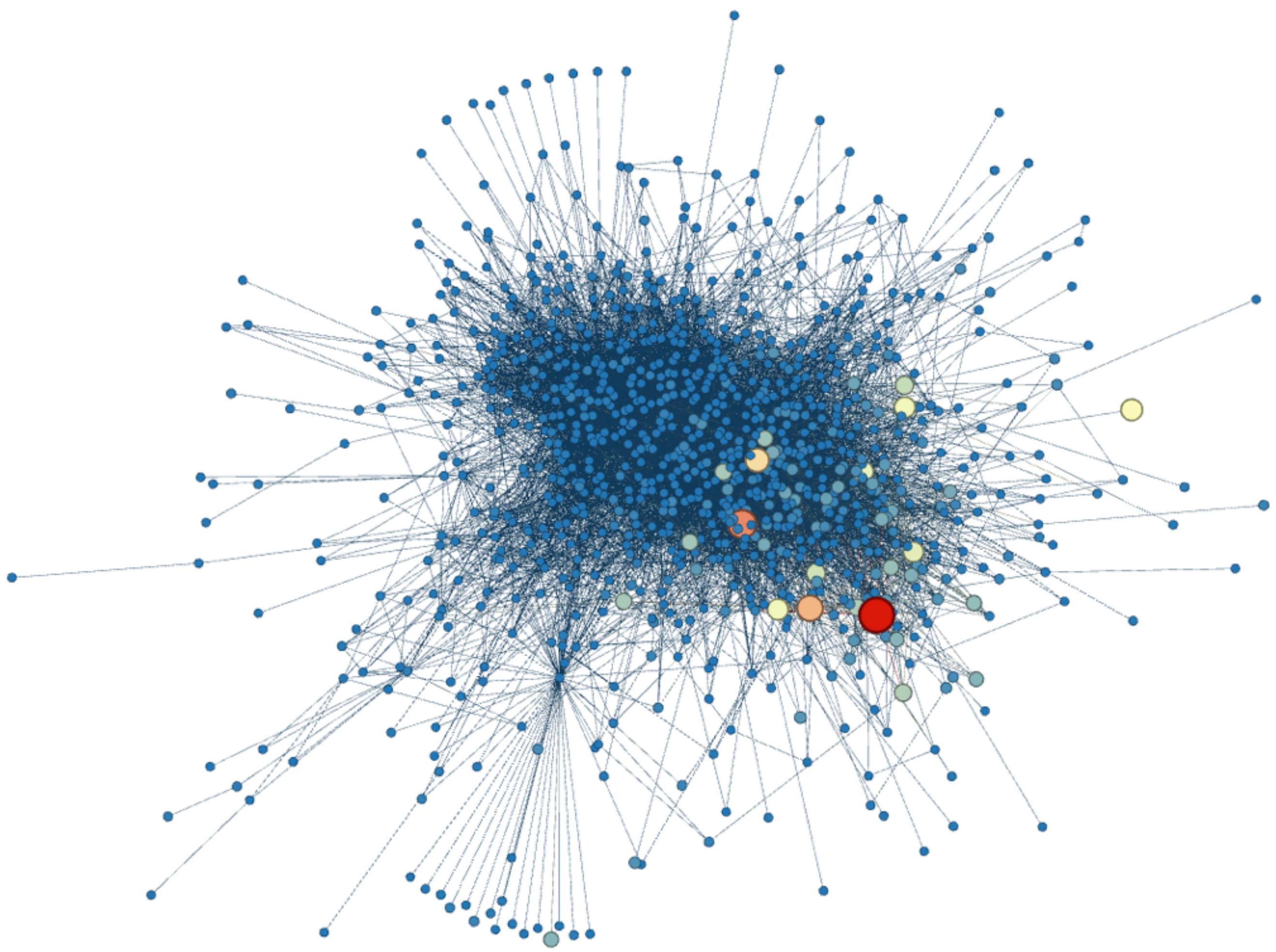
Graph signals are pervasive



- Vertices:
  - 9000 grid cells in London
- Edges:
  - Connecting cells that are geographically close
- Signal:
  - # Flickr users who have taken photos in two and a half year

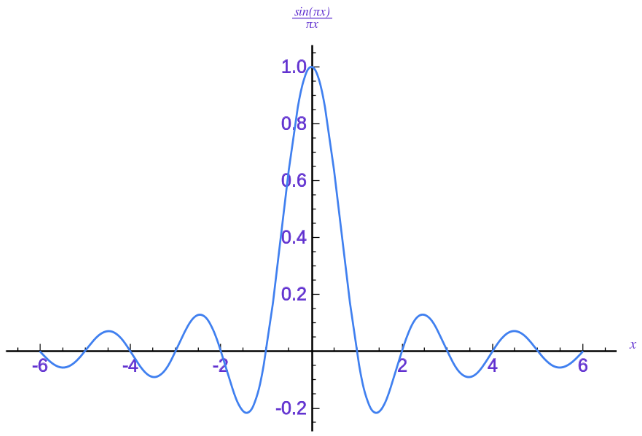


Graph signals are pervasive

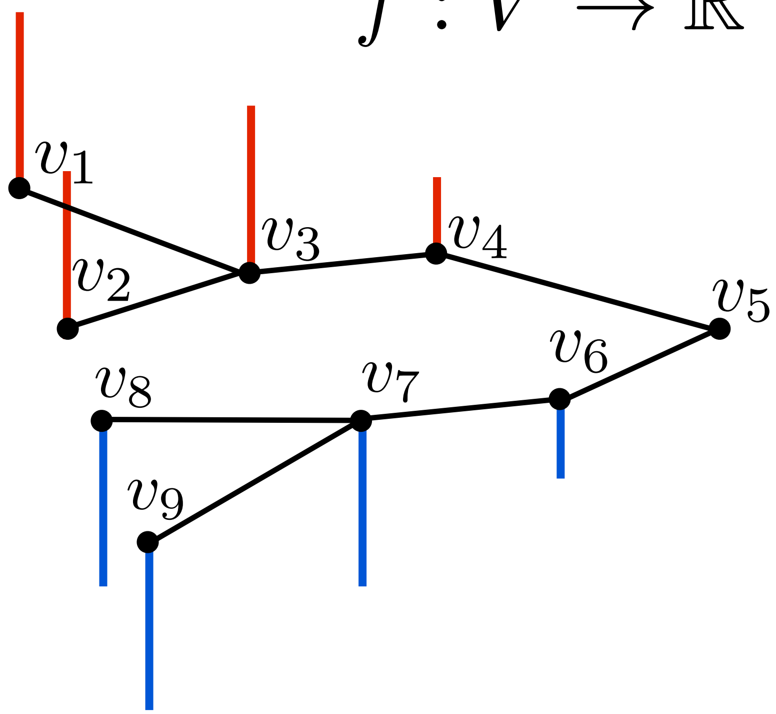


- Vertices:
  - 1000 Twitter users
- Edges:
  - Connecting users that have following relationship
- Signal:
  - # Apple-related hashtags they have posted in six weeks

Research challenges



$$f : V \to \mathbb{R}^N$$

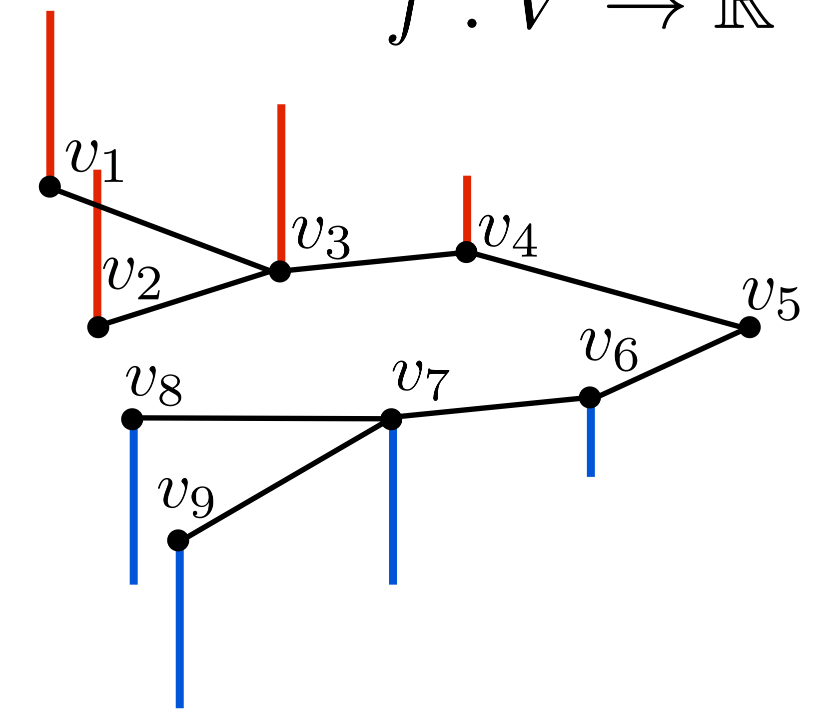


How to generalize classical signal processing tools on irregular domains such as graphs?

Graph signal processing

- Graph signals provide a nice compact format to encode structure within data
- Generalization of classical signal processing tools can greatly benefit analysis of such data
- Numerous applications: Transportation, biomedical, social network analysis, etc.
- An increasingly rich literature
  - classical signal processing
  - algebraic and spectral graph theory
  - computational harmonic analysis

$$f : V \to \mathbb{R}^N$$

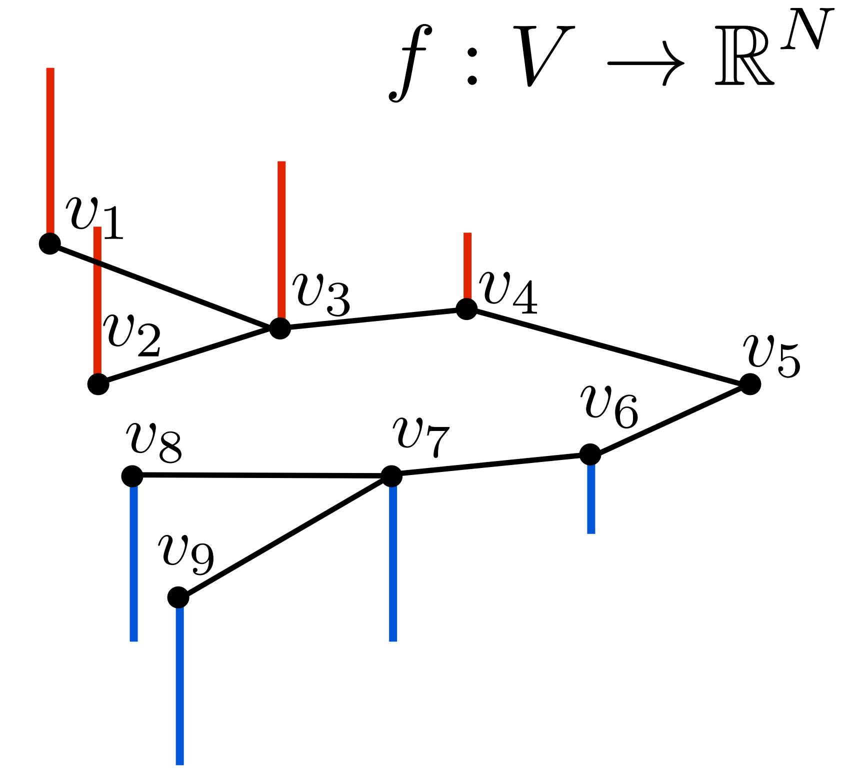


Outline

- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Applications and perspectives

Two paradigms

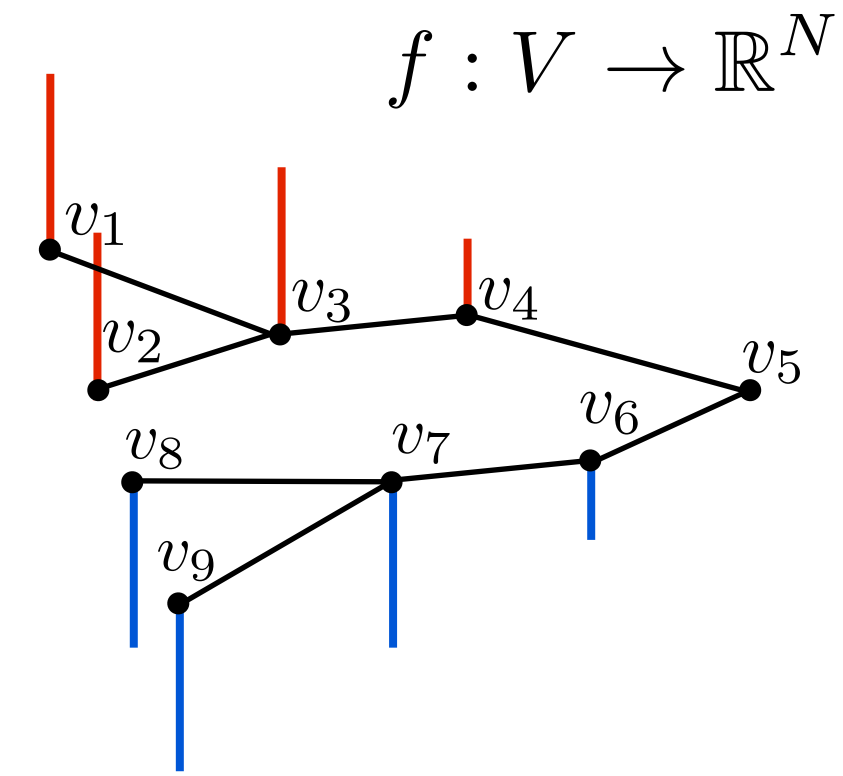
- The main approaches can be categorized into two families:
  - Vertex (spatial) domain designs
  - Frequency (graph spectral) domain designs



Two paradigms

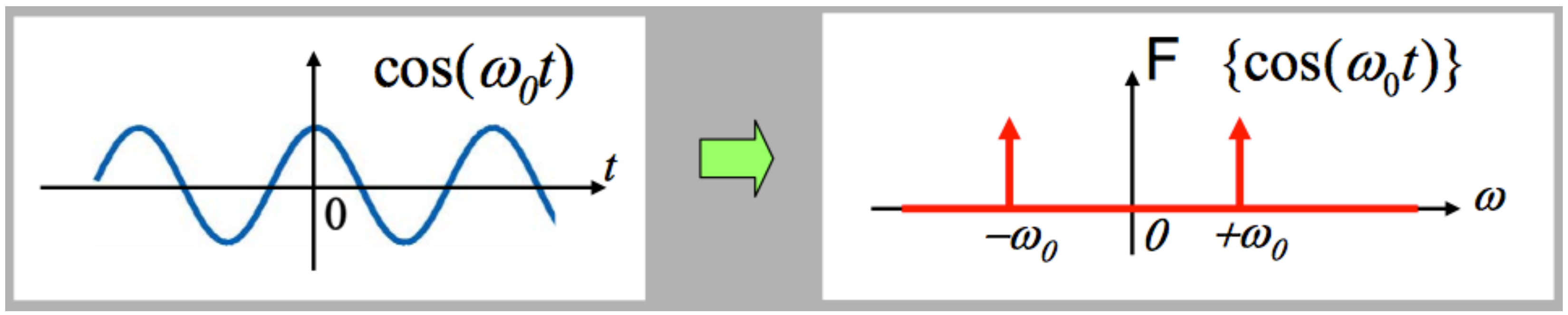
- The main approaches can be categorized into two families:
  - Vertex (spatial) domain designs
  - Frequency (graph spectral) domain designs

Important for analysis of signal properties

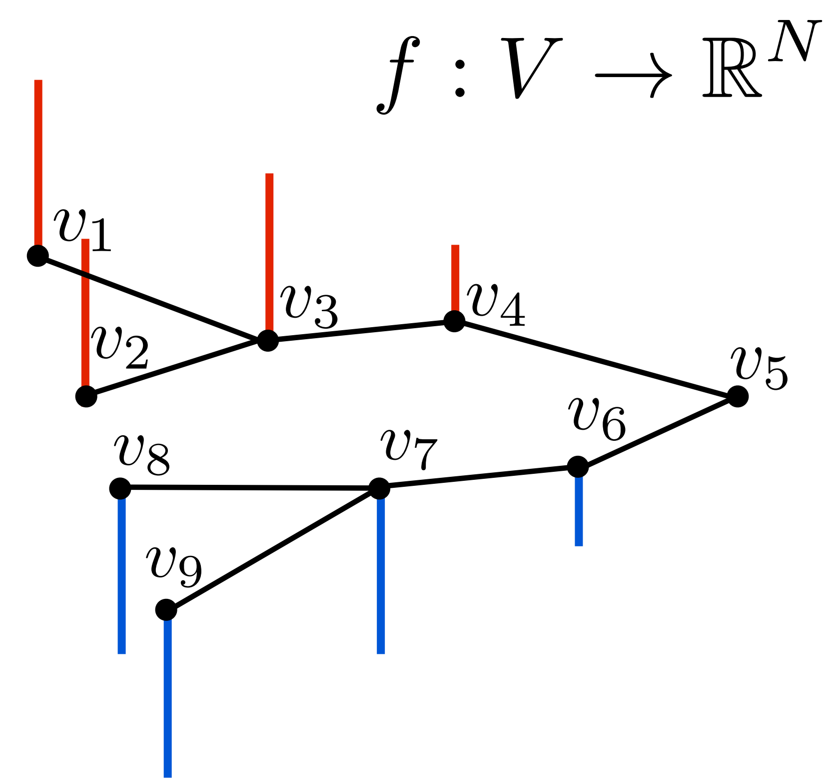


Need for frequency

- Classical Fourier transform provides the frequency domain representation of the signals

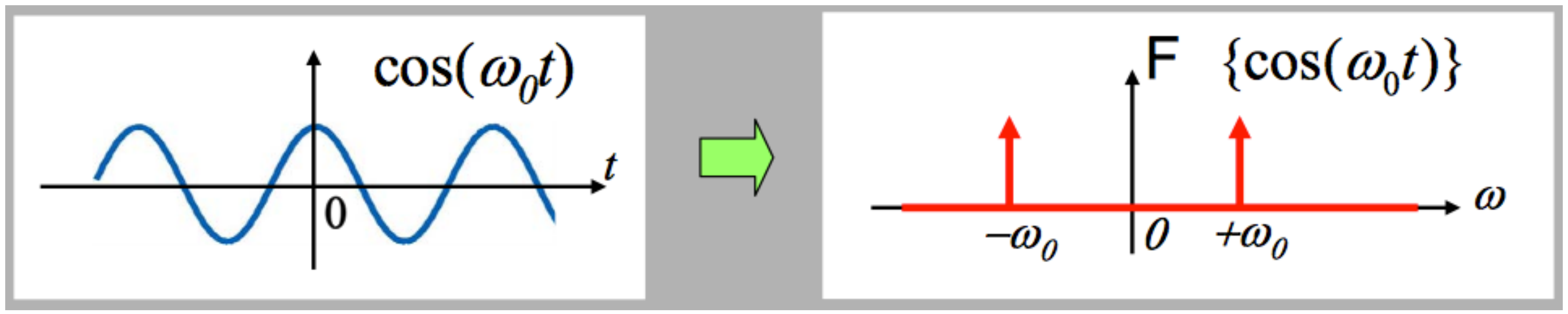


Source: http://www.physik.uni-kl.de



Need for frequency

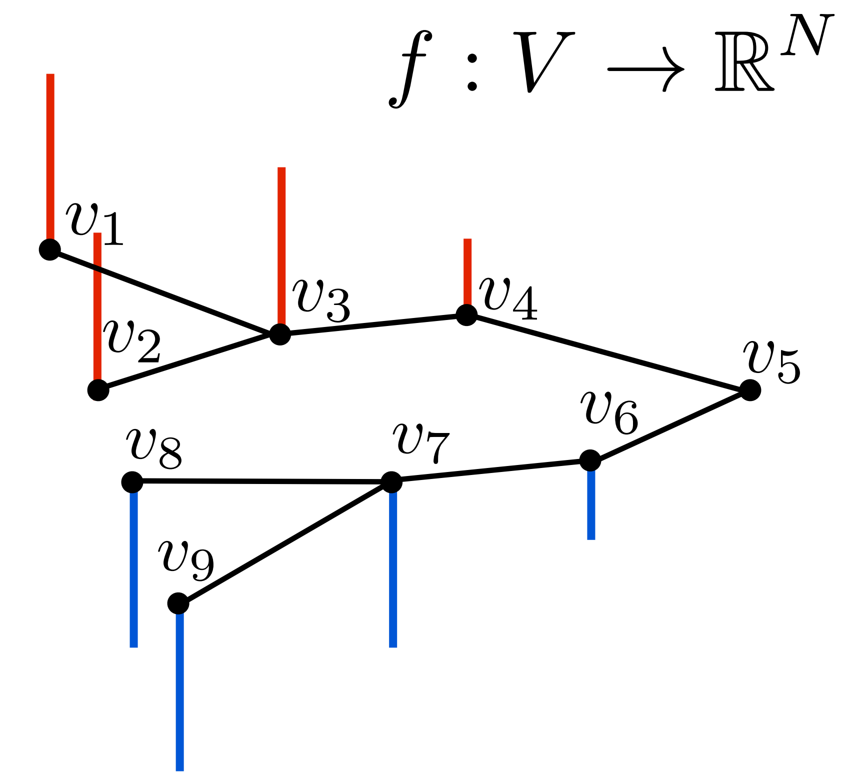
- Classical Fourier transform provides the frequency domain representation of the signals



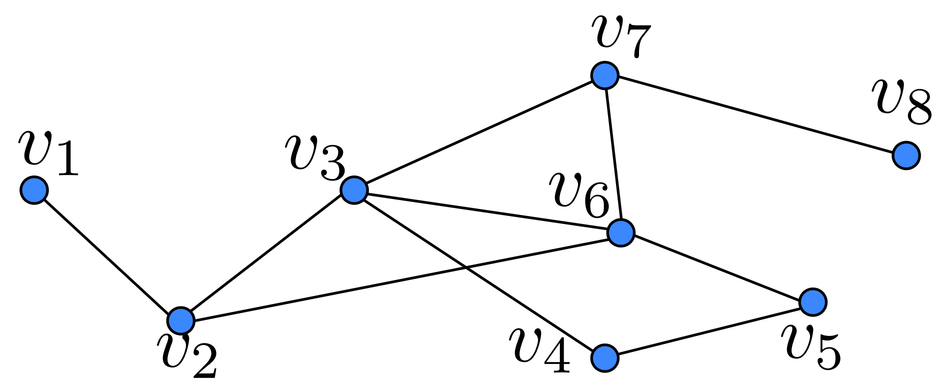
Source: http://www.physik.uni-kl.de

A notion of frequency for graph signals:

We need the graph Laplacian matrix



Graph Laplacian



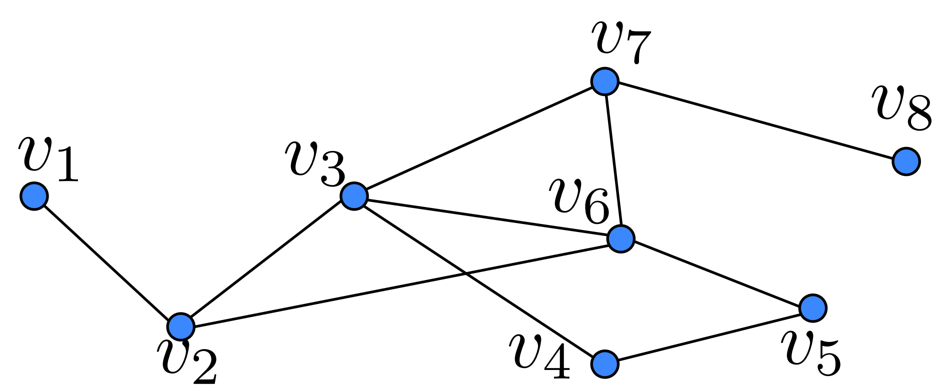
Weighted and undirected graph:

$G = \{V, E\}$

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

A

Graph Laplacian



Weighted and undirected graph:

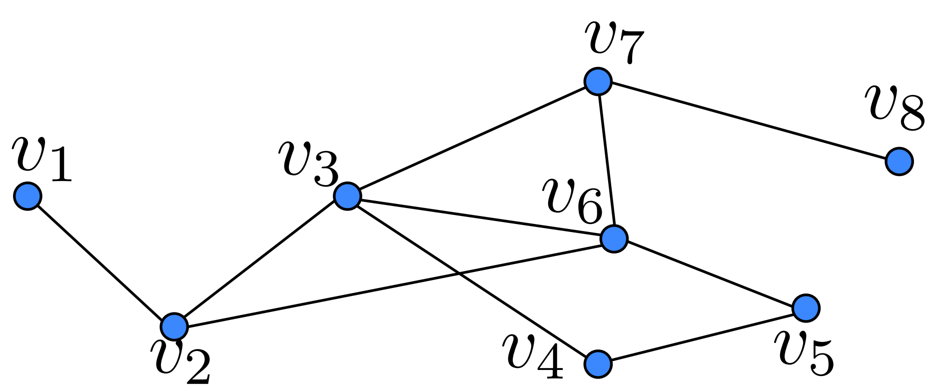
$G = \{V, E\}$

$D = \mathrm{diag}(\mathrm{degree}(v_1) \quad \ldots \quad \mathrm{degree}(v_N))$

$$\begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&3&0&0&0&0&0&0\\ 0&0&4&0&0&0&0&0\\ 0&0&0&2&0&0&0&0\\ 0&0&0&0&2&0&0&0\\ 0&0&0&0&0&4&0&0\\ 0&0&0&0&0&0&3&0\\ 0&0&0&0&0&0&0&1 \end{pmatrix} \qquad \begin{pmatrix} 0&1&0&0&0&0&0&0\\ 1&0&1&0&0&1&0&0\\ 0&1&0&1&0&1&1&0\\ 0&0&1&0&1&0&0&0\\ 0&0&0&1&0&1&0&0\\ 0&1&1&0&1&0&1&0\\ 0&0&1&0&0&1&0&1\\ 0&0&0&0&0&0&1&0 \end{pmatrix}$$

D $\qquad$ A

Graph Laplacian



Weighted and undirected graph:

$G = \{V, E\}$

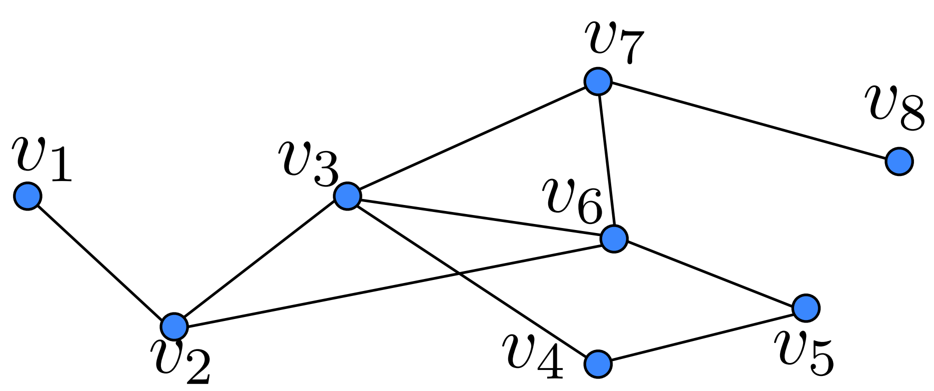
$D = \mathrm{diag}(\mathrm{degree}(v_1) \quad \ldots \quad \mathrm{degree}(v_N))$

$L = D - A$ **Equivalent to G!**

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

D A L

Graph Laplacian



Weighted and undirected graph:

$G = \{V, E\}$

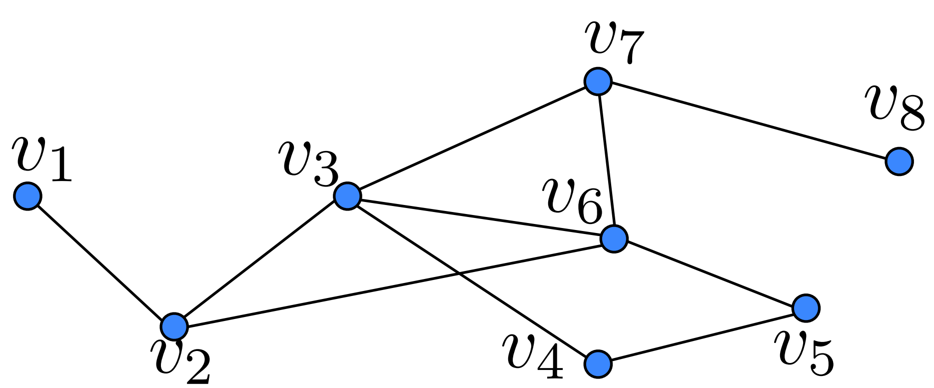
$D = \text{diag}(\text{degree}(v_1) \quad \dots \quad \text{degree}(v_N))$

$L = D - A$ **Equivalent to G!**

$$\underbrace{\begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&3&0&0&0&0&0&0\\ 0&0&4&0&0&0&0&0\\ 0&0&0&2&0&0&0&0\\ 0&0&0&0&2&0&0&0\\ 0&0&0&0&0&4&0&0\\ 0&0&0&0&0&0&3&0\\ 0&0&0&0&0&0&0&1 \end{pmatrix}}_{D} - \underbrace{\begin{pmatrix} 0&1&0&0&0&0&0&0\\ 1&0&1&0&0&1&0&0\\ 0&1&0&1&0&1&1&0\\ 0&0&1&0&1&0&0&0\\ 0&0&0&1&0&1&0&0\\ 0&1&1&0&1&0&1&0\\ 0&0&1&0&0&1&0&1\\ 0&0&0&0&0&0&1&0 \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 1&-1&0&0&0&0&0&0\\ -1&3&-1&0&0&-1&0&0\\ 0&-1&4&-1&0&-1&-1&0\\ 0&0&-1&2&-1&0&0&0\\ 0&0&0&-1&2&-1&0&0\\ 0&-1&-1&0&-1&4&-1&0\\ 0&0&-1&0&0&-1&3&-1\\ 0&0&0&0&0&0&-1&1 \end{pmatrix}}_{L}$$

- Symmetric
- Off-diagonal entries non-positive
- Rows sum up to zero

Graph Laplacian



Weighted and undirected graph:

$G = \{V, E\}$

$D = \mathrm{diag}(\mathrm{degree}(v_1) \quad \dots \quad \mathrm{degree}(v_N))$

$L = D - A$ **Equivalent to G!**

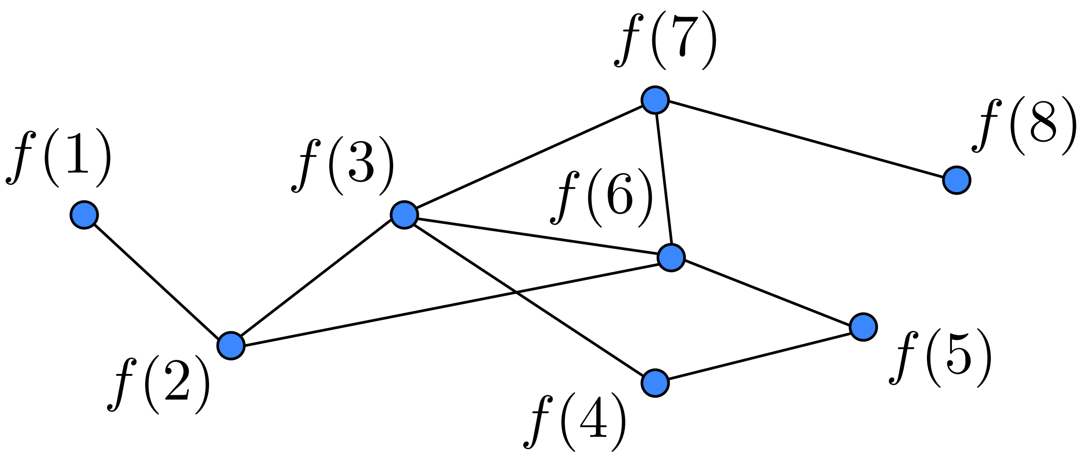
$$\underbrace{\begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&3&0&0&0&0&0&0\\ 0&0&4&0&0&0&0&0\\ 0&0&0&2&0&0&0&0\\ 0&0&0&0&2&0&0&0\\ 0&0&0&0&0&4&0&0\\ 0&0&0&0&0&0&3&0\\ 0&0&0&0&0&0&0&1 \end{pmatrix}}_{D} - \underbrace{\begin{pmatrix} 0&1&0&0&0&0&0&0\\ 1&0&1&0&0&1&0&0\\ 0&1&0&1&0&1&1&0\\ 0&0&1&0&1&0&0&0\\ 0&0&0&1&0&1&0&0\\ 0&1&1&0&1&0&1&0\\ 0&0&1&0&0&1&0&1\\ 0&0&0&0&0&0&1&0 \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 1&-1&0&0&0&0&0&0\\ -1&3&-1&0&0&-1&0&0\\ 0&-1&4&-1&0&-1&-1&0\\ 0&0&-1&2&-1&0&0&0\\ 0&0&0&-1&2&-1&0&0\\ 0&-1&-1&0&-1&4&-1&0\\ 0&0&-1&0&0&-1&3&-1\\ 0&0&0&0&0&0&-1&1 \end{pmatrix}}_{L}$$

- Symmetric
- Off-diagonal entries non-positive
- Rows sum up to zero

Why graph Laplacian?

- standard stencil approximation of the Laplace operator
- leads to a Fourier-like transform

Graph Laplacian

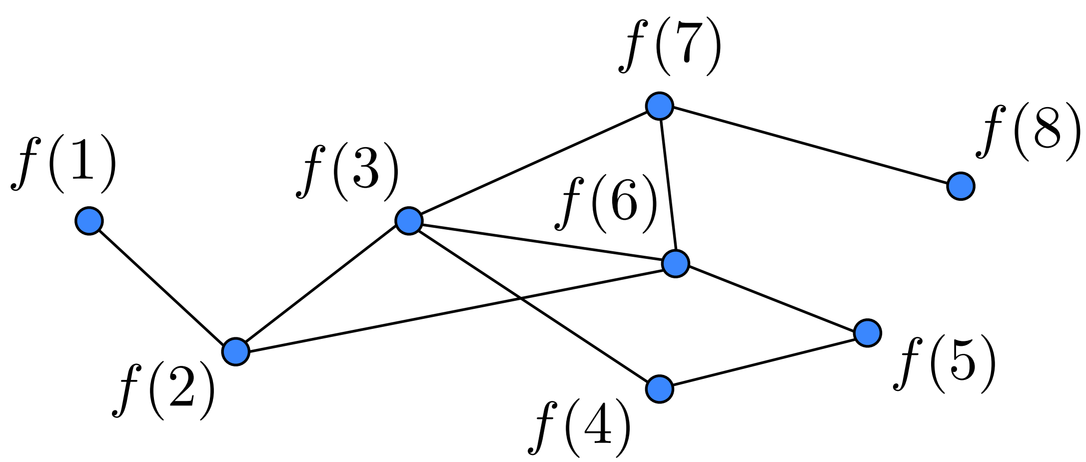


Graph signal $f : V \to \mathbb{R}^N$

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

$$L$$

Graph Laplacian



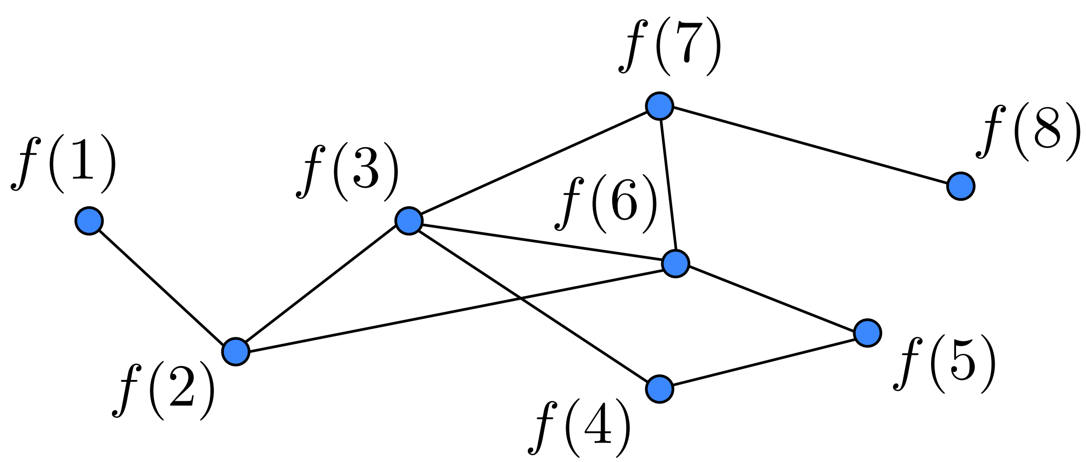
Graph signal $f : V \to \mathbb{R}^N$

$$\underbrace{\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \end{pmatrix}}_{f}$$

A difference operator:

$$Lf = \sum_{i,j=1}^{N} A_{ij} \left(f(i) - f(j) \right)$$

Graph Laplacian



Graph signal $f : V \to \mathbb{R}^N$

$$\underbrace{\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \\ f(8) \end{pmatrix}}_{f}$$

A difference operator:

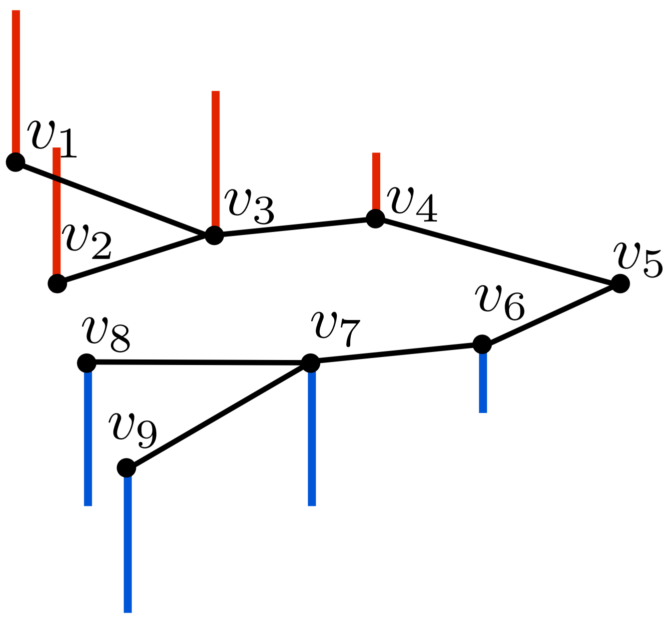
$$Lf = \sum_{i,j=1}^{N} A_{ij} \left(f(i) - f(j) \right)$$

Laplacian quadratic form:

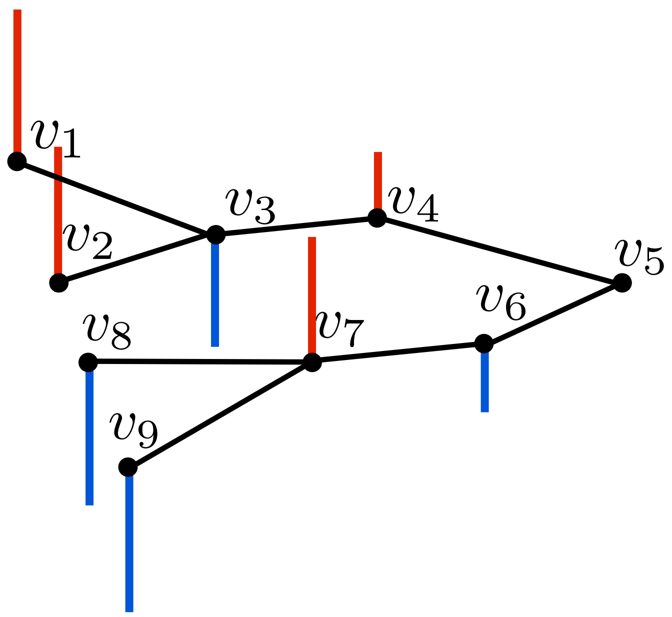
$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{N} A_{ij} \left(f(i) - f(j) \right)^2$$

A measure of "smoothness" [Zhou04]

Graph Laplacian



$f^T L f = 1$



$f^T L f = 21$

Graph Laplacian

- L has a complete set of orthonormal eigenvectors: $L = \chi \Lambda \chi^T$

$$L = \underbrace{\begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N\text{-}1} \\ | & & | \end{bmatrix}}_{\chi} \underbrace{\begin{bmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_{N-1} \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} — & \chi_0 & — \\ & \cdots & \\ — & \chi_{N\text{-}1} & — \end{bmatrix}}_{\chi^T}$$

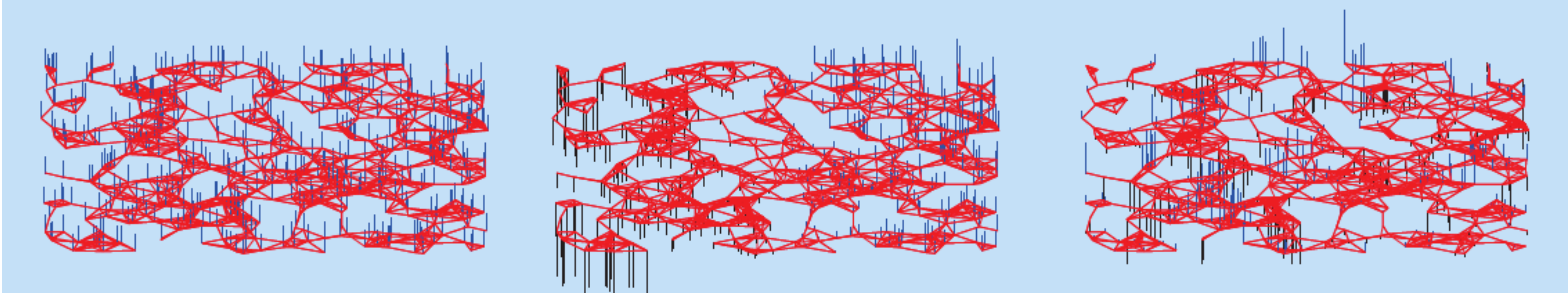
Graph Laplacian

- L has a complete set of orthonormal eigenvectors: $L = \chi \Lambda \chi^T$

$$L = \underbrace{\begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N\text{-}1} \\ | & & | \end{bmatrix}}_{\chi} \underbrace{\begin{bmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_{N-1} \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} — & \chi_0 & — \\ & \cdots & \\ — & \chi_{N\text{-}1} & — \end{bmatrix}}_{\chi^T}$$

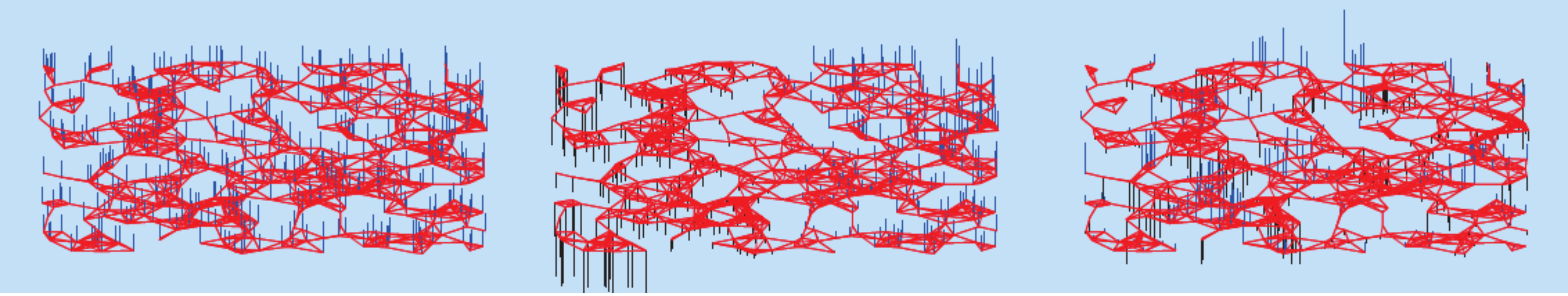
- Eigenvalues are usually sorted increasingly: $0 = \lambda_0 < \lambda_1 \leq \ldots \leq \lambda_{N-1}$

Graph Fourier transform



χ_0 χ_1 χ_{50} [Shuman13]

Graph Fourier transform



χ_0 χ_1 χ_{50} [Shuman13]

Low frequency → High frequency

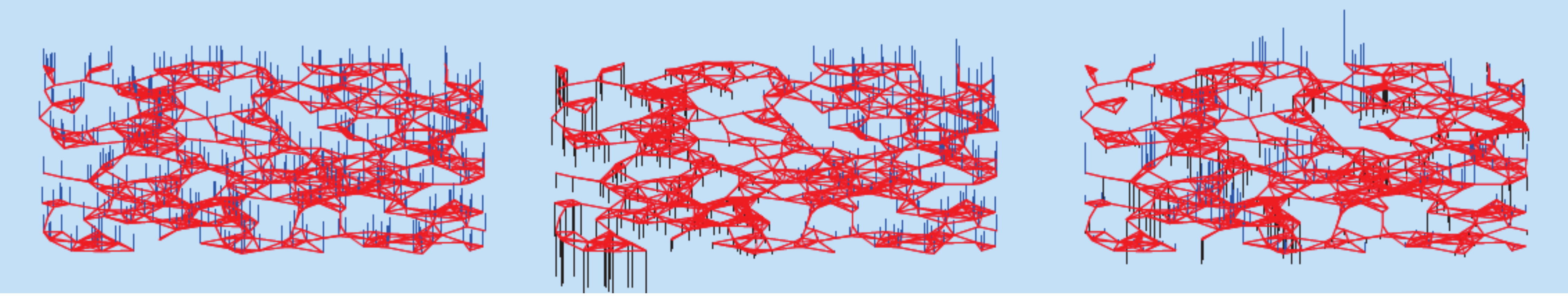
$L = \chi \Lambda \chi^T$

$\chi_0^T L \chi_0 = \lambda_0 = 0$

$\chi_{50}^T L \chi_{50} = \lambda_{50}$

- Eigenvectors associated with smaller eigenvalues have values that vary less rapidly along the edges

Graph Fourier transform



χ_0 χ_1 χ_{50} [Shuman13]

Low frequency → High frequency

$L = \chi \Lambda \chi^T$

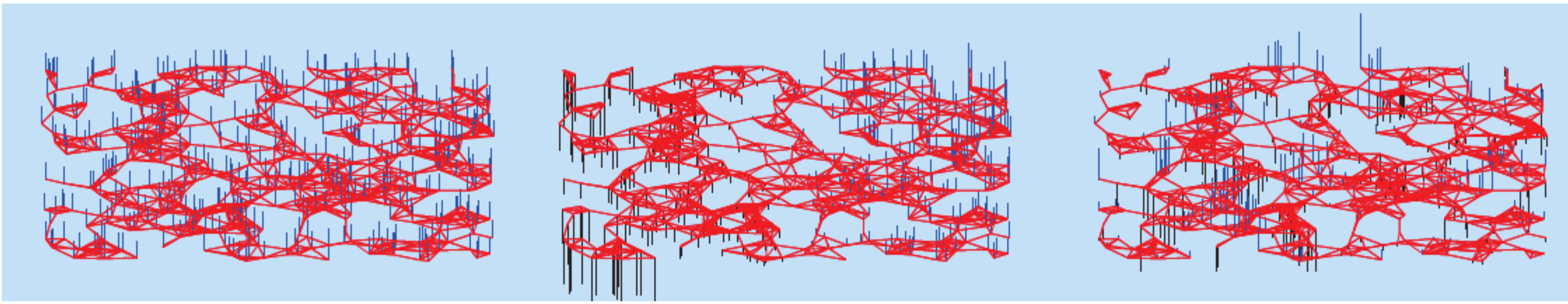
$\chi_0^T L \chi_0 = \lambda_0 = 0$

$\chi_{50}^T L \chi_{50} = \lambda_{50}$

Graph Fourier transform:
[Hammond11]

$$\hat{f}(\ell) = \langle \chi_\ell, f \rangle : \quad \begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N\text{-}1} \\ | & & | \end{bmatrix}^T \begin{matrix} | \\ f \\ | \end{matrix}$$

Graph Fourier transform



χ_0 χ_1 χ_{50} [Shuman13]

Low frequency → High frequency

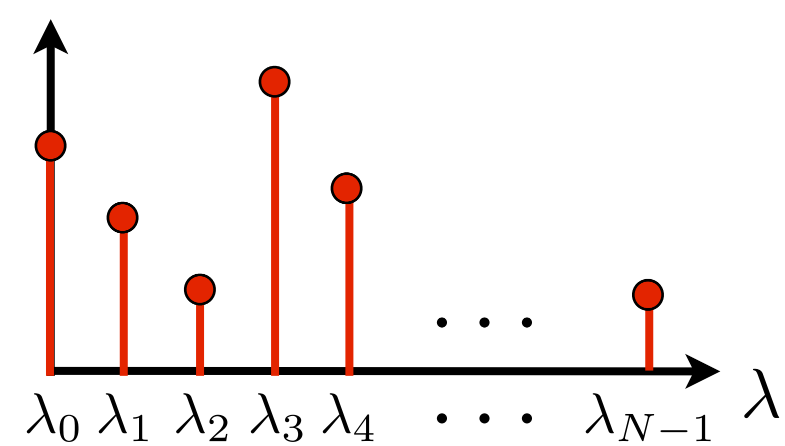
$L = \chi \Lambda \chi^T$

$\chi_0^T L \chi_0 = \lambda_0 = 0$

$\chi_{50}^T L \chi_{50} = \lambda_{50}$

<u>Graph Fourier transform:</u>
[Hammond11]

$$\hat{f}(\ell) = \langle \chi_\ell, f \rangle : \quad \begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N\text{-}1} \\ | & & | \end{bmatrix}^T \begin{matrix} | \\ f \\ | \end{matrix}$$



Low frequency High frequency

Graph Fourier transform

- The Laplacian L admits the following eigendecomposition: $L\chi_\ell = \lambda_\ell \chi_\ell$

Graph Fourier transform

- The Laplacian L admits the following eigendecomposition: $L\chi_\ell = \lambda_\ell \chi_\ell$

one-dimensional Laplace operator: $\frac{d^2}{dx^2}$

↓

eigenfunctions: $e^{j\omega x}$

↓

Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx$

$$f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$$

Graph Fourier transform

- The Laplacian L admits the following eigendecomposition: $L\chi_\ell = \lambda_\ell \chi_\ell$

one-dimensional Laplace operator: $\frac{d^2}{dx^2}$

eigenfunctions: $e^{j\omega x}$

Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x)dx$

$$f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$$

graph Laplacian: L

eigenvectors: χ_ℓ

$f : V \to \mathbb{R}^N$

Graph FT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i)$

$$f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell)\chi_\ell(i)$$

Graph Fourier transform

- The Laplacian L admits the following eigendecomposition: $L\chi_\ell = \lambda_\ell \chi_\ell$

one-dimensional Laplace operator: $\frac{d^2}{dx^2}$

eigenfunctions: $e^{j\omega x}$

Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx$

$$f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$$

graph Laplacian: L

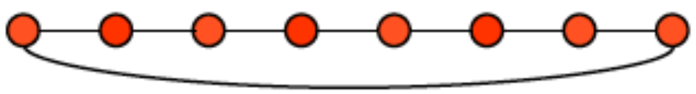
eigenvectors: χ_ℓ

$f : V \to \mathbb{R}^N$

Graph FT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i)$

$$f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$$

Two special cases



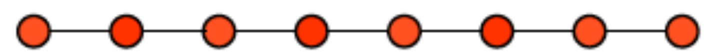
- (Unordered) Laplacian eigenvalues: $\lambda_\ell = 2 - 2\cos\left(\frac{2\ell\pi}{N}\right)$
- One possible choice of orthogonal Laplacian eigenvectors:

$$\chi_\ell = \left[1, \omega^\ell, \omega^{2\ell}, \ldots, \omega^{(N-1)\ell}\right], \text{ where } \omega = e^{\frac{2\pi j}{N}}$$

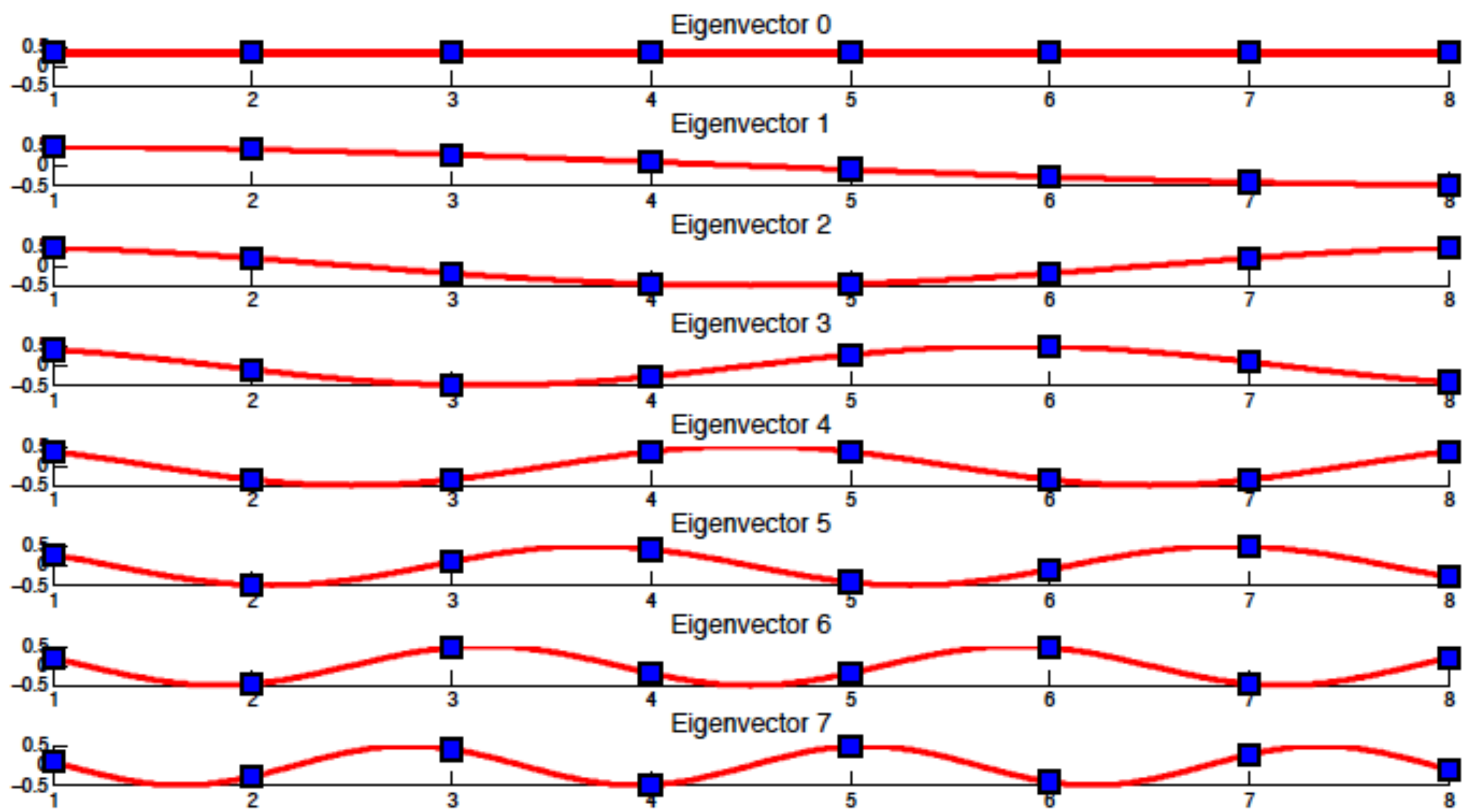
- $\begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N-1} \\ | & & | \end{bmatrix}$ is the Discrete Fourier Transform (DFT) matrix

[Vandergheynst11]

Two special cases



$\lambda_\ell = 2 - 2\cos\left(\frac{\pi \ell}{N}\right)$ $\chi_0(i) = \frac{1}{\sqrt{N}}$, $\chi_\ell(i) = \sqrt{\frac{2}{N}} \cos\left(\frac{\pi \ell (i-0.5)}{N}\right)$, $\ell = 1, 2, \ldots, N-1$



$\begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{N-1} \\ | & & | \end{bmatrix}$ is the Discrete Cosine Transform matrix (DCT-II, Strang, 1999), which is used in JPEG image compression

[Vandergheynst11]

Outline

- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Applications and perspectives

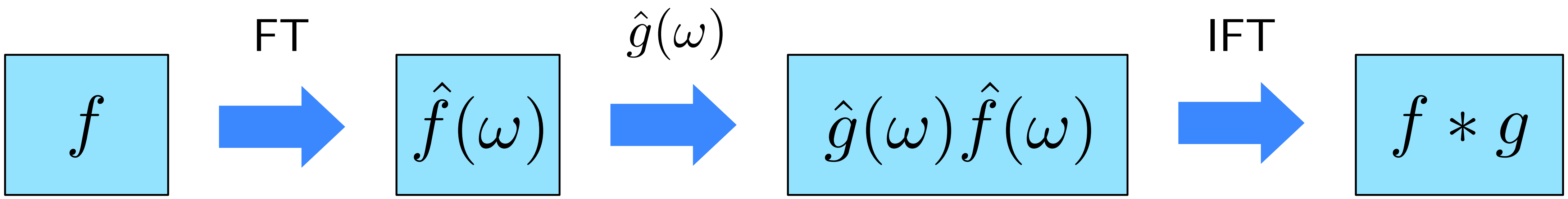
Classical frequency filtering

Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx \quad f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$

Classical frequency filtering

Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx \qquad f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$

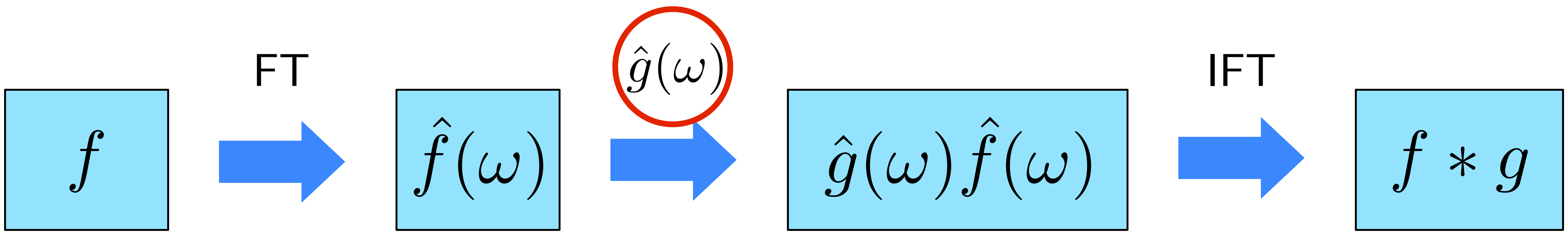
Apply filter with transfer function $\hat{g}(\cdot)$ to a signal f

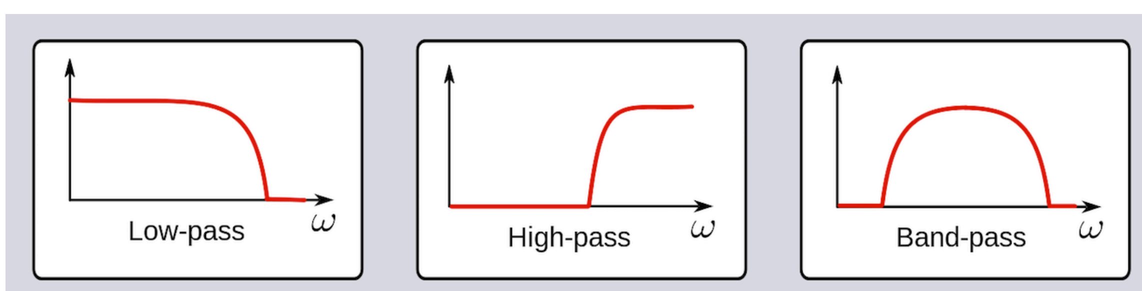


Classical frequency filtering

Classical FT: $\hat{f}(\omega) = \int (e^{j\omega x})^* f(x) dx \qquad f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{j\omega x} d\omega$

Apply filter with transfer function $\hat{g}(\cdot)$ to a signal f





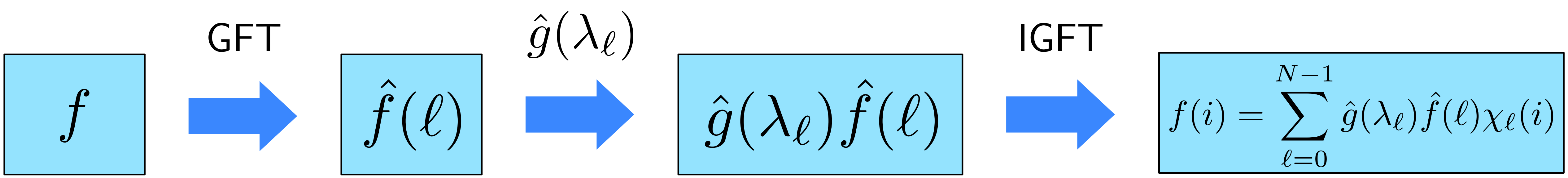
Graph spectral filtering

GFT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$

Graph spectral filtering

GFT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$

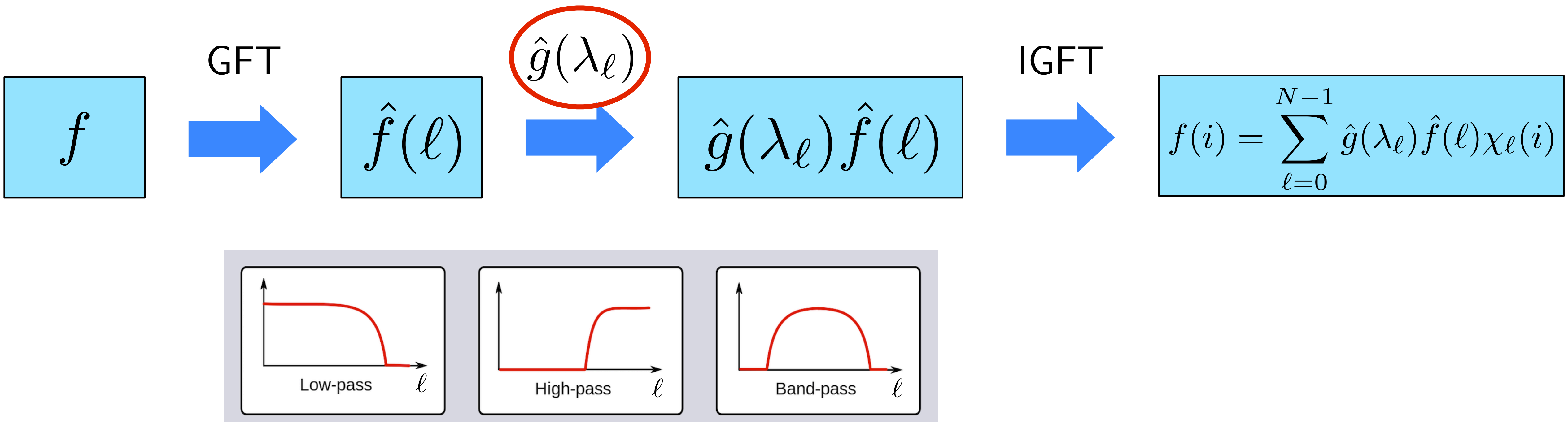
Apply filter with transfer function $\hat{g}(\cdot)$ to a graph signal $f : V \to \mathbb{R}^n$



Graph spectral filtering

GFT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell)\chi_\ell(i)$

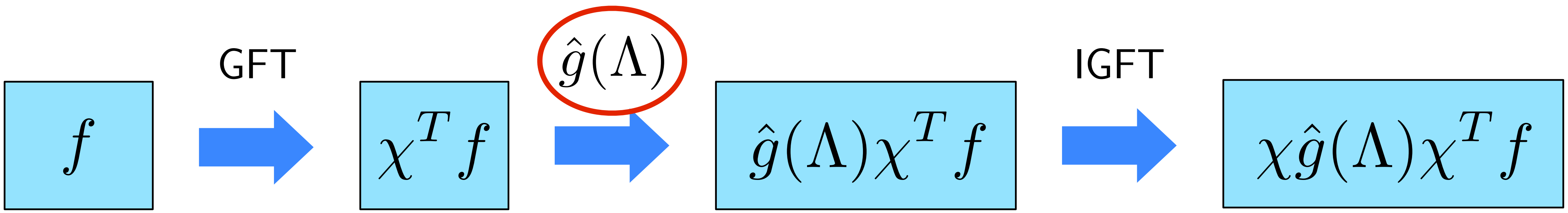
Apply filter with transfer function $\hat{g}(\cdot)$ to a graph signal $f : V \to \mathbb{R}^n$



Graph spectral filtering

GFT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell)\chi_\ell(i)$

Apply filter with transfer function $\hat{g}(\cdot)$ to a graph signal $f : V \to \mathbb{R}^n$



$$\hat{g}(\Lambda) = \begin{bmatrix} \hat{g}(\lambda_0) & & 0 \\ & \ddots & \\ 0 & & \hat{g}(\lambda_{N-1}) \end{bmatrix}$$

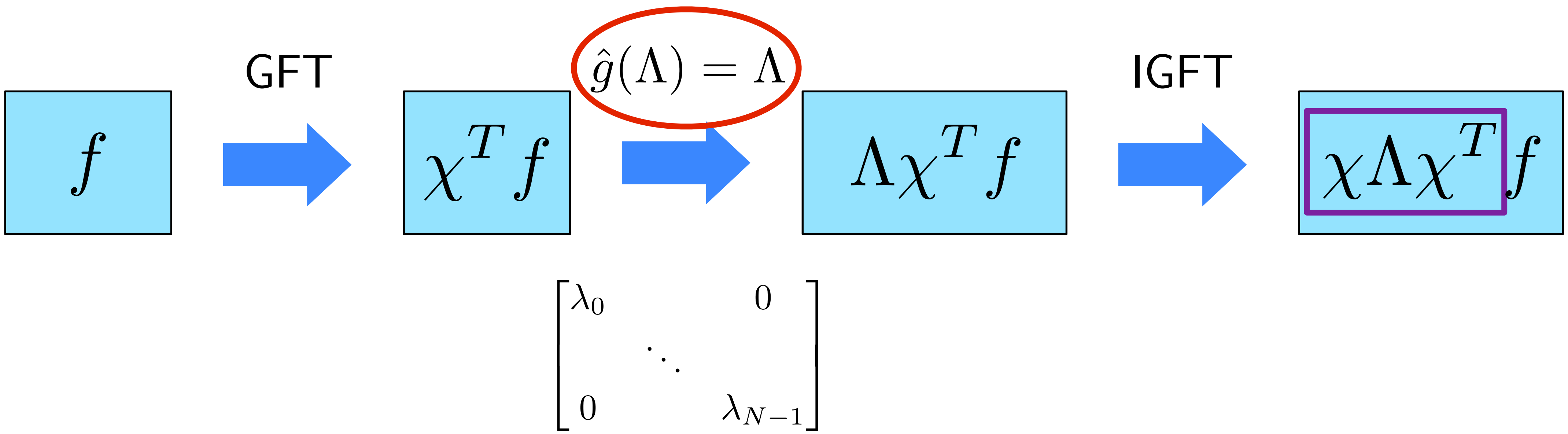
Graph Laplacian revisited

GFT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$

Graph Laplacian revisited

$$\text{GFT:}\quad \hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell)\chi_\ell(i)$$

The Laplacian L is a difference operator: $Lf = \chi \Lambda \chi^T f$

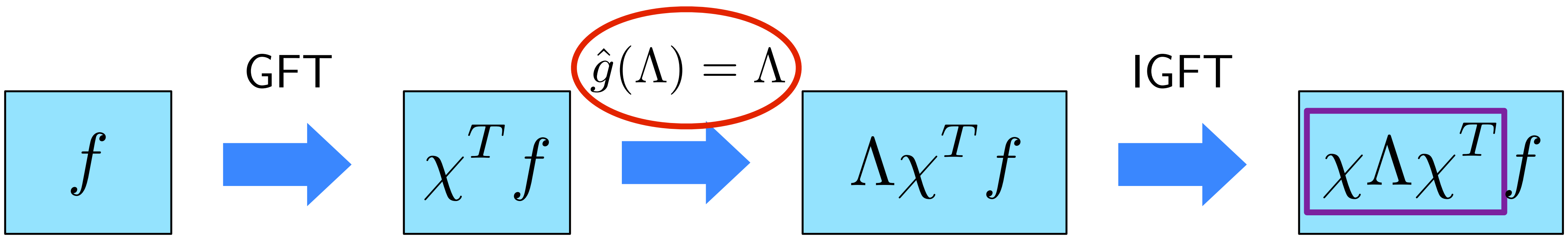


The Laplacian operator filters the signal in the spectral domain by its eigenvalues!

Graph Laplacian revisited

$$\text{GFT:} \quad \hat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{i=1}^{N} \chi_\ell^*(i) f(i) \qquad f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \chi_\ell(i)$$

The Laplacian L is a difference operator: $Lf = \chi \Lambda \chi^T f$



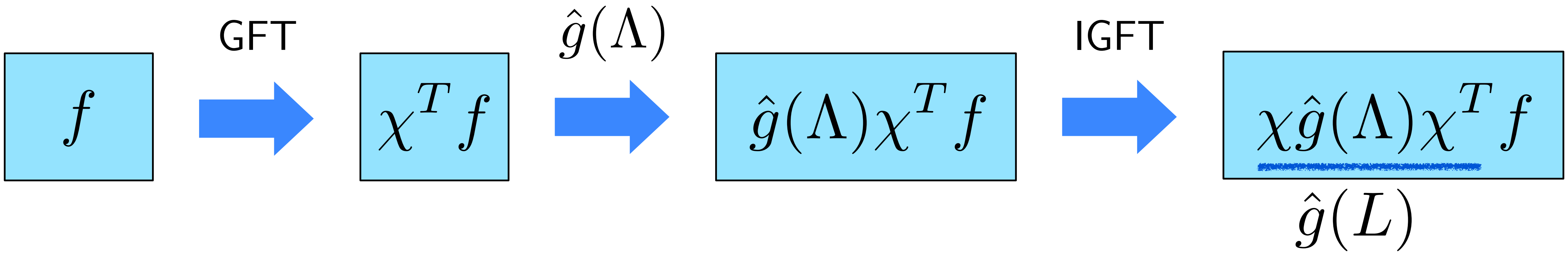
$$\begin{bmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_{N-1} \end{bmatrix}$$

The Laplacian operator filters the signal in the spectral domain by its eigenvalues!

The Laplacian quadratic form: $f^T L f = ||L^{\frac{1}{2}} f||_2 = ||\chi \Lambda^{\frac{1}{2}} \chi^T f||_2$

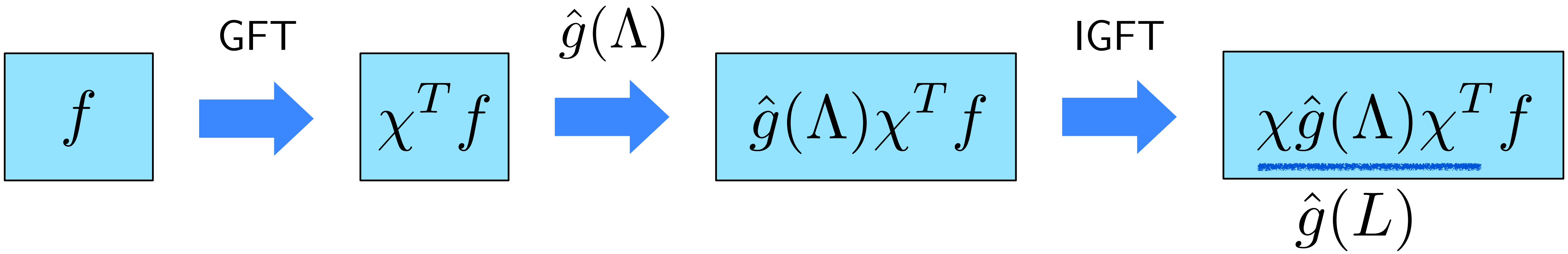
Graph transform/dictionary design

- Transforms and dictionaries can be designed through graph spectral filtering: Functions of graph Laplacian!



Graph transform/dictionary design

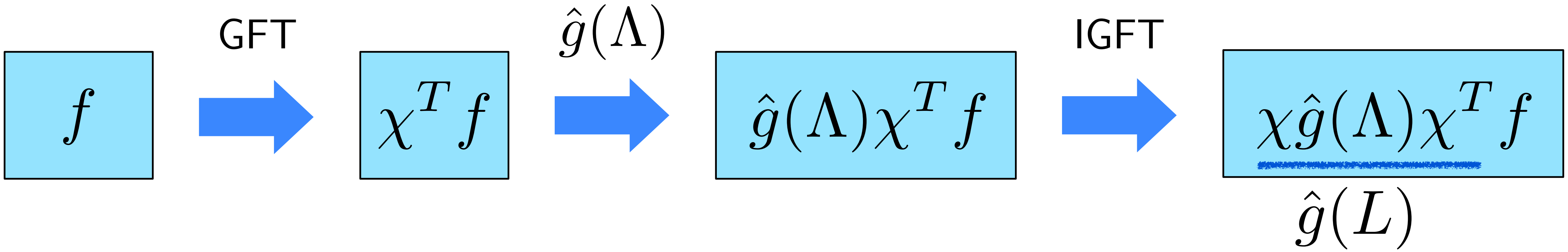
- Transforms and dictionaries can be designed through graph spectral filtering: Functions of graph Laplacian!



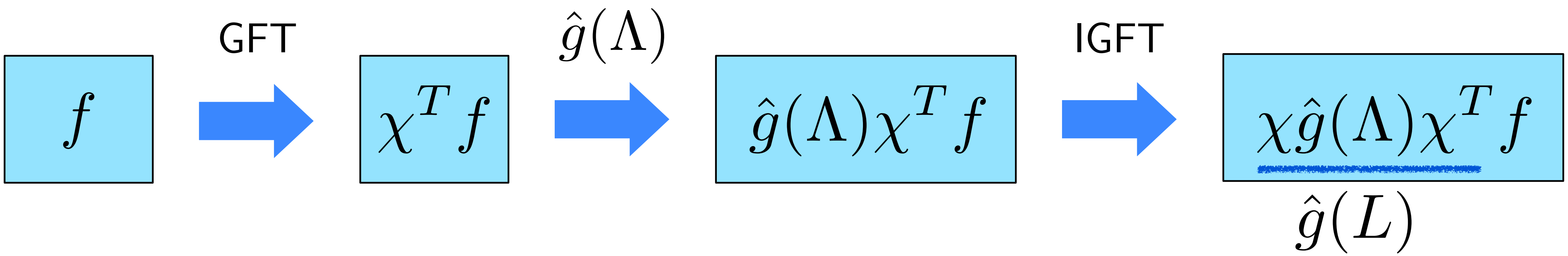
- Important properties can be achieved by properly defining $\hat{g}(L)$, such as localization of atoms

- Closely related to kernels and regularization on graphs [Smola03]

A simple example



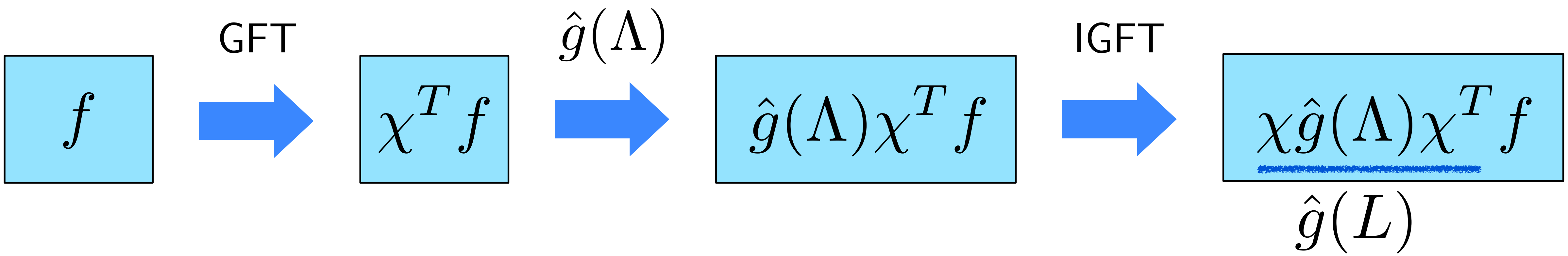
A simple example



Problem: We observe a noisy graph signal $f = y_0 + \eta$ and wish to recover y_0

$$y^* = \arg\min_y \{||y - f||_2^2 + \gamma y^T L y\}$$

A simple example



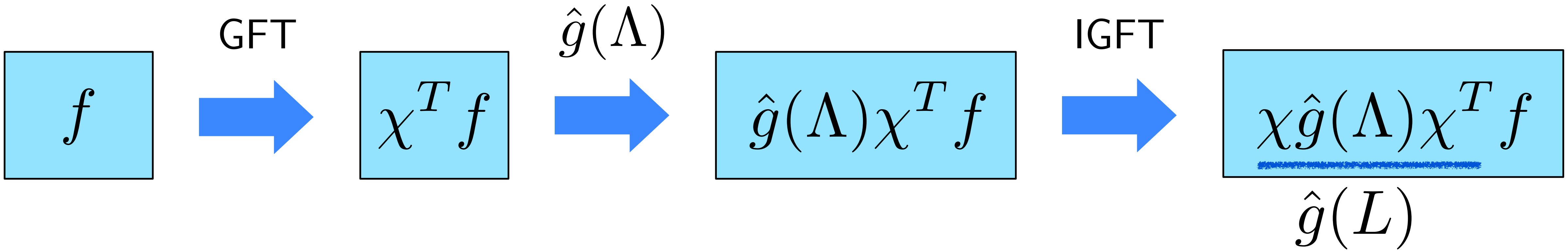
Problem: We observe a noisy graph signal $f = y_0 + \eta$ and wish to recover y_0

$$y^* = \arg\min_y \{||y - f||_2^2 + \gamma y^T L y\}$$

Data fitting term

"Smoothness" assumption

A simple example



Problem: We observe a noisy graph signal $f = y_0 + \eta$ and wish to recover y_0

$$y^* = \arg\min_y \{||y - f||_2^2 + \gamma y^T L y\}$$

Data fitting term

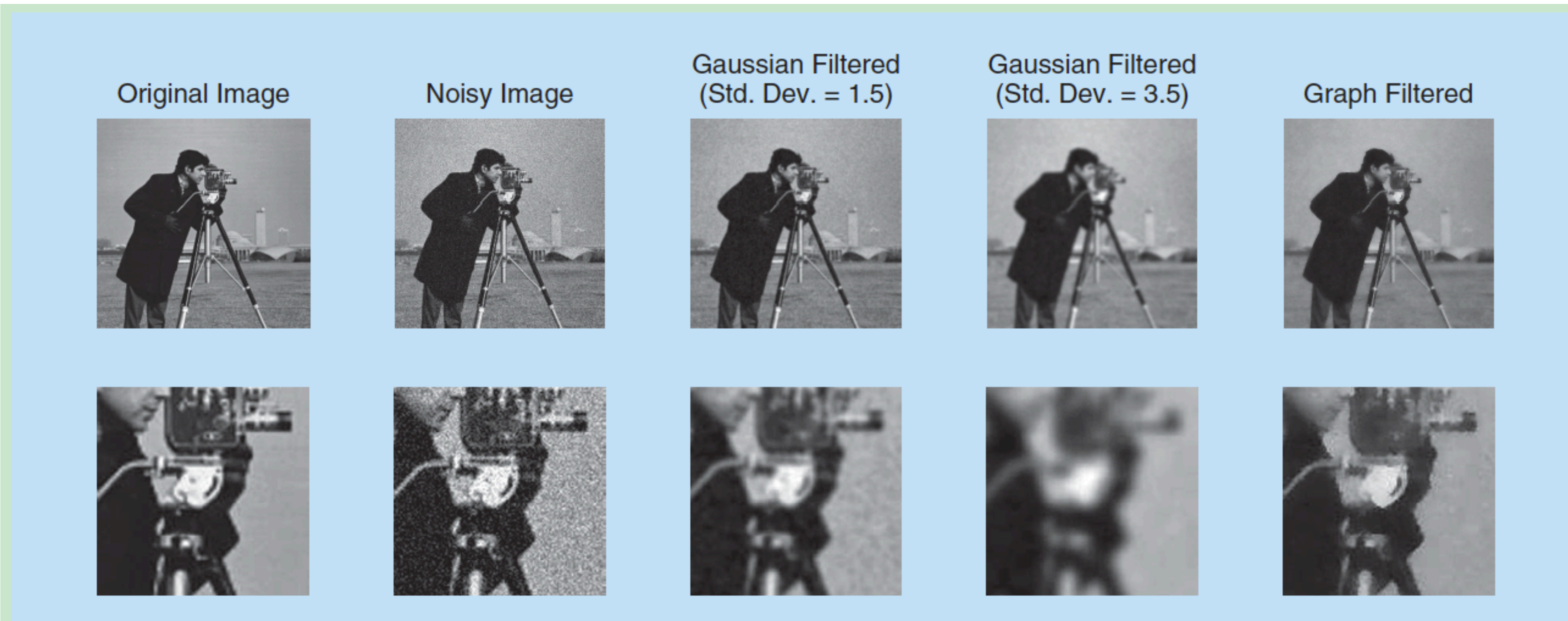
"Smoothness" assumption

$$y^* = \underbrace{(I + \gamma L)^{-1}}_{\hat{g}(L)} f$$

Laplacian (Tikhonov) regularization is equivalent to low-pass filtering in the graph spectral domain!

Example designs

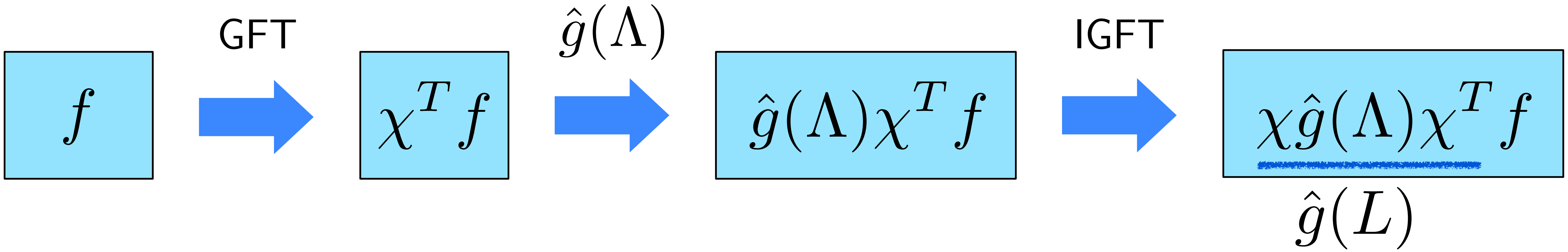
- Consider a noisy image as the observed noisy graph signal
- Consider a regular grid graph (weights inv. prop. to pixel value difference)



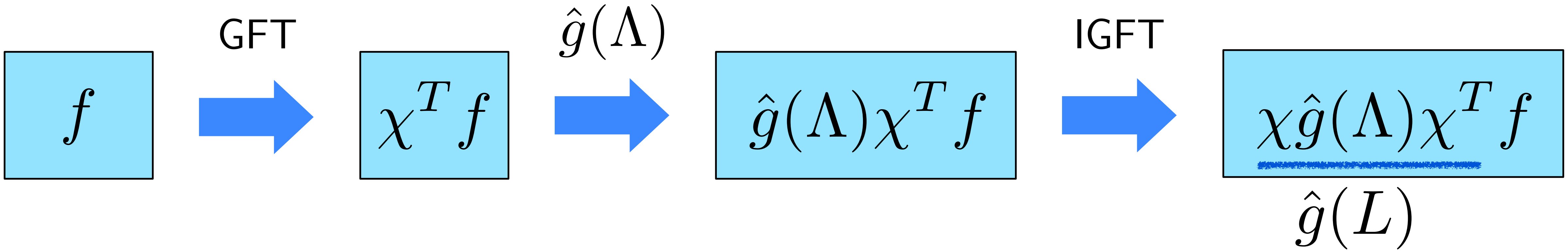
[FIGS2] Image denoising via classical Gaussian filtering and graph spectral filtering.

[Shuman13]

Example designs



Example designs



Low-pass filters: $\hat{g}(L) = (I + \gamma L)^{-1} = \chi(I + \gamma\Lambda)^{-1}\chi^T$

Shifted and dilated band-pass filters: Spectral graph wavelets $\hat{g}(sL)$ [Hammond11, Shuman11, Dong13]

Window kernel: Windowed graph Fourier transform [Shuman12]

Parametric polynomials: $\hat{g}_s(L) = \sum_{k=0}^{K} \alpha_{sk} L^k = \chi(\sum_{k=0}^{K} \alpha_{sk} \Lambda^k)\chi^T$ [Thanou14]

Adapted kernels: Learn values of $\hat{g}(L)$ directly from data [Zhang12]

Spectral graph wavelets

$$\psi_{s,a}(x) = \frac{1}{s}\psi\left(\frac{x-a}{s}\right)$$

$$W_f(s,a) = \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)\,dx$$

$$\bar{\psi}_s(x) = \frac{1}{s}\psi^*\left(\frac{-x}{s}\right)$$

$$(T^s f)(a) = \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)\,dx = \int_{-\infty}^{\infty} \bar{\psi}_s(a-x) f(x)\,dx$$

$$= (\bar{\psi}_s \star f)(a)$$

$$\widehat{T^s f}(\omega) = \hat{\bar{\psi}}_s(\omega)\hat{f}(\omega) = \hat{\psi}^*(s\omega)\hat{f}(\omega)$$

$$(T^s f)(a) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega a}\boxed{\hat{\psi}^*(s\omega)}\hat{f}(\omega)\,d\omega$$

Fourier multiplier operator: scaled kernel $\hat{\psi}^*(s\omega)$

Spectral graph wavelets

$$\psi_{s,a}(x) = \frac{1}{s}\psi\left(\frac{x-a}{s}\right)$$

$$W_f(s,a) = \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)\, dx$$

$$\bar{\psi}_s(x) = \frac{1}{s}\psi^*\left(\frac{-x}{s}\right)$$

$$(T^s f)(a) = \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)\, dx = \int_{-\infty}^{\infty} \bar{\psi}_s(a-x) f(x)\, dx$$

$$= (\bar{\psi}_s \star f)(a)$$

$$\widehat{T^s f}(\omega) = \hat{\bar{\psi}}_s(\omega)\hat{f}(\omega) = \hat{\psi}^*(s\omega)\hat{f}(\omega)$$

$$(T^s f)(a) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega a}\boxed{\hat{\psi}^*(s\omega)}\hat{f}(\omega)\, d\omega$$

Fourier multiplier operator: scaled kernel $\hat{\psi}^*(s\omega)$

$T_g^s = g(sL)$

$$\widehat{T_g^s f}(\ell) = g(s\lambda_\ell)\hat{f}(\ell)$$

Spectral graph wavelets

$$\psi_{s,a}(x) = \frac{1}{s}\psi\left(\frac{x-a}{s}\right)$$

$$W_f(s,a) = \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)\,dx$$

$$\bar{\psi}_s(x) = \frac{1}{s}\psi^*\left(\frac{-x}{s}\right)$$

$$(T^s f)(a) = \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)\,dx = \int_{-\infty}^{\infty} \bar{\psi}_s(a-x) f(x)\,dx$$

$$= (\bar{\psi}_s \star f)(a)$$

$$\widehat{T^s f}(\omega) = \hat{\bar{\psi}}_s(\omega)\hat{f}(\omega) = \hat{\psi}^*(s\omega)\hat{f}(\omega)$$

$$(T^s f)(a) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega a}\boxed{\hat{\psi}^*(s\omega)}\hat{f}(\omega)\,d\omega$$

Fourier multiplier operator: scaled kernel $\hat{\psi}^*(s\omega)$

$T_g^s = g(sL)$

$$\widehat{T_g^s f}(\ell) = g(s\lambda_\ell)\hat{f}(\ell)$$

$$(T_g^s f)(i) = \sum_{\ell=0}^{N-1} g(s\lambda_\ell)\hat{f}(\ell)\chi_\ell(i)$$

Spectral graph wavelets

$$\psi_{s,a}(x) = \frac{1}{s}\psi\left(\frac{x-a}{s}\right)$$

$$W_f(s,a) = \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)\, dx$$

$$\bar{\psi}_s(x) = \frac{1}{s}\psi^*\left(\frac{-x}{s}\right)$$

$$(T^s f)(a) = \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)\, dx = \int_{-\infty}^{\infty} \bar{\psi}_s(a-x) f(x)\, dx$$

$$= (\bar{\psi}_s \star f)(a)$$

$$\widehat{T^s f}(\omega) = \hat{\bar{\psi}}_s(\omega)\hat{f}(\omega) = \hat{\psi}^*(s\omega)\hat{f}(\omega)$$

$$(T^s f)(a) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega a}\hat{\psi}^*(s\omega)\hat{f}(\omega)\, d\omega$$

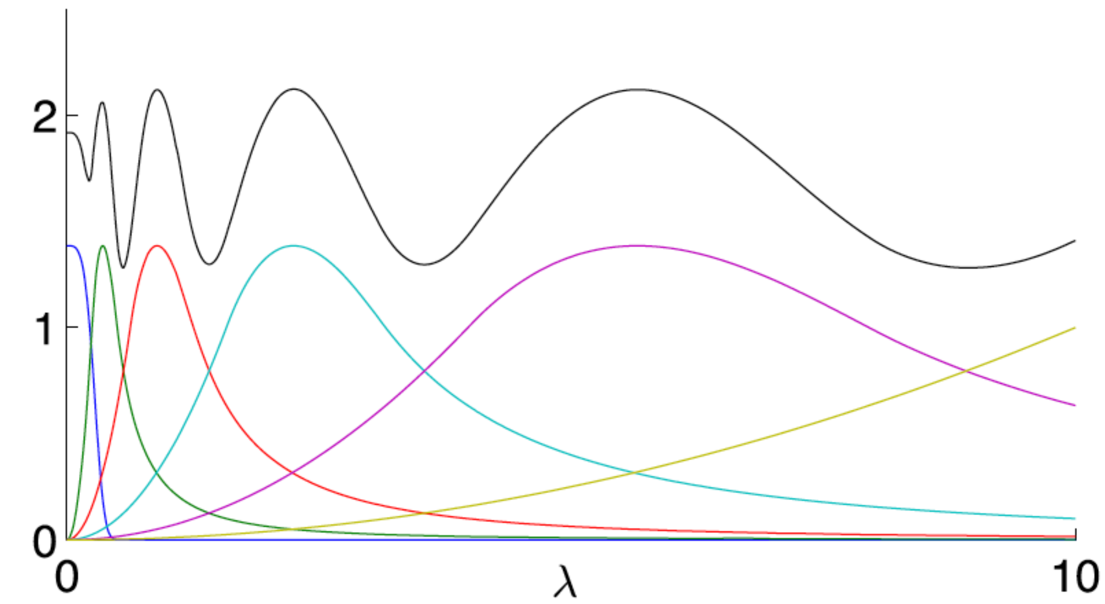
Fourier multiplier operator: scaled kernel $\hat{\psi}^*(s\omega)$

$$T_g^s = g(sL)$$

$$\widehat{T_g^s f}(\ell) = g(s\lambda_\ell)\hat{f}(\ell)$$

$$(T_g^s f)(i) = \sum_{\ell=0}^{N-1} g(s\lambda_\ell)\hat{f}(\ell)\chi_\ell(i)$$

Spectral graph wavelets



$$\psi_{s,a}(x) = \frac{1}{s}\psi\left(\frac{x-a}{s}\right)$$

$$W_f(s,a) = \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)\, dx$$

$$\bar{\psi}_s(x) = \frac{1}{s}\psi^*\left(\frac{-x}{s}\right)$$

$$(T^s f)(a) = \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)\, dx = \int_{-\infty}^{\infty} \bar{\psi}_s(a-x) f(x)\, dx$$

$$= (\bar{\psi}_s \star f)(a)$$

$$\widehat{T^s f}(\omega) = \hat{\bar{\psi}}_s(\omega)\hat{f}(\omega) = \hat{\psi}^*(s\omega)\hat{f}(\omega)$$

$$(T^s f)(a) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega a}\, \hat{\psi}^*(s\omega)\, \hat{f}(\omega)\, d\omega$$

Fourier multiplier operator: scaled kernel $\hat{\psi}^*(s\omega)$

$$T_g^s = g(sL)$$

$$\widehat{T_g^s f}(\ell) = g(s\lambda_\ell)\hat{f}(\ell)$$

$$(T_g^s f)(i) = \sum_{\ell=0}^{N-1} g(s\lambda_\ell)\, \hat{f}(\ell)\, \chi_\ell(i)$$

Spectral graph wavelets

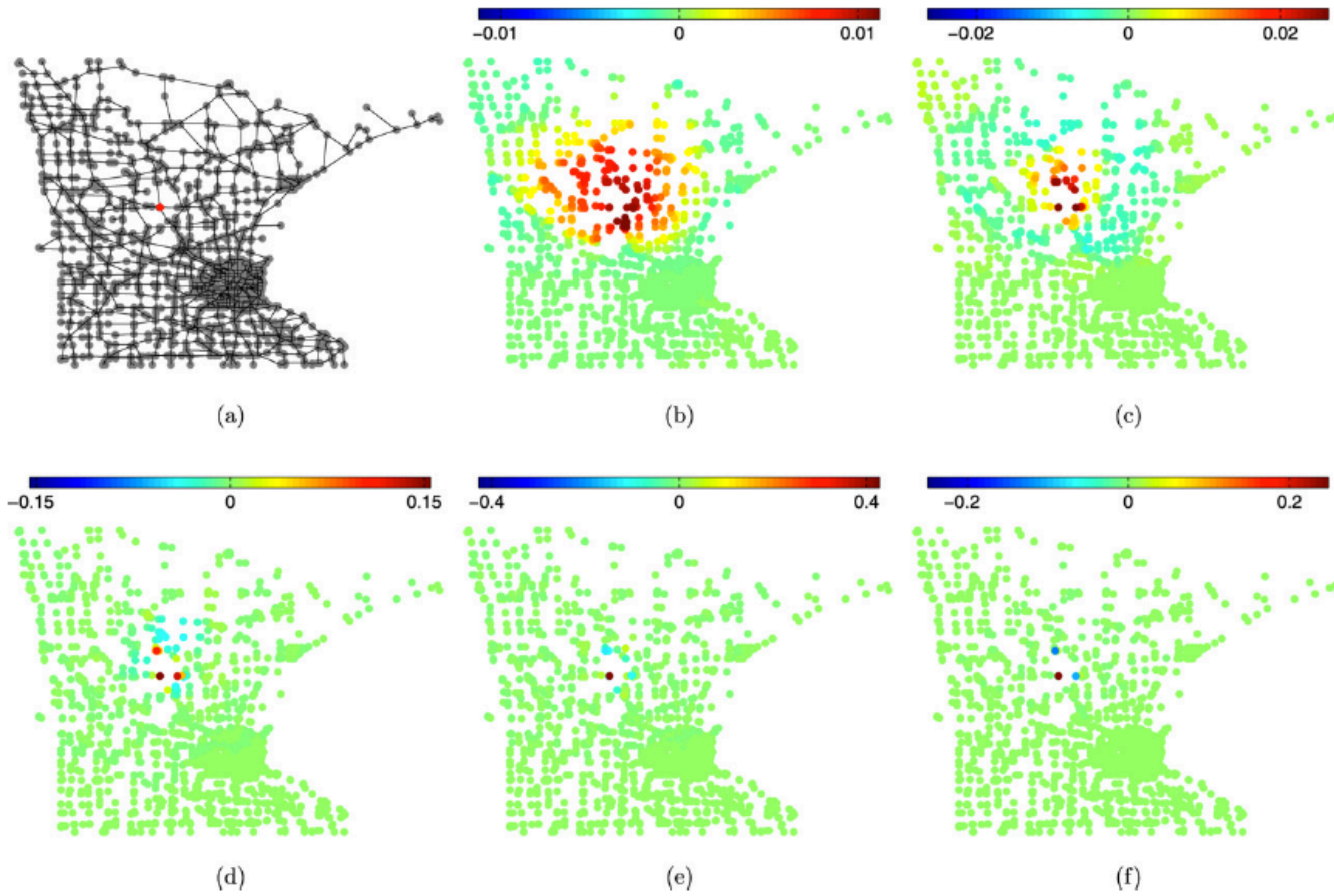


Fig. 4. Spectral graph wavelets on Minnesota road graph, with $K = 100$, $J = 4$ scales. (a) Vertex at which wavelets are centered, (b) scaling function, (c)–(f) wavelets, scales 1–4.

[Hammond11]

Outline

- Motivation
- Graph signal processing (GSP): Basic concepts
- Spectral filtering: Basic tools of GSP
- Applications and perspectives

Applications of GSP

- Signal processing and machine learning tasks
  - Denoising [Graichen15, Liu16]
  - Semi-supervised learning / Classification [Kipf16, Manessi17]
  - Clustering [Tremblay14, Tremblay16]
  - Dimensionality reduction [Rui16]

Applications of GSP

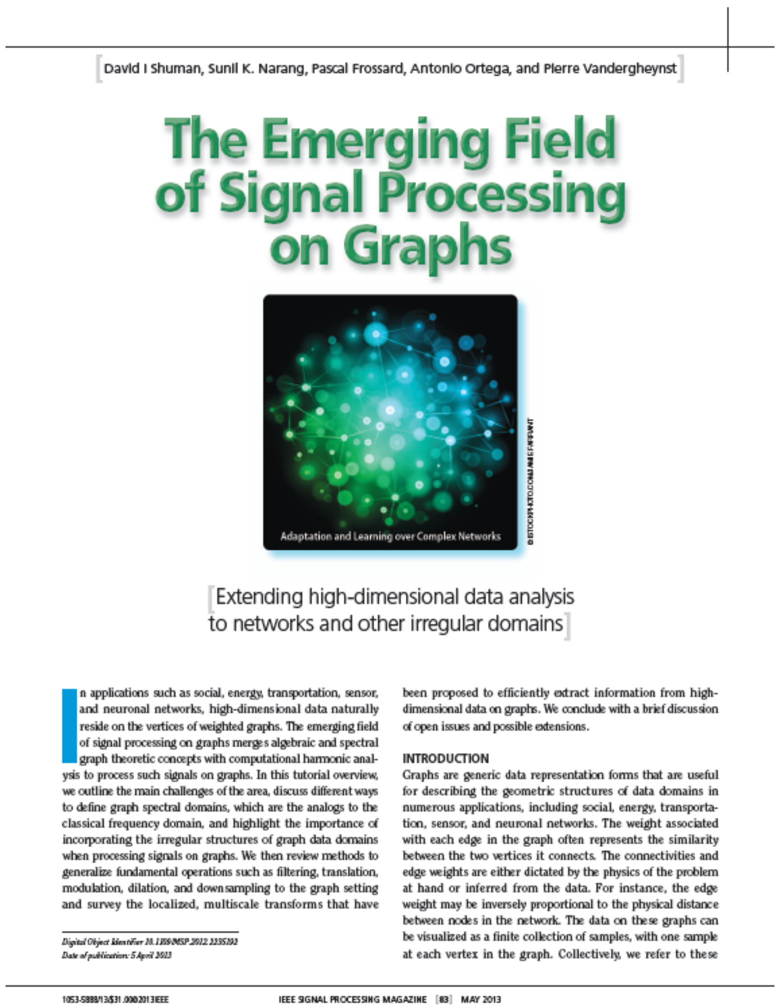
- Signal processing and machine learning tasks
  - Denoising [Graichen15, Liu16]
  - Semi-supervised learning / Classification [Kipf16, Manessi17]
  - Clustering [Tremblay14, Tremblay16]
  - Dimensionality reduction [Rui16]
- Application domains
  - Neuroimaging / Brain activity analysis [Huang16, Smith17, Ktena17]
  - Social network analysis (e.g., community detection [Bruna17], recommendation [Monti17], link prediction [Schlichtkrull17])
  - Urban computing (e.g., mobility inference [Dong13])
  - Computer graphics [Monti16, Yi16, Wang17, Simonovsky17]
  - Geoscience and remote sensing [Bayram17]

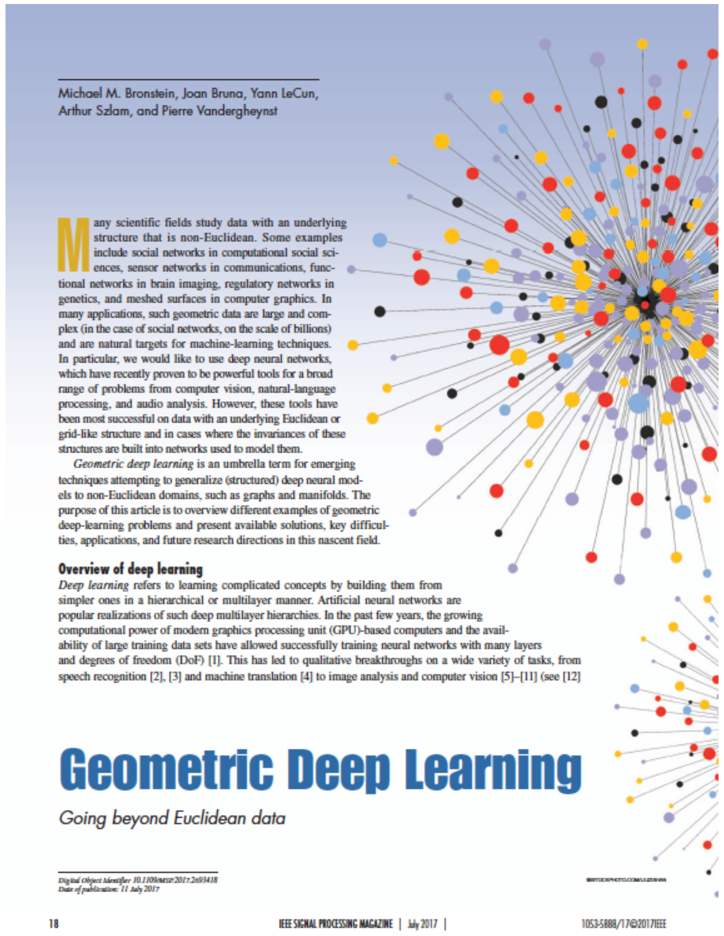
Future of GSP

- Mathematical models for graph signals
  - global and local smoothness / regularity
  - underlying physical processes
- Graph construction
  - how to infer topologies given observed data?
- Fast implementation
  - fast graph Fourier transform
  - distributed processing
- Connection to / combination with other fields
  - statistical machine learning
  - deep learning (on graphs and manifolds)
- Applications

References

- Three review papers:





Resources

- Graph signal processing
  - MATLAB toolbox: https://lts2.epfl.ch/gsp/
  - Python toolbox: https://pygsp.readthedocs.io/en/stable/
- Spectral graph wavelet transform
  - MATLAB toolbox: https://wiki.epfl.ch/sgwt
  - Python toolbox: https://github.com/aweinstein/pysgwt
- Topology inference
  - Tutorial: http://web.media.mit.edu/~xdong/presentation/GSP_GraphLearning.pdf
- Geometric deep learning
  - Workshops, tutorials, papers and code: http://geometricdeeplearning.com

contact: xiaowen.dong@eng.ox.ac.uk